



Benutzerhandbuch für Version 1

AWS Command Line Interface



AWS Command Line Interface: Benutzerhandbuch für Version 1

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

.....	xiv
Über den AWS CLI	1
Informationen zu AWS CLI Version 1	2
Wartung und Support für SDK Hauptversionen	2
Über Amazon Web Services	2
Informationen zu den Beispielen	3
Zusätzliche Dokumentation und Ressourcen	4
AWS CLI Dokumentation und Ressourcen	4
Sonstiges AWS SDKs und Tools	5
Installieren Sie das AWS CLI	6
Anforderungen an Python-Versionen	6
Amazon Linux	7
Voraussetzungen	7
pip	8
yum	9
Behebung von AWS CLI Installations- und Deinstallationsfehlern	10
Linux	10
Voraussetzungen	11
Installieren und Deinstallieren mit dem gebündelten Installationsprogramm	11
Installieren und Deinstallieren mit pip	17
Installieren und Deinstallieren mit pip	21
Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem Befehlszeilenpfad hinzu	23
Behebung von AWS CLI Installations- und Deinstallationsfehlern	24
macOS	24
Voraussetzungen	25
Installieren und Deinstallieren mit dem gebündelten Installationsprogramm	25
Installieren und Aktualisieren mit pip	31
Behebung von AWS CLI Installations- und Deinstallationsfehlern	35
Windows	35
Installation, Aktualisierung und Deinstallation mithilfe des MSI Installationsprogramms	36
Installieren, Aktualisieren und Deinstallieren mit Python und pip	38
Fügen Sie die AWS CLI ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu	39
Behebung von AWS CLI Installations- und Deinstallationsfehlern	41
Virtualenv	42

Voraussetzungen	42
Installieren und Aktualisieren in einer virtuellen Umgebung	43
Behebung von AWS CLI Installations- und Deinstallationsfehlern	44
Konfigurieren Sie die AWS CLI	45
Vorrang der Konfiguration und der Anmeldeinformationen	45
Weitere Themen in diesem Abschnitt	46
Einstellungen für die Konfiguration und die Anmeldeinformationsdatei in der AWS CLI	47
Formatieren der Konfigurations- und Anmeldeinformationsdateien	47
Wo werden Konfigurationseinstellungen gespeichert?	53
Verwenden von benannten Profilen	54
Festlegen und Anzeigen von Konfigurationseinstellungen mithilfe von Befehlen	55
Befehlsbeispiele für das Festlegen neuer Konfigurationen und Anmeldeinformationen	57
Unterstützte Einstellungen in der config-Datei	58
Umgebungsvariablen	77
Festlegen von Umgebungsvariablen	77
AWS CLI unterstützte Umgebungsvariablen	79
Befehlszeilenoptionen in der AWS CLI	89
Verwenden von Befehlszeilenoptionen	90
AWS CLI unterstützte globale Befehlszeilenoptionen	90
Häufige Verwendungsweisen von Befehlszeilenoptionen	94
Konfiguration der Befehlsvervollständigung in AWS CLI	94
Funktionsweise	95
Konfigurieren der Befehlsvervollständigung unter Linux oder macOS	96
Konfigurieren der Befehlsvervollständigung unter Windows	100
Wiederholversuche	101
Verfügbare Wiederholungsmodi	101
Konfigurieren eines Wiederholungsversuchsmodus	104
Anzeigen von Protokollen von Wiederholungsversuchen	105
Verwendung eines HTTP Proxys für AWS CLI	106
Verwenden der -Beispiele	107
Authentifizieren bei einem Proxy	108
Verwendung eines Proxys auf EC2 Amazon-Instances	108
Fehlerbehebung	109
Endpunkte	109
Festlegen eines Endpunkts für einen einzelnen Befehl	110
Legen Sie den globalen Endpunkt für alle fest AWS-Services	110

Stellen Sie ein, dass FIPs Endpunkte für alle verwendet werden AWS-Services	111
So einstellen, dass Dual-Stack-Endpunkte für alle AWS-Services verwendet werden	113
Festlegen servicespezifischer Endpunkte	114
Priorität der Endpunktkonfiguration und der Einstellungen	117
Authentifizierung und Anmeldeinformationen	119
Vorrang der Konfiguration und der Anmeldeinformationen	119
Weitere Themen in diesem Abschnitt	120
Kurzfristige Anmeldeinformationen	120
IAMRollen	122
Voraussetzungen	122
Überblick über die Verwendung von IAM Rollen	122
Konfigurieren und Verwenden einer Rolle	124
Verwenden MFA	126
Kontenübergreifende Rollen und externe ID	128
Angabe eines Rollensitzungsnamens für eine einfachere Prüfung	128
Übernehmen einer Rolle mit Web-Identität	129
Anmeldeinformationen aus dem Cache löschen	130
IAMBenutzer	131
Schritt 1: Erstellen Sie Ihren IAM Benutzer	131
Schritt 2: Abrufen Ihrer Zugriffsschlüssel	132
Konfigurieren Sie AWS CLI	132
Verwendung von EC2 Amazon-Instance-Metadaten als Anmeldeinformationen in der AWS CLI	133
Voraussetzungen	134
Konfiguration eines Profils für EC2 Amazon-Metadaten	134
Externe Anmeldeinformationen	135
Mit dem AWS CLI	139
Hilfe	140
Der integrierte AWS CLI -help-Befehl	140
AWS CLI Referenzhandbuch	145
APIDokumentation	145
Behebung von Fehlern	146
Weitere Hilfe	146
Befehlsstruktur	146
Befehlsstruktur	146
Wait-Befehle	148

Angeben von Parameterwerten	149
Allgemeine Parametertypen	150
Anführungszeichen mit Zeichenfolgen	155
Parameter aus Dateien	160
Generieren Sie eine Skeleton-Vorlage CLI	164
Syntax-Kurznotation	170
Steuern der Befehlsausgabe	173
Sensible Ausgabe	173
Serverseitige und clientseitige Ausgabeoptionen	174
Ausgabeformat	175
Paginierung	183
-Filterausgabe	185
Rückgabecodes	210
Aliasnamen	211
Voraussetzungen	212
Schritt 1: Erstellen der Aliasdatei	212
Schritt 2: Erstellen eines Alias	213
Schritt 3: Aufruf eines Alias	216
Beispiele für das Alias-Repository	218
Ressourcen	220
Codebeispiele	221
Beispiele für geführte Befehle	221
DynamoDB	222
Amazon EC2	226
S3 Glacier	246
IAM	253
Amazon S3	257
Amazon SNS	277
Befehlsbeispiele	279
ACM	286
APIGateway	297
APIGateway HTTP und WebSocket API	354
APIGateway-Verwaltung API	400
App Mesh	402
App Runner	447
AWS AppConfig	482

Application Auto Scaling	517
Application Discovery Service	535
AppRegistry	541
Athena	553
Auto Scaling	587
Auto Scaling Plans	658
AWS Backup	666
AWS Batch	672
AWS Budgets	687
Amazon Chime	698
Cloud Control API	772
AWS Cloud Map	778
AWS Cloud9	788
AWS CloudFormation	797
CloudFront	846
Amazon CloudSearch	916
CloudTrail	917
CloudWatch	934
CloudWatch Logs	949
CloudWatch Netzwerküberwachung	955
CodeArtifact	968
CodeBuild	996
CodeCommit	1061
CodeDeploy	1136
CodeGuru Prüfer	1176
CodePipeline	1195
AWS CodeStar Benachrichtigungen	1227
CodeConnections	1238
Amazon Cognito Identity	1247
Amazon Cognito Identity Provider	1253
Amazon Comprehend	1321
Amazon Comprehend Medical	1458
AWS Config	1494
Amazon Connect	1518
AWS Cost and Usage Report	1535
Cost Explorer Explorer-Dienst	1538

Firehose	1546
Amazon Data Lifecycle Manager	1549
AWS Data Pipeline	1555
DataSync	1565
DAX	1570
Detektivisch	1589
Device Farm	1601
AWS Direct Connect	1606
AWS Directory Service	1657
AWS DMS	1659
Amazon DocumentDB	1703
DynamoDB	1761
DynamoDB-Streams	1858
Amazon EC2	1865
Amazon EC2 Instance Connect	2534
Amazon ECR	2535
Amazon ECR Public	2567
Amazon ECS	2575
Amazon EFS	2660
Amazon EKS	2668
Elastic Beanstalk	2748
Elastic Load Balancing — Version 1	2778
Elastic Load Balancing — Version 2	2806
Elastic Transcoder	2860
ElastiCache	2888
MediaStore	2994
Amazon EMR	3012
Amazon EMR auf EKS	3061
EventBridge	3062
Firewall Manager	3069
AWS FIS	3079
Amazon GameLift	3098
Global Accelerator	3132
AWS Glue	3171
GuardDuty	3193
AWS Health	3212

HealthImaging	3219
HealthLake	3246
HealthOmics	3258
IAM	3326
IAM Access Analyzer	3467
Image Builder	3503
Incident Manager	3545
Incident-Manager-Kontakte	3567
Amazon Inspector	3591
AWS IoT	3635
AWS IoT 1-Click Geräte	3817
AWS IoT 1-Click Projekte	3827
AWS IoT Analytics	3838
Device Advisor	3866
AWS IoT data	3881
AWS IoT Events	3884
AWS IoT Events-Data	3909
AWS IoT Greengrass	3935
AWS IoT Greengrass V2	4021
AWS IoT Jobs SDK release	4047
AWS IoT SiteWise	4051
AWS IoT Things Graph	4101
AWS IoT Wireless	4128
Amazon IVS	4165
Amazon IVS Chat	4205
Amazon IVS Echtzeit-Streaming	4218
Amazon Kendra	4249
Kinesis	4258
AWS KMS	4277
Lake Formation	4349
Lambda	4401
License Manager	4443
Lightsail	4456
Macie	4582
Amazon Managed Grafana	4587
MediaConnect	4589

MediaConvert	4605
MediaLive	4631
MediaPackage	4637
MediaPackage VOD	4652
MediaStore Datenebene	4665
MediaTailor	4670
MemoryDB	4675
Amazon MSK	4712
Network Manager	4721
Nimble Studio	4759
OpenSearch Service	4778
AWS OpsWorks	4792
AWS OpsWorks CM	4849
Organisationen	4865
AWS Outposts	4904
AWS Payment Cryptography	4908
AWS Payment Cryptography Datenebene	4928
Amazon Pinpoint	4938
Amazon Polly	4961
AWS-Preisliste	4968
AWS Private CA	4973
AWS Proton	4980
QLDB	4993
Amazon RDS	5016
RDSAmazon-Datenservice	5215
RDSPerformance Insights von Amazon	5219
Amazon-Redshift	5223
Amazon Rekognition	5303
AWS RAM	5380
Ressourcen-Explorer	5404
Ressourcengruppen	5427
Kennzeichen von Resource Groups API	5440
AWS RoboMaker	5444
Route 53	5480
Route 53-Domainregistrierung	5494
Route 53 53-Profile	5521

Route 53 Resolver	5533
Amazon S3	5577
Amazon S3 S3-Steuerung	5671
S3 Glacier	5687
Secrets Manager	5710
Security Hub	5739
Security Lake	5818
AWS Serverless Application Repository	5854
Servicekatalog	5856
Service Quotas	5887
Amazon SES	5897
Shield	5911
Signer	5927
Snowball	5937
Amazon SNS	5939
Amazon SQS	5960
Storage Gateway	5981
AWS STS	5985
AWS Support	5994
Amazon SWF	6007
Systems Manager	6023
Amazon Textract	6201
Amazon Transcribe	6213
Amazon Translate	6257
Trusted Advisor	6258
Verified Permissions	6279
VPCGitter	6305
AWS WAF Classic	6333
AWS WAF Classic Regional	6339
AWS WAFV2	6345
Amazon WorkDocs	6390
Amazon WorkMail	6423
WorkMail Amazon-Nachrichtenfluss	6447
WorkSpaces	6449
X-Ray	6464
Bash-Skript-Beispiele	6482

DynamoDB	6483
Amazon EC2	6555
HealthImaging	6661
IAM	6670
Amazon S3	6725
AWS STS	6749
Sicherheit	6752
Datenschutz	6753
Datenverschlüsselung	6754
Identitäts- und Zugriffsverwaltung	6754
Zielgruppe	6755
Authentifizierung mit Identitäten	6755
Verwalten des Zugriffs mit Richtlinien	6759
Wie AWS-Services arbeiten Sie mit IAM	6762
Fehlerbehebung bei AWS Identität und Zugriff	6762
Compliance-Validierung	6765
Ausfallsicherheit	6766
Sicherheit der Infrastruktur	6767
Erzwingen einer Mindestversion TLS	6768
Beheben von Fehlern	6772
Allgemeine Fehlerbehebung, die Sie zuerst versuchen sollten	6772
Überprüfen Sie die Formatierung Ihrer AWS CLI Befehle	6773
Überprüfen Sie, ob AWS-Region Ihr AWS CLI Befehl verwendet	6773
Sicherstellen, dass Sie eine aktuelle Version der AWS CLI ausführen	6774
Verwenden der Option --debug	6774
Aktivieren und überprüfen Sie die AWS CLI Befehlsverlaufsprotokolle	6780
Vergewissern Sie sich, dass Ihr konfiguriert ist AWS CLI	6781
Fehler aufgrund eines nicht gefundenen Befehls	6781
Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück	6784
Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI	6785
Hat einen Befehl mit einem unvollständigen Parameternamen AWS CLI verarbeitet	6786
Fehler aufgrund einer Zugriffsverweigerung	6788
Ungültige Anmeldeinformationen und Schlüsselfehler	6789
Fehler aufgrund einer nicht übereinstimmenden Signatur	6790
Fehler aufgrund nicht gefundener Windows-Konsole	6792
SSLZertifikatsfehler	6792

Ungültige JSON Fehler	6794
Weitere Ressourcen	6796
Dokumentverlauf	6797

Diese Dokumentation bezieht sich AWS CLI nur auf Version 1 von. Dokumentation zu Version 2 von finden Sie im [Benutzerhandbuch für Version 2. AWS CLI](#)

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.

Was ist die AWS Command Line Interface Version 1?

Note

Die AWS CLI Version 1 ist nicht die neueste Version von AWS CLI. Einige in AWS CLI Version 2 eingeführte Funktionen werden nicht auf Version 1 zurückportiert und Sie müssen ein Upgrade durchführen, um auf diese Funktionen zugreifen zu können. Es gibt einige größere Änderungen gegenüber Version 1, die möglicherweise eine Änderung Ihrer Skripts erfordern. Eine Liste der größeren Änderungen in Version 2 finden Sie unter [Größere Änderungen](#) im AWS CLI -Benutzerhandbuch der Version 2.

The AWS Command Line Interface (AWS CLI) ist ein Open-Source-Tool, mit dem Sie mithilfe von Befehlen in Ihrer Befehlszeilen-Shell mit AWS Diensten interagieren können. Mit minimaler Konfiguration AWS CLI können Sie mit der Ausführung von Befehlen beginnen, die Funktionen implementieren, die denen entsprechen, die vom Browser bereitgestellt werden, von der Befehlszeile in Ihrem Terminalprogramm AWS Management Console aus:

- Linux-Shells – Verwenden Sie häufig genutzte Shell-Programme, wie z. B. [bash](#), [zsh](#) und [tcsh](#), um Befehle in Linux oder macOS auszuführen.
- Windows-Befehlszeile — Führen Sie unter Windows Befehle an der Windows-Eingabeaufforderung oder in PowerShell aus.
- Remote — Führen Sie Befehle auf Amazon Elastic Compute Cloud (AmazonEC2) -Instances über ein Remote-Terminal-Programm wie PuTTY oder SSH, oder mit AWS Systems Manager.

Alle Verwaltungs-, AWS Verwaltungs- und Zugriffsfunktionen von IaaS (Infrastructure as a Service) in der AWS Management Console sind im AWS API und AWS CLI verfügbar. Neue AWS IaaS-Funktionen und -Services bieten volle AWS Management Console Funktionalität während API und CLI bei der Markteinführung oder innerhalb von 180 Tagen nach der Markteinführung.

Das AWS CLI bietet direkten Zugang zur Öffentlichkeit APIs von AWS Diensten. Mit dem können Sie die AWS CLI Funktionen eines Dienstes erkunden und Shell-Skripts zur Verwaltung Ihrer Ressourcen entwickeln. Zusätzlich zu den API gleichwertigen Befehlen auf niedriger Ebene bieten mehrere AWS Dienste Anpassungen für. AWS CLI Anpassungen können Befehle auf höherer Ebene beinhalten, die die Verwendung eines Dienstes mit komplexen Funktionen vereinfachen. API

Informationen zu AWS CLI Version 1

Die AWS CLI Version 1 ist das Original AWS CLI, und wir unterstützen sie weiterhin. Wichtige neue Funktionen, die in AWS CLI Version 2 eingeführt wurden, werden jedoch möglicherweise nicht auf AWS CLI Version 1 zurückportiert. Um diese Funktionen nutzen zu können, müssen Sie AWS CLI Version 2 installieren. Die AWS CLI Version 1 wurde mit SDK for Python erstellt und erfordert daher die Installation einer kompatiblen Version von Python.

Informationen zur Installation der AWS CLI Version 1 finden Sie unter [Installieren Sie das AWS CLI](#).

Wenn Sie die zurzeit installierte Version überprüfen möchten, verwenden Sie den folgenden Befehl:

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Den Versionsverlauf finden Sie im [Changelog für AWS CLI Version 1. GitHub](#)

Wartung und Support für SDK Hauptversionen

Informationen zu Wartung und Support für SDK Hauptversionen und die ihnen zugrunde liegenden Abhängigkeiten finden Sie im Referenzhandbuch [AWS SDKs und im Tools-Referenzhandbuch](#):

- [AWS SDKs Richtlinien zur Wartung von Tools](#)
- [AWS SDKs und Matrix zur Unterstützung der Tools-Versionen](#)

Über Amazon Web Services

Amazon Web Services (AWS) ist eine Sammlung von digitalen Infrastruktur-Services, die Entwickler bei der Anwendungsentwicklung nutzen können. Die Dienste umfassen Computer-, Speicher-, Datenbank- und Anwendungssynchronisierung (Messaging und Queuing). AWS verwendet ein pay-as-you-go Servicemodell. Berechnet werden Ihnen nur die Services, die Sie bzw. Ihre Anwendungennutzen. Außerdem AWS bietet es ein kostenloses AWS Nutzungskontingent an, um es als Plattform für Prototyping und Experimente zugänglicher zu machen. Im Rahmen dieses Kontingents sind die Services bis zu einem bestimmten Nutzungsumfang kostenlos. [Weitere Informationen zu den AWS Kosten und dem kostenlosen Kontingent finden Sie unter AWS Kostenloses Kontingent](#). Um ein AWS Konto zu erhalten, öffnen Sie die [AWS Startseite](#) und wählen Sie dann [AWS Konto erstellen](#).

Über die Beispiele im AWS CLI Benutzerhandbuch

Die Beispiele AWS Command Line Interface (AWS CLI) in diesem Handbuch sind nach den folgenden Konventionen formatiert:

- **Aufforderung** – Die Eingabeaufforderung verwendet die Linux-Eingabeaufforderung und wird als (\$) angezeigt. Für Windows-spezifische Befehle wird C:\> als Eingabeaufforderung verwendet. Lassen Sie das Eingabeaufforderungssymbol weg, wenn Sie Befehle eingeben.
- **Verzeichnis** – Wenn Befehle in einem bestimmten Verzeichnis ausgeführt werden müssen, steht der Name des Verzeichnisses vor dem Eingabeaufforderungssymbol.
- **Benutzereingabe** – Befehlstext, den Sie in der Befehlszeile eingeben, ist als **user input** formatiert.
- **Ersetzbarer Text** — Variabler Text, einschließlich der Namen von Ressourcen, die Sie auswählen oder von AWS Diensten IDs generiert wurden und die Sie in Befehle aufnehmen müssen, wird wie folgt formatiert *replaceable text*. Bei mehrzeiligen Befehlen oder Befehlen, für die eine bestimmte Tastatureingabe erforderlich ist, können Tastaturbefehle auch als austauschbarer Text angezeigt werden.
- **Ausgabe** — Von AWS Diensten zurückgegebene Ausgaben werden unter Benutzereingaben angezeigt und sind formatiert als `computer output`

Das folgende **aws configure**-Befehlsbeispiel veranschaulicht die Benutzereingabe, den ersetzbaren Text und die Ausgabe:

1. Geben Sie in der Befehlszeile **aws configure** ein, und drücken Sie dann die Eingabetaste.
2. Es werden Textzeilen AWS CLI ausgegeben, in denen Sie aufgefordert werden, zusätzliche Informationen einzugeben.
3. Geben Sie Ihre entsprechenden Zugriffsschlüssel der Reihe nach ein und betätigen Sie dann die Eingabetaste.
4. Geben Sie dann einen AWS Regionsnamen im angezeigten Format ein, drücken Sie die EINGABETASTE und drücken Sie dann ein letztes Mal die EINGABETASTE, um die Einstellung für das Ausgabeformat zu überspringen.
5. Der Befehl Enter wird in diesem Fall als ersetzbarer Text dargestellt, da es für diese Zeile keine Benutzereingabe gibt.

```
$ aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: ENTER
```

Das folgende Beispiel zeigt einen einfachen Befehl mit Ausgabe. Zum Verwenden dieses Beispiels geben Sie den vollständigen Befehlstext (den markierten Text nach der Eingabeaufforderung) ein und drücken Sie dann die Eingabetaste. Der Name der Sicherheitsgruppe, *my-sg*, ist durch den gewünschten Sicherheitsgruppennamen ersetzbar. Das JSON Dokument, einschließlich der geschweiften Klammern, wird ausgegeben. Wenn Sie Ihre Ausgabe so konfigurieren CLI, dass sie im Text- oder Tabellenformat ausgegeben wird, wird die Ausgabe anders formatiert. [JSON](#) ist das Standardausgabeformat.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group"
{
  "GroupId": "sg-903004f8"
}
```

Zusätzliche Dokumentation und Ressourcen für AWS CLI

AWS CLI Dokumentation und Ressourcen

Zusätzlich zu diesem Benutzerhandbuch finden Sie im Folgenden wertvolle Online-Ressourcen, wenn Sie den verwenden AWS CLI.

- [AWS CLI Referenzhandbuch für Version 1](#)
- AWS CLI Repository mit [Beispielen für Bash-Skriptcode](#). Beispiele für Open-Source-Bash-Skripte. Bash-Skriptbeispiele finden Sie im [AWS Code Examples Repository unter GitHub](#)
- [AWS CLI GitHub Repository](#). Sie können den Quellcode für das AWS CLI On einsehen und forken GitHub. Treten Sie der Benutzergemeinschaft bei, GitHubum Feedback zu geben, Funktionen anzufordern und Ihre eigenen Beiträge einzureichen. Dies beinhaltet das Anzeigen und Bereitstellen von Befehlsbeispielen für die AWS CLI Dokumentation.
- [AWS CLI Alias-Beispiel-Repository](#), in dem Sie AWS CLI Alias-Beispiele ansehen und forken können GitHub.
- [AWS CLI Version 1: Changelog](#)
- [AWS CLI Änderungsprotokoll für Version 2](#)

Sonstiges AWS SDKs und Tools

Abhängig von Ihrem Anwendungsfall möchten Sie vielleicht eines der beiden Tools auswählen, das AWS SDKs Ihren Anforderungen besser entspricht:

- [AWS SDKs und Referenzhandbuch für Tools](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for Kotlin](#)
- [AWS SDK for .NET](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for PHP](#)
- [AWS Tools for PowerShell](#)
- [AWS SDK for Ruby](#)
- [AWS SDK for Rust](#)
- [AWS SDK für SAP ABAP](#)
- [AWS SDK for Swift](#)
- [AWS Amplify](#)

Installation, Aktualisierung und Deinstallation von AWS CLI

In diesem Thema finden Sie Links zum Installieren, Aktualisieren und Deinstallieren der Originalversion der AWS Command Line Interface (AWS CLI). Die AWS CLI Version 1 wird derzeit unterstützt, aber neue Funktionen, die der AWS CLI Version 2 hinzugefügt wurden, werden der AWS CLI Version 1 möglicherweise nicht hinzugefügt. Um diese Funktionen nutzen zu können, müssen Sie AWS CLI Version 2 installieren. Informationen zur Installation von Version 2 finden Sie unter [AWS CLI Version 2 installieren](#).

AWS CLI Anweisungen zur Installation, Aktualisierung und Deinstallation:

- [Anforderungen an Python-Versionen](#)
- [Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 auf Amazon Linux](#)
- [Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 unter Linux](#)
- [Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 auf macOS](#)
- [Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 unter Windows](#)
- [Installation und Aktualisierung der AWS CLI Version 1 in einer virtuellen Umgebung](#)

Anforderungen an Python-Versionen

Die AWS CLI Version 1 wurde mit SDK for Python erstellt und erfordert daher die Installation einer kompatiblen Version von Python.

Matrix zur Unterstützung der Python-Version

AWS CLI Version	Unterstützte Python-Versionen
1.32.0 bis aktuell	Python 3.8 oder höher
1.27.0 bis 1.31.x	Python 3.7+
1.20.0 bis 1.26.x	Python 3.6+
1.19.0 bis 1.19.x	Python 2.7+ oder Python 3.6+
1.17 - 1.18.x	Python 2.7+, Python 3.4+

AWS CLI Version	Unterstützte Python-Versionen
1.0 - 1.16.x	Python 2.6 und älter, Python 3.3 und älter

Informationen zu den neuesten Versionen von AWS CLI finden Sie im [Changelog für AWS CLI Version 2](#). [GitHub](#)

Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 auf Amazon Linux

Die AWS CLI Version 1 ist auf Amazon Linux und Amazon Linux 2 vorinstalliert. Prüfen Sie die zurzeit installierte Version mit dem folgenden Befehl.

```
$ aws --version  
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Je nachdem, wann Sie Ihre Amazon Linux-Instance erstellt haben, ist AWS CLI Version 1 mit einem der folgenden Paketmanager vorinstalliert:

- [pip](#)
- [yum](#)

Voraussetzungen

Sie müssen auch Python 3.8 oder höher installieren. Installationsanweisungen finden Sie auf der Seite „[Python herunterladen](#)“ im Python-Anfängerhandbuch.

Matrix zur Unterstützung der Python-Version

AWS CLI Version	Unterstützte Python-Versionen
1.32.0 bis aktuell	Python 3.8 oder höher
1.27.0 bis 1.31.x	Python 3.7+
1.20.0 bis 1.26.x	Python 3.6+

AWS CLI Version	Unterstützte Python-Versionen
1.19.0 bis 1.19.x	Python 2.7+ oder Python 3.6+
1.17 - 1.18.x	Python 2.7+, Python 3.4+
1.0 - 1.16.x	Python 2.6 und älter, Python 3.3 und älter

Installieren, Aktualisieren und Deinstallieren mithilfe von pip

Die meisten Amazon Linux-Instances verwenden Pip, um die AWS CLI Version 1 vorzuinstallieren.

Installieren oder aktualisieren Sie die AWS CLI Version 1 auf Amazon Linux mit pip

Gehen Sie wie folgt vor, um die neueste AWS CLI Version von Version 1 für den aktuellen Benutzer zu installieren.

1. Wenn Sie Python Version 3 oder höher installiert haben, empfehlen wir Ihnen, `pip3` zu verwenden. Verwenden Sie diese Option `pip3 install`, um die neueste Version von AWS CLI Version 1 zu installieren oder auf diese zu aktualisieren. Wenn Sie den Befehl innerhalb einer [virtuellen Python-Umgebung \(venv\)](#) ausführen, müssen Sie die Option `--user` nicht verwenden.

```
$ pip3 install --upgrade --user awscli
```

2. Stellen Sie sicher, dass der Ordner, in dem `aws` enthalten ist, Teil Ihrer `PATH`-Variablen ist.
 - a. Suchen Sie das Profilskript der Shell in Ihrem Benutzerverzeichnis. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~  
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

- b. Fügen Sie einen Exportbefehl an das Ende Ihres Profilskripts hinzu und orientieren Sie sich dabei an folgendem Beispiel.

```
export PATH=$HOME/.local/bin:$PATH
```

Dieser Befehl fügt den Pfad `$HOME/.local/bin` in diesem Beispiel vor der bestehenden `$PATH`-Variablen ein.

- c. Laden Sie das Profil erneut in Ihre aktuelle Sitzung, damit die Änderungen wirksam werden.

```
$ source ~/.bash_profile
```

3. Verwenden Sie den Befehl `aws --version`, um zu überprüfen, ob die neue Version ausgeführt wird.

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Deinstallieren Sie die AWS CLI Version 1 mit Pip

Wenn Sie das deinstallieren müssen AWS CLI, verwenden `pip uninstall` Sie.

```
$ pip3 uninstall awscli
```

Installieren, Aktualisieren und Deinstallieren mithilfe von yum

Die meisten Amazon Linux 2-Instances verwenden Yum, um die AWS CLI Version 1 vorzuinstallieren.

Installieren oder aktualisieren Sie die AWS CLI Version 1 auf Amazon Linux mit Yum

Führen Sie den folgenden Befehl aus, um die neueste AWS CLI Version der Version 1 zu installieren, die auf Amazon Linux verfügbar ist.

```
$ sudo yum install awscli
```

Führen Sie den folgenden Befehl aus, um auf die neueste AWS CLI Version der Version 1 zu aktualisieren, die auf Amazon Linux verfügbar ist.

```
$ sudo yum update awscli
```

Verwenden Sie den Befehl `aws --version`, um zu überprüfen, ob Sie die neuere Version ausführen.

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Deinstallieren Sie AWS CLI Version 1 mit `yum`

Um das zu deinstallieren AWS CLI, verwenden Sie `yum remove`.

```
$ sudo yum remove awscli
```

Behebung von AWS CLI Installations- und Deinstallationsfehlern

Wenn Sie nach der Installation oder Deinstallation von auf Probleme stoßen AWS CLI, finden Sie die Schritte [Beheben von Fehlern](#) zur Fehlerbehebung unter. Die wichtigsten Maßnahmen zur Fehlerbehebung finden Sie unter [the section called "Fehler aufgrund eines nicht gefundenen Befehls"](#), [the section called "Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück"](#) und [the section called "Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI"](#).

Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 unter Linux

Sie können die AWS Command Line Interface (AWS CLI) Version 1 und ihre Abhängigkeiten auf den meisten Linux-Distributionen installieren, indem Sie den `pip` Paketmanager oder das mitgelieferte Installationsprogramm verwenden.

Das `awscli` Paket ist zwar in Repositories für andere Paketmanager wie `apt` und `verfügbar` `yum`, diese werden jedoch nicht von produziert, verwaltet oder unterstützt. AWS Wir empfehlen, dass Sie das nur AWS CLI von den offiziellen AWS Verteilungspunkten aus installieren, wie in diesem Handbuch beschrieben.

Sections

- [Voraussetzungen](#)
- [Installieren und deinstallieren Sie die AWS CLI Version 1 unter Linux mit dem mitgelieferten Installationsprogramm](#)

- [Installieren und deinstallieren Sie die AWS CLI Version 1 mit Pip](#)
- [Installieren und deinstallieren Sie die AWS CLI Version 1 mit Snapcraft](#)
- [Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem Befehlszeilenpfad hinzu](#)
- [Behebung von AWS CLI Installations- und Deinstallationsfehlern](#)

Voraussetzungen

Sie müssen auch Python 3.8 oder höher installieren. Installationsanweisungen finden Sie auf der Seite „[Python herunterladen](#)“ im Python-Anfängerhandbuch.

Matrix zur Unterstützung der Python-Version

AWS CLI Version	Unterstützte Python-Versionen
1.32.0 bis aktuell	Python 3.8 oder höher
1.27.0 bis 1.31.x	Python 3.7+
1.20.0 bis 1.26.x	Python 3.6+
1.19.0 bis 1.19.x	Python 2.7+ oder Python 3.6+
1.17 - 1.18.x	Python 2.7+, Python 3.4+
1.0 - 1.16.x	Python 2.6 und älter, Python 3.3 und älter

Installieren und deinstallieren Sie die AWS CLI Version 1 unter Linux mit dem mitgelieferten Installationsprogramm

Unter Linux oder macOS können Sie Version 1 der AWS CLI mithilfe des gebündelten Installationsprogramms installieren. Das gebündelte Installationsprogramm enthält alle Abhängigkeiten und kann offline verwendet werden.

Note

Das gebündelte Installationsprogramm unterstützt keine Installation in Pfaden, die Leerzeichen enthalten.

Themen

- [Installieren Sie die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm mit sudo](#)
- [Installieren Sie die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm ohne sudo](#)
- [Deinstallieren Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1.](#)

Installieren Sie die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm mit **sudo**

Mit den folgenden Schritten können Sie die AWS CLI Version 1 über die Befehlszeile auf einem beliebigen Build von Linux oder macOS installieren.

Hier finden Sie eine Zusammenfassung der im folgenden beschriebenen Installationsbefehle, die Sie ausschneiden und einfügen können, um sie als einzelnen Befehlssatz auszuführen.

Verwenden Sie für die AWS CLI neueste Version von den folgenden Befehlsblock:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
unzip awscli-bundle.zip
sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
unzip awscli-bundle.zip
sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

Folgen Sie diesen Schritten von der Befehlszeile aus, um AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm zu installieren.

Um die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm zu installieren

1. Laden Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1 mit einer der folgenden Methoden herunter.
 - Download über den `curl`-Befehl.

Verwenden Sie für die neueste Version von den AWS CLI folgenden Befehlsblock:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
```

- Download über den direkten Link.

Für die neueste Version der AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu der folgenden URL `awscli-bundle-1.16.312.zip` führen <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Extrahieren Sie die Dateien aus dem Paket. Wenn Sie nicht über `unzip` zum Extrahieren der Dateien verfügen, verwenden Sie den integrierten Paket-Manager der Linux-Distribution, um es zu installieren.

```
$ unzip awscli-bundle.zip
```

3. Führen Sie das Installationsprogramm aus. Das Installationsprogramm installiert das AWS CLI at `/usr/local/aws` und erstellt den Symlink `aws` im `/usr/local/bin` Verzeichnis. Durch die Verwendung der Option `-b` zum Erstellen eines symbolischen Links entfällt die Notwendigkeit, das Installationsverzeichnis in der `$PATH`-Variablen des Benutzers anzugeben. Dies sollte es allen Benutzern ermöglichen, das aufzurufen, AWS CLI indem sie es von einem beliebigen Verzeichnis `aws` aus aufrufen.

```
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

Standardmäßig wird das Installationskript unter der Standard-Systemversion von Python ausgeführt. Wenn Sie eine alternative Version von Python installiert haben und diese Version

zur Installation von verwenden möchten AWS CLI, führen Sie das Installationskript mit dieser Version über den absoluten Pfad zur ausführbaren Python-Datei wie folgt aus.

```
$ sudo /usr/local/bin/python3.7 awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

4. Stellen Sie sicher, dass das AWS CLI Programm korrekt installiert ist.

```
$ aws --version  
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Wenn Sie eine Fehlermeldung erhalten, finden Sie weitere Informationen unter [Behebung von Fehlern für den AWS CLI](#).

Installieren Sie die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm ohne **sudo**

Wenn Sie keine sudo Berechtigungen haben oder das Programm AWS CLI nur für den aktuellen Benutzer installieren möchten, können Sie eine modifizierte Version der vorherigen Befehle verwenden. Die ersten beiden Befehle sind identisch.

Verwenden Sie für die AWS CLI neueste Version von den folgenden Befehlsblock:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"  
unzip awscli-bundle.zip  
./awscli-bundle/install -b ~/bin/aws
```

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"  
unzip awscli-bundle.zip  
./awscli-bundle/install -b ~/bin/aws
```

Um die AWS CLI Version 1 für den aktuellen Benutzer zu installieren

1. Laden Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1 auf eine der folgenden Arten herunter.

- Download über den `curl`-Befehl.

Verwenden Sie für die neueste Version von den AWS CLI folgenden Befehlsblock:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
```

- Download über den direkten Link.

Für die neueste Version der AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu der folgenden URL `awscli-bundle-1.16.312.zip` führen <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Extrahieren Sie die Dateien mithilfe von `unzip` aus dem Paket. Wenn Sie nicht über `unzip` verfügen, verwenden Sie den integrierten Paket-Manager der Linux-Verteilung, um es zu installieren.

```
$ unzip awscli-bundle.zip
```

3. Führen Sie das Installationsprogramm aus. Das Installationsprogramm installiert das AWS CLI at `/usr/local/aws` und erstellt den Symlink `aws` im `/usr/local/bin` Verzeichnis. Der Befehl verwendet den `-b`-Parameter, um das Verzeichnis anzugeben, in dem das Installationsprogramm die `aws`-Symlink-Datei ablegt. Sie müssen über Schreibberechtigungen für den angegebenen Ordner verfügen.

```
$ ./awscli-bundle/install -b ~/bin/aws
```

Dadurch wird das AWS CLI an der Standardposition (`~/local/lib/aws`) installiert und ein symbolischer Link (Symlink) unter `~/bin/aws` erstellt. Stellen Sie sicher, dass `~/bin` in Ihrer PATH-Umgebungsvariablen enthalten ist, damit der Symlink funktioniert.

```
$ echo $PATH | grep ~/bin // See if $PATH contains ~/bin (output will be empty
if it doesn't)
$ export PATH=~/bin:$PATH // Add ~/bin to $PATH if necessary
```

4. Stellen Sie sicher, dass das Verzeichnis, in dem AWS CLI Version 1 Teil Ihrer PATH Variablen ist.
 - a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~
. .. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
 - Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc` oder `.login`.
- b. Fügen Sie einen Exportbefehl an das Ende Ihres Profilskripts hinzu und orientieren Sie sich dabei an folgendem Beispiel.

```
export PATH=~/local/bin:$PATH
```

Dieser Befehl fügt den Pfad `~/local/bin` in diesem Beispiel vor der bestehenden PATH-Variablen ein.

- c. Laden Sie das Profil erneut in Ihre aktuelle Sitzung, damit die Änderungen wirksam werden.

```
$ source ~/.bash_profile
```

5. Stellen Sie sicher, dass das korrekt AWS CLI installiert ist.

```
$ aws --version
```

```
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```


Wenn Sie eine Fehlermeldung erhalten, finden Sie weitere Informationen unter [Behebung von Fehlern für den AWS CLI](#).

Deinstallieren Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1.

1. Wenn Sie das AWS CLI mit dem mitgelieferten Installationsprogramm installiert haben, folgen Sie diesen Anweisungen. Das gebündelte Installationsprogramm speichert bis auf den symbolischen Link nichts außerhalb des Installationsverzeichnisses. Die Deinstallation kann daher einfach durch Löschen dieser beiden Elemente erfolgen.

```
$ sudo rm -rf /usr/local/aws
$ sudo rm -rf /usr/local/bin/aws
```

2. (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

 Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `~/.aws/`. Wenn Ihr Benutzer Schreibrechte für dieses Verzeichnis hat, müssen Sie es nicht verwenden `sudo`.

```
$ sudo rm -r ~/.aws/
```

Installieren und deinstallieren Sie die AWS CLI Version 1 mit Pip

Themen

- [Installieren von pip](#)
- [Installieren und aktualisieren Sie die AWS CLI Version 1 mit Pip](#)

- [Deinstallieren Sie das AWS CLI mithilfe von Pip](#)

Installieren von pip

Wenn Sie `pip` noch nicht installiert haben, können Sie es mit dem Skript installieren, das von der Python Packaging Authority bereitgestellt wird. Führen Sie `pip --version` aus, um zu überprüfen, ob Ihre Version von Linux bereits Python und `pip` enthält. Wir empfehlen Ihnen, den `pip3`-Befehl zu verwenden, wenn Sie Python-Version 3 oder höher installiert haben.

1. Laden Sie das Installationsskript mit dem Befehl `curl` herunter. Der folgende Befehl verwendet den Parameter `-O` (Großbuchstabe „O“), um anzugeben, dass die heruntergeladene Datei im aktuellen Verzeichnis unter Verwendung desselben Namens auf dem Remote-Host gespeichert werden soll.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

2. Führen Sie das Skript mit dem `python-` oder `python3-`Befehl aus, um die aktuelle Version von `pip` und andere erforderliche Supportpakete herunterzuladen. Wenn Sie den Schalter `--user` einschließen, installiert das Skript `pip` im Pfad `~/local/bin`.

```
$ python3 get-pip.py --user
```

3. Stellen Sie sicher, dass das Verzeichnis, in dem `pip` enthalten ist, Teil Ihrer `PATH`-Variablen ist.
 - a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~  
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
 - Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc` oder `.login`.
- b. Fügen Sie einen Exportbefehl an das Ende Ihres Profilskripts hinzu und orientieren Sie sich dabei an folgendem Beispiel.

```
export PATH=~/local/bin:$PATH
```

Dieser Befehl fügt den Pfad `~/local/bin` in diesem Beispiel vor der bestehenden PATH-Variablen ein.

- c. Laden Sie das Profil erneut in Ihre aktuelle Sitzung, damit die Änderungen wirksam werden.

```
$ source ~/.bash_profile
```

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob `pip` oder `pip3` korrekt installiert ist.

```
$ pip3 --version
pip 24.0 from ~/.local/lib/python3.7/site-packages (python 3.7)
```

Installieren und aktualisieren Sie die AWS CLI Version 1 mit Pip

1. Verwenden Sie den `pip`- oder `pip3`-Befehl, um die AWS CLI zu installieren oder zu aktualisieren. Wir empfehlen Ihnen, bei Verwendung von Python-Version 3 oder höher den `pip3`-Befehl zu verwenden. Der `--user` Switch `pip` installiert das AWS CLI zu `~/local/bin`.

Verwenden Sie für die AWS CLI neueste Version von den folgenden Befehlsblock:

```
$ pip3 install awscli --upgrade --user
```

Hängen Sie für eine bestimmte Version von zwei Gleichheitszeichen `=` und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version `1.16.312` wäre `==1.16.312` was zu dem folgenden Befehl führt:

```
$ pip3 install awscli==1.16.312 --upgrade --user
```

Note

Verwenden Sie geeignete Anführungszeichenregeln für Ihr Terminal. Um das Zeichen `=` verwenden zu können, müssen Sie möglicherweise einfache oder doppelte Anführungszeichen zur korrekten Maskierung verwenden. Im folgenden Beispiel werden einfache Anführungszeichen zur Maskierung verwendet:

```
$ pip3 install 'awscli==1.16.312' --upgrade --user
```

2. Stellen Sie sicher, dass das korrekt AWS CLI installiert ist.

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Wenn Sie eine Fehlermeldung erhalten, finden Sie weitere Informationen unter [Behebung von Fehlern für den AWS CLI](#).

Deinstallieren Sie das AWS CLI mithilfe von Pip

1. Wenn Sie die AWS CLI Version 1 mit installiert haben `pip`, müssen Sie auch mit `pip` deinstallieren.

```
$ pip uninstall awscli
```

Wenn Sie Version Python 2 oder 3 verwenden, müssen Sie möglicherweise den Befehl `pip2` oder `pip3` verwenden. Verwenden Sie den `aws --version` Befehl, um die Python-Version zu ermitteln, die Ihrer installierten AWS CLI Version 1 zugeordnet ist.

```
$ pip3 uninstall awscli
```

Möglicherweise müssen Sie das Eingabeaufforderungsfenster oder den Computer neu starten, um alle Dateien zu entfernen.

2. (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `~/.aws/`. Wenn Ihr Benutzer Schreibrechte für dieses Verzeichnis hat, müssen Sie es nicht verwenden `sudo`.


```
$ sudo rm -r ~/.aws/
```

Installieren und deinstallieren Sie die AWS CLI Version 1 mit Snapcraft

Themen

- [Installieren Sie Snap](#)
- [Installieren und aktualisieren Sie AWS CLI Version 1 mit Snap](#)
- [Deinstallieren Sie das AWS CLI mithilfe von Snap](#)

Installieren Sie Snap

Wenn Sie es noch nicht snap installiert haben, können Sie es mithilfe der Anweisungen von Canonical Snapcraft installieren. Führen Sie `snap version` aus, um zu sehen, ob Ihre Version von Linux bereits Folgendes enthält. `snap`

1. Installieren Sie Snapcraft auf Ihrer Plattform. Informationen zur Installation von Snapcraft finden Sie in der [Snap-Dokumentation unter Installation des Daemons](#).
2. Starten Sie Ihr System neu, damit Ihre PATH Variablen korrekt aktualisiert werden. Wenn Sie Probleme mit der Installation haben, folgen Sie den Schritten unter [Häufig auftretende Probleme beheben](#) in der Snap-Dokumentation.
3. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob die Installation korrekt snap ist.

```
$ snap version
```

Installieren und aktualisieren Sie AWS CLI Version 1 mit Snap

1. Führen Sie den folgenden `snap install` Befehl für AWS CLI Version 1 aus.

```
$ snap install aws-cli --channel=v1/stable --classic
```

Abhängig von Ihren Berechtigungen müssen Sie Ihren Befehl möglicherweise erweitern `sudo`.

```
$ sudo snap install aws-cli --channel=v1/stable --classic
```

2. Stellen Sie sicher, dass der korrekt AWS CLI installiert ist.

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Wenn Sie eine Fehlermeldung erhalten, finden Sie weitere Informationen unter [Behebung von Fehlern für den AWS CLI](#).

Deinstallieren Sie das AWS CLI mithilfe von Snap

1. Wenn Sie die AWS CLI Version 1 mit installiert habensnap, müssen Sie auch die Deinstallation mitsnap.

```
$ snap remove aws-cli
```

Möglicherweise müssen Sie das Eingabeaufforderungsfenster oder den Computer neu starten, um alle Dateien zu entfernen.

2. (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `~/.aws/`. Wenn Sie Schreibberechtigungen für dieses Verzeichnis haben, müssen Sie es nicht verwenden `sudo`.

```
$ sudo rm -r ~/.aws/
```

Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem Befehlszeilenpfad hinzu

Nach der Installation mit `pip` oder `snap` müssen Sie die `aws` ausführbare Datei möglicherweise der `PATH` Umgebungsvariablen Ihres Betriebssystems hinzufügen.

Sie können überprüfen, AWS CLI in welchem Ordner das `pip` installiert wurde, indem Sie den folgenden Befehl ausführen.

```
$ which aws
/home/username/.local/bin/aws
```

Sie können dies als `~/.local/bin/` referenzieren, da `/home/username` in Linux `~` entspricht.

Wenn Sie den Schalter `--user` ausgelassen und somit nicht im Benutzermodus installiert haben, befindet sich die ausführbare Datei möglicherweise im Ordner `bin` Ihrer Python-Installation. Wenn Sie nicht wissen, wo Python installiert ist, führen Sie diesen Befehl aus.

```
$ which python
/usr/local/bin/python
```

Die Ausgabe ist möglicherweise der Pfad zu einem Symlink, nicht zu der tatsächlichen ausführbaren Datei. Führen Sie `ls -al` aus, um festzustellen, wohin dieser verweist.

```
$ ls -al /usr/local/bin/python
/usr/local/bin/python -> ~/.local/Python/3.6/bin/python3.6
```

`pip` installiert Programme in demselben Ordner, der auch die Python-Anwendung enthält. Fügen Sie diesen Ordner Ihrer `PATH`-Variablen hinzu.

So ändern Sie die Variable **PATH**

1. Suchen Sie das Profilskript der Shell in Ihrem Benutzerverzeichnis. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`

- Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc` oder `.login`
2. Fügen Sie dem Profilskript einen Exportbefehl hinzu.

```
export PATH=~/.local/bin:$PATH
```

Dieser Befehl fügt der aktuellen PATH-Variablen einen Pfad, in diesem Beispiel `~/.local/bin`, hinzu.

3. Laden Sie hochgeladene Profil in Ihre aktuelle Sitzung.

```
$ source ~/.bash_profile
```

Behebung von AWS CLI Installations- und Deinstallationsfehlern

Wenn Sie nach der Installation oder Deinstallation von auf Probleme stoßen AWS CLI, finden Sie die Schritte [Beheben von Fehlern](#) zur Fehlerbehebung unter. Die wichtigsten Maßnahmen zur Fehlerbehebung finden Sie unter [the section called “Fehler aufgrund eines nicht gefundenen Befehls”](#), [the section called “Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück”](#) und [the section called “Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI”](#).

Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 auf macOS

Sie können die AWS Command Line Interface (AWS CLI) Version 1 und ihre Abhängigkeiten auf macOS installieren, indem Sie das mitgelieferte Installationsprogramm verwenden oder `pip`.

Sections

- [Voraussetzungen](#)
- [Installiere, aktualisiere und deinstalliere AWS CLI Version 1 auf macOS mit dem mitgelieferten Installer](#)
- [Installieren, aktualisieren und deinstallieren Sie die AWS CLI Version 1 mit Pip](#)
- [Behebung von AWS CLI Installations- und Deinstallationsfehlern](#)

Voraussetzungen

Bevor Sie die AWS CLI Version 1 auf macOS installieren können, stellen Sie sicher, dass Sie Python 3.8 oder höher installiert haben. Installationsanweisungen finden Sie auf der Seite „[Python herunterladen](#)“ im Python-Anfängerhandbuch.

Matrix zur Unterstützung der Python-Version

AWS CLI Version	Unterstützte Python-Versionen
1.32.0 bis aktuell	Python 3.8 oder höher
1.27.0 bis 1.31.x	Python 3.7+
1.20.0 bis 1.26.x	Python 3.6+
1.19.0 bis 1.19.x	Python 2.7+ oder Python 3.6+
1.17 - 1.18.x	Python 2.7+, Python 3.4+
1.0 - 1.16.x	Python 2.6 und älter, Python 3.3 und älter

Installiere, aktualisiere und deinstalliere AWS CLI Version 1 auf macOS mit dem mitgelieferten Installer

Unter Linux oder macOS können Sie Version 1 der AWS Command Line Interface (AWS CLI) mithilfe des gebündelten Installationsprogramms installieren. Das gebündelte Installationsprogramm enthält alle Abhängigkeiten und kann offline verwendet werden.

Das gebündelte Installationsprogramm unterstützt keine Installation in Pfaden, die Leerzeichen enthalten.

Themen

- [Installieren Sie die AWS CLI Version 1 mithilfe des mitgelieferten Installationsprogramms mit sudo](#)
- [Installieren Sie die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm ohne sudo](#)
- [Deinstallieren Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1.](#)

Installieren Sie die AWS CLI Version 1 mithilfe des mitgelieferten Installationsprogramms mit **sudo**

Mit den folgenden Schritten können Sie die AWS CLI Version 1 über die Befehlszeile auf einem beliebigen Build von macOS installieren.

Im Folgenden finden Sie eine Zusammenfassung der Installationsbefehle, die Sie ausschneiden und einfügen können, um sie als einzelnen Befehlssatz auszuführen.

Verwenden Sie für die AWS CLI neueste Version von den folgenden Befehlsblock:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
unzip awscli-bundle.zip
sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
unzip awscli-bundle.zip
sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

Um die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm zu installieren

1. Laden Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1 auf eine der folgenden Arten herunter:

- Download über den `curl`-Befehl.

Installieren Sie die neueste Version des AWS CLI mit dem folgenden Befehlsblock.

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```

Für eine spezifische Version von AWS CLI fügen Sie einen Bindestrich und die Versionsnummer an den Dateinamen an. In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
```

- Download über den direkten Link.

Für die neueste Version der AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu der folgenden URL `awscli-bundle-1.16.312.zip` führen <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Extrahieren (entpacken) Sie die Dateien aus dem Paket. Falls nichtunzip, verwenden Sie den integrierten Paketmanager Ihrer macOS Distribution, um ihn zu installieren.

```
$ unzip awscli-bundle.zip
```

3. Führen Sie das Installationsprogramm aus. Das Installationsprogramm installiert das AWS CLI at `/usr/local/aws` und erstellt den Symlink `aws` im `/usr/local/bin` Ordner. Durch die Verwendung der Option `-b` zum Erstellen eines symbolischen Links entfällt die Notwendigkeit, den Installationsordner in der `$PATH`-Variablen des Benutzers anzugeben. Dies sollte es allen Benutzern ermöglichen, das aufzurufen, AWS CLI indem sie es von einem beliebigen Verzeichnis `aws` aus aufrufen.

```
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

Standardmäßig wird das Installationskript unter der Standard-Systemversion von Python ausgeführt. Wenn Sie eine alternative Version von Python installiert haben und diese zur Installation von verwenden möchten AWS CLI, führen Sie das Installationskript mit dieser Version über den absoluten Pfad zur ausführbaren Python-Datei wie folgt aus.

```
$ sudo /usr/local/bin/python3.7 awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

4. Stellen Sie sicher, dass das korrekt AWS CLI installiert ist.

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Wenn Sie eine Fehlermeldung erhalten, finden Sie weitere Informationen unter [Behebung von Fehlern für den AWS CLI](#).

Installieren Sie die AWS CLI Version 1 mit dem mitgelieferten Installationsprogramm ohne **sudo**

Wenn Sie keine sudo Berechtigungen haben oder das Programm AWS CLI nur für den aktuellen Benutzer installieren möchten, können Sie eine modifizierte Version der vorherigen Befehle verwenden. Die ersten beiden Befehle sind identisch.

Verwenden Sie für die AWS CLI neueste Version von den folgenden Befehlsblock:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
unzip awscli-bundle.zip
./awscli-bundle/install -b ~/bin/aws
```

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
unzip awscli-bundle.zip
./awscli-bundle/install -b ~/bin/aws
```

Um die AWS CLI Version 1 für den aktuellen Benutzer zu installieren

1. Laden Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1 mit einer der folgenden Methoden herunter:
 - Download über den `curl`-Befehl.

Installieren Sie die neueste Version des AWS CLI mit dem folgenden Befehlsblock.

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```


Für eine spezifische Version von AWS CLI fügen Sie einen Bindestrich und die Versionsnummer an den Dateinamen an. In diesem Beispiel der Dateiname für Version **1.16.312** würde zu dem folgenden Befehl `awscli-bundle-1.16.312.zip` führen:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip" -o "awscli-bundle.zip"
```

- Download über den direkten Link.

Für die neueste Version der AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Hängen Sie für eine bestimmte Version von einen Bindestrich und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version **1.16.312** würde zu der folgenden URL `awscli-bundle-1.16.312.zip` führen <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Extrahieren Sie die Dateien aus dem Paket. Wenn Sie nicht über `unzip` verfügen, verwenden Sie den integrierten Paket-Manager der Linux-Verteilung, um es zu installieren.

```
$ unzip awscli-bundle.zip
```

3. Führen Sie das Installationsprogramm aus. Das Installationsprogramm installiert das AWS CLI at `/usr/local/aws` und erstellt den Symlink `aws` im `/usr/local/bin` Verzeichnis. Der Befehl verwendet den `-b`-Parameter, um das Verzeichnis anzugeben, in dem das Installationsprogramm die `aws`-Symlink-Datei ablegt. Sie müssen über Schreibberechtigungen für das angegebene Verzeichnis verfügen.

```
$ ./awscli-bundle/install -b ~/bin/aws
```

Dadurch wird das AWS CLI an der Standardposition (`~/local/lib/aws`) installiert und ein symbolischer Link (Symlink) unter erstellt. `~/bin/aws` Stellen Sie sicher, dass `~/bin` in Ihrer `$PATH`-Umgebungsvariablen enthalten ist, damit der Symlink funktioniert.

```
$ echo $PATH | grep ~/bin // See if $PATH contains ~/bin (output will be empty if it doesn't)
$ export PATH=~/bin:$PATH // Add ~/bin to $PATH if necessary
```

4. Stellen Sie sicher, dass der Ordner, in dem AWS CLI Version 1 installiert ist, Teil Ihrer `$PATH` Variablen ist.

- a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~  
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

- b. Fügen Sie einen Exportbefehl an das Ende Ihres Profilskripts hinzu und orientieren Sie sich dabei an folgendem Beispiel.

```
export PATH=~/.local/bin:$PATH
```

Dieser Befehl fügt den Pfad `~/.local/bin` in diesem Beispiel vor der bestehenden `PATH`-Variablen ein.

- c. Laden Sie das Profil erneut in Ihre aktuelle Sitzung, damit die Änderungen wirksam werden.

```
$ source ~/.bash_profile
```

5. Stellen Sie sicher, dass das korrekt AWS CLI installiert ist.

```
$ aws --version  
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Wenn Sie eine Fehlermeldung erhalten, finden Sie weitere Informationen unter [Behebung von Fehlern für den AWS CLI](#).

Deinstallieren Sie das mitgelieferte Installationsprogramm der AWS CLI Version 1.

1. Das gebündelte Installationsprogramm packt alles in das Installationsverzeichnis mit Ausnahme des optionalen Symlinks. Zur Deinstallation müssen Sie nur diese beiden Elemente löschen.

```
$ sudo rm -rf /usr/local/aws  
$ sudo rm /usr/local/bin/aws
```

- (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `~/.aws/`. Wenn Ihr Benutzer Schreibrechte für dieses Verzeichnis hat, müssen Sie es nicht verwenden `sudo`.

```
$ sudo rm ~/.aws/
```

Installieren, aktualisieren und deinstallieren Sie die AWS CLI Version 1 mit Pip

Sie können `pip` direkt zur Installation der AWS CLI verwenden.

Themen

- [Installieren von pip](#)
- [Installieren und aktualisieren Sie das AWS CLI mit Pip](#)
- [Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem macOS-Befehlszeilenpfad hinzu](#)
- [Deinstalliere das AWS CLI mit Pip](#)

Installieren von pip

Wenn Sie `pip` noch nicht installiert haben, können Sie es mit dem Skript installieren, das von der Python Packaging Authority bereitgestellt wird. Führen Sie `pip --version` aus, um zu überprüfen, ob Ihre Version von Linux bereits Python und `pip` enthält. Wir empfehlen Ihnen, den `pip3`-Befehl zu verwenden, wenn Sie Python-Version 3 oder höher installiert haben.

- Laden Sie das Installationsskript mit dem Befehl `curl` herunter. Der folgende Befehl verwendet den Parameter `-O` (Großbuchstabe „O“), um anzugeben, dass die heruntergeladene Datei im

aktuellen Ordner unter Verwendung desselben Namens auf dem Remote-Host gespeichert werden soll.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

2. Führen Sie das Skript mit dem `python`- oder `python3`-Befehl aus, um die aktuelle Version von `pip` und andere erforderliche Supportpakete herunterzuladen. Wenn Sie den Schalter `--user` einschließen, installiert das Skript `pip` im Pfad `~/local/bin`.

```
$ python3 get-pip.py --user
```

Installieren und aktualisieren Sie das AWS CLI mit Pip

1. Verwenden Sie den `pip`- oder `pip3`-Befehl, um die AWS CLI zu installieren. Wir empfehlen Ihnen, bei Verwendung von Python-Version 3 oder höher den `pip3`-Befehl zu verwenden.

Verwenden Sie für die neueste Version von den AWS CLI folgenden Befehlsblock:

```
$ pip3 install awscli --upgrade --user
```

Hängen Sie für eine bestimmte Version von zwei Gleichheitszeichen `=` und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version `1.16.312` wäre `==1.16.312` was zu dem folgenden Befehl führt:

```
$ pip3 install awscli==1.16.312 --upgrade --user
```

Note

Verwenden Sie geeignete Anführungszeichenregeln für Ihr Terminal. Um das Zeichen `=` verwenden zu können, müssen Sie möglicherweise einfache oder doppelte Anführungszeichen zur korrekten Maskierung verwenden. Im folgenden Beispiel werden einfache Anführungszeichen zur Maskierung verwendet:

```
$ pip3 install 'awscli==1.16.312' --upgrade --user
```

2. Stellen Sie sicher, dass der AWS CLI korrekt installiert ist.

```
$ aws --version
aws-cli/1.33.33 Python/3.11.6 Darwin/23.3.0 botocore/1.18.6
```

Wenn das Programm nicht gefunden wird, [fügen Sie es dem Befehlszeilenpfad hinzu](#).

Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem macOS-Befehlszeilenpfad hinzu

Nach der Installation mit `pip` müssen Sie möglicherweise das Programm `aws` zur `PATH`-Umgebungsvariablen des Betriebssystems hinzufügen. Der Speicherort des Programms hängt davon ab, wo Python installiert ist.

Example AWS CLI Installationsort — macOS mit Python 3.6 und **pip** (Benutzermodus)

```
~/Library/Python/3.7/bin
```

Geben Sie anstelle der Version in dem Beispiel oben Ihre Python-Version an.

Wenn Sie nicht wissen, wo Python installiert ist, führen Sie `which python` aus

```
$ which python
/usr/local/bin/python
```

Die Ausgabe ist möglicherweise der Pfad zu einem Symlink, nicht das tatsächliche Programm. Führen Sie `ls -al` aus, um festzustellen, wohin dieser verweist.

```
$ ls -al /usr/local/bin/python
~/Library/Python/3.7/bin/python3.7
```

`pip` installiert Programme in demselben Ordner, der auch die Python-Anwendung enthält. Fügen Sie diesen Ordner Ihrer `PATH`-Variablen hinzu.

So ändern Sie die Variable **PATH**

1. Suchen Sie das Profilskript der Shell in Ihrem Benutzerverzeichnis. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~
```

```
. .. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`

2. Fügen Sie dem Profilskript einen Exportbefehl hinzu.

```
export PATH=~/.local/bin:$PATH
```

Dieser Befehl fügt der aktuellen PATH-Variablen einen Pfad, in diesem Beispiel `~/.local/bin`, hinzu.

3. Laden Sie hochgeladene Profil in Ihre aktuelle Sitzung.

```
$ source ~/.bash_profile
```

Deinstalliere das AWS CLI mit Pip

1. Wenn Sie die AWS CLI Version 1 mit installiert haben `pip`, müssen Sie auch mit `pip` deinstallieren.

```
$ pip uninstall awscli
```

Wenn Sie Version Python 2 oder 3 verwenden, müssen Sie möglicherweise den Befehl `pip2` oder `pip3` verwenden. Verwenden Sie den `aws --version` Befehl, um die Python-Version zu ermitteln, die Ihrer installierten AWS CLI Version 1 zugeordnet ist.

```
$ pip3 uninstall awscli
```

Möglicherweise müssen Sie das Eingabeaufforderungsfenster oder den Computer neu starten, um alle Dateien zu entfernen.

2. (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

⚠ Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `~/.aws/`. Wenn Ihr Benutzer Schreibrechte für dieses Verzeichnis hat, müssen Sie es nicht verwenden `sudo`.

```
$ sudo rm ~/.aws/
```

Behebung von AWS CLI Installations- und Deinstallationsfehlern

Wenn Sie nach der Installation oder Deinstallation von auf Probleme stoßen AWS CLI, finden Sie die Schritte [Beheben von Fehlern](#) zur Fehlerbehebung unter. Die wichtigsten Maßnahmen zur Fehlerbehebung finden Sie unter [the section called “Fehler aufgrund eines nicht gefundenen Befehls”](#), [the section called “Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück”](#) und [the section called “Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI”](#).

Installation, Aktualisierung und Deinstallation der AWS CLI Version 1 unter Windows

Sie können Version 1 von AWS Command Line Interface (AWS CLI) unter Windows mit einem eigenständigen Installationsprogramm (empfohlen) oder `pip`, einem Paketmanager für Python, installieren.

Wenn Sie einen Befehl eingeben, lassen Sie das Eingabeaufforderungssymbol (`C:\>`) weg. Diese dienen in Programm-Listings dazu, die Befehle, die Sie eingeben, von der Ausgabe zu unterscheiden, die von der AWS CLI zurückgegeben wird. Im restlichen Teil dieses Handbuchs wird das allgemeine Eingabeaufforderungssymbol (`$`) verwendet, außer in Fällen, in denen ein Befehl Windows-spezifisch ist.

Themen

- [Installieren, aktualisieren und deinstallieren Sie AWS CLI Version 1 mithilfe des MSI Installationsprogramms](#)
- [Installieren, aktualisieren und deinstallieren Sie die AWS CLI Version 1 mit Python und Pip unter Windows](#)
- [Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem Befehlszeilenpfad hinzu](#)
- [Behebung von AWS CLI Installations- und Deinstallationsfehlern](#)

Installieren, aktualisieren und deinstallieren Sie AWS CLI Version 1 mithilfe des MSI Installationsprogramms

Die AWS CLI Version 1 wird unter Windows XP oder höher unterstützt. Für Windows-Benutzer bietet das MSI Installationspaket eine vertraute und bequeme Möglichkeit, die AWS CLI Version 1 zu installieren, ohne weitere Voraussetzungen zu installieren.

Installieren und aktualisieren Sie die AWS CLI Version 1 mithilfe des MSI Installationsprogramms

Auf der Seite [Veröffentlichungen](#) GitHub können Sie nachsehen, wann die neueste Version veröffentlicht wurde. Wenn Aktualisierungen veröffentlicht werden, müssen Sie die Installation wiederholen, um die aktuelle Version des AWS CLI Version 1 zu erhalten.

1. Laden Sie das entsprechende MSI Installationsprogramm herunter:
 - AWS CLI MSI Installationsprogramm für Windows (64-Bit): <https://s3.amazonaws.com/aws-cli/AWSCLI64PY3.msi>
 - AWS CLI MSI Installationsprogramm für Windows (32-Bit): <https://s3.amazonaws.com/aws-cli/AWSCLI32PY3>
 - AWS CLI kombinierte Setup-Datei für Windows: <https://s3.amazonaws.com/aws-cli/AWSCLISetup.exe> (enthält sowohl das 32-Bit- als auch das MSI 64-Bit-Installationsprogramm und installiert automatisch die richtige Version)
2. Führen Sie das heruntergeladene MSI Installationsprogramm oder die Setup-Datei aus.
3. Folgen Sie den Anweisungen auf dem Bildschirm. Standardmäßig wird AWS CLI Version 1 auf C:\Program Files\Amazon\AWSCLI (64-Bit-Version) oder C:\Program Files (x86)\Amazon\AWSCLI (32-Bit-Version) installiert.

4. Zum Bestätigen der Installation verwenden Sie in einer Eingabeaufforderung den Befehl `aws --version` (Öffnen Sie das Start-Menü und suchen Sie nach `cmd`, um eine Eingabeaufforderung zu öffnen).

```
C:\> aws --version
aws-cli/1.33.33 Python/3.11.6 Windows/10 botocore/1.18.6
```

Wenn Windows das Programm nicht finden kann, müssen Sie möglicherweise die Befehlszeile schließen und erneut öffnen, um den Pfad zu aktualisieren, oder [das Installationsverzeichnis manuell zu Ihrer PATH Umgebungsvariablen hinzufügen](#).

Deinstallieren Sie die AWS CLI Version 1

Um die folgenden Deinstallationsanweisungen verwenden zu können, müssen Sie AWS CLI Version 1 mit der MSI Installations- oder Setup-Datei installiert haben.

1. Öffnen Sie Programme und Funktionen indem Sie einen der folgenden Schritte ausführen:
 - Öffnen Sie die Systemsteuerung und wählen Sie dann Programme und Funktionen aus.
 - Öffnen Sie eine Eingabeaufforderung und geben Sie dann den folgenden Befehl ein.

```
C:\> appwiz.cpl
```

2. Wählen Sie den Eintrag namens AWS Command Line Interface aus und wählen Sie dann Deinstallieren aus, um das Deinstallationsprogramm zu starten.
3. Bestätigen Sie, dass Sie das deinstallieren möchten AWS CLI.
4. (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `%UserProfile%\aws`.

```
$ rmdir %UserProfile%\aws
```

Installieren, aktualisieren und deinstallieren Sie die AWS CLI Version 1 mit Python und Pip unter Windows

Die Python Software Foundation bietet Installationsprogramme für Windows, die `pip` beinhalten.

Voraussetzungen

Sie müssen auch Python 3.8 oder höher installieren. Installationsanweisungen finden Sie auf der Seite „[Python herunterladen](#)“ im Python-Anfängerhandbuch.

Installieren und aktualisieren Sie die AWS CLI Version 1 mit Pip

1. Um die AWS CLI Version 1 zu installieren, verwenden Sie den `pip3` Befehl (wenn Sie Python Version 3 oder höher verwenden) oder den `pip` Befehl.

Verwenden Sie für die AWS CLI neueste Version von den folgenden Befehlsblock:

```
C:\> pip3 install awscli --upgrade --user
```

Hängen Sie für eine bestimmte Version von ein Kleiner-als-Zeichen `<` und die Versionsnummer an den Dateinamen an. AWS CLI In diesem Beispiel der Dateiname für Version `1.16.312` wäre `<1.16.312` was zu dem folgenden Befehl führt:

```
C:\> pip3 install awscli<1.16.312 --upgrade --user
```

2. Stellen Sie sicher, dass AWS CLI Version 1 korrekt installiert ist. Wenn keine Antwort erfolgt, lesen Sie in Abschnitt [Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem Befehlszeilenpfad hinzu](#) nach.

```
C:\> aws --version  
aws-cli/1.33.33 Python/3.11.6 Windows/10 botocore/1.18.6
```

Deinstallieren Sie die AWS CLI Version 1 mit pip

1. Wenn Sie AWS CLI Version 1 mit installiert haben `pip`, müssen Sie auch mit `pip` deinstallieren.

```
C:\> pip uninstall awscli
```

Wenn Sie Version Python 2 oder 3 verwenden, müssen Sie möglicherweise den Befehl `pip2` oder `pip3` verwenden. Verwenden Sie den `aws --version` Befehl, um die Python-Version zu ermitteln, die Ihrer installierten AWS CLI Version 1 zugeordnet ist.

```
C:\> pip3 uninstall awscli
```

Möglicherweise müssen Sie das Eingabeaufforderungsfenster oder den Computer neu starten, um alle Dateien zu entfernen.

2. (Optional) Entfernen Sie die geteilten Informationen AWS SDK und die AWS CLI Einstellungsinformationen aus dem `.aws` Ordner.

Warning

Diese Einstellungen für Konfiguration und Anmeldeinformationen werden von allen AWS SDKs gemeinsam genutzt AWS CLI. Wenn Sie diesen Ordner entfernen, können keine Benutzer mehr darauf zugreifen AWS SDKs, die sich noch auf Ihrem System befinden.

Der Standardspeicherort des `.aws` Ordners ist je nach Plattform unterschiedlich. Standardmäßig befindet sich der Ordner in `%UserProfile%\aws`.

```
$ rmdir %UserProfile%\aws
```

Fügen Sie die ausführbare AWS CLI Version 1 zu Ihrem Befehlszeilenpfad hinzu

Nachdem Sie AWS CLI Version 1 mit installiert haben `pip`, fügen Sie das `aws` Programm der `PATH` Umgebungsvariablen Ihres Betriebssystems hinzu. Bei einer MSI Installation sollte dies automatisch geschehen. Wenn der `aws`-Befehl jedoch nach der Installation nicht ausgeführt wird, müssen Sie ihn möglicherweise manuell festlegen.

1. Verwenden Sie den `where`-Befehl, um den Dateispeicherort von `aws` zu finden. Der `where`-Befehl zeigt standardmäßig an, wo sich ein bestimmtes Programm im PATH des Systems befindet.

```
C:\> where aws
```

Welche Pfade angezeigt werden, hängt von Ihrer Plattform und der Methode ab, mit der Sie die AWS CLI installiert haben. Ordnernamen mit Versionsnummern, die sich ändern können. Diese Beispiele beziehen sich auf die Verwendung von Python Version 3.7. Ersetzen Sie die Version bei Bedarf durch die von Ihnen verwendete Versionsnummer. Nachfolgend finden Sie Beispiele für typische Pfade:

- Python 3 und **pip3** – `C:\Program Files\Python37\Scripts\`
- Python 3 und **pip3** --Benutzer-Option auf früheren Versionen von Windows – `%USERPROFILE%\AppData\Local\Programs\Python\Python37\Scripts`
- Python 3 und **pip3** --user-Option unter Windows 10 – `%USERPROFILE%\AppData\Roaming\Python\Python37\Scripts`
- MSI-Installer (64-Bit) — `C:\Program Files\Amazon\AWSCLI\bin`
- MSI-Installationsprogramm (32-Bit) — `C:\Program Files (x86)\Amazon\AWSCLI\bin`

Führen Sie die folgenden Schritte aus, je nachdem, ob ein Dateipfad zurückgegeben wird.

A file path is returned

```
C:\> where aws
C:\Program Files\Amazon\AWSCLI\bin\aws.exe
```

Mit dem folgenden Befehl finden Sie heraus, wo das `aws`-Programm installiert ist.

```
C:\> where c:\ aws
C:\Program Files\Python37\Scripts\aws
```

A file path is NOT returned

Wenn der `where`-Befehl den folgenden Fehler zurückgibt, ist er nicht im System-PATH und Sie können ihn nicht ausführen, indem Sie seinen Namen eingeben.

```
C:\> where c:\ aws
INFO: Could not find files for the given pattern(s).
```

Führen Sie in diesem Fall den where-Befehl mit dem Parameter `/R path` aus, um ihn zum Durchsuchen aller Ordner anzuweisen. Fügen Sie den Pfad dann manuell hinzu. Verwenden Sie die Befehlszeile oder den Datei-Explorer, um herauszufinden, wo es auf Ihrem Computer installiert ist.

```
C:\> where /R c:\ aws
c:\Program Files\Amazon\AWSCLI\bin\aws.exe
c:\Program Files\Amazon\AWSCLI\bincompat\aws.cmd
c:\Program Files\Amazon\AWSCLI\runtime\Scripts\aws
c:\Program Files\Amazon\AWSCLI\runtime\Scripts\aws.cmd
...
```

2. Betätigen Sie die Windows-Taste und geben Sie **environment variables** ein.
3. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
4. Wählen PATHS Sie und dann Bearbeiten.
5. Fügen Sie den gefundenen Pfad in das Feld Variablenwert ein, z. B. ***C:\Program Files\Amazon\AWSCLI\bin\aws.exe***.
6. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
7. Schließen Sie alle laufenden Eingabeaufforderungen und öffnen Sie das Eingabeaufforderungsfenster erneut.

Behebung von AWS CLI Installations- und Deinstallationsfehlern

Wenn Sie nach der Installation oder Deinstallation von auf Probleme stoßen AWS CLI, finden Sie die Schritte [Beheben von Fehlern](#) zur Fehlerbehebung unter. Die wichtigsten Maßnahmen zur Fehlerbehebung finden Sie unter [the section called "Fehler aufgrund eines nicht gefundenen Befehls"](#), [the section called "Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück"](#) und [the section called "Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI"](#).

Installation und Aktualisierung der AWS CLI Version 1 in einer virtuellen Umgebung

Sie können Versionskonflikte zwischen Anforderungen und anderen pip Paketen vermeiden, indem Sie Version 1 von AWS Command Line Interface (AWS CLI) in einer virtuellen Umgebung installieren.

Themen

- [Voraussetzungen](#)
- [Installieren und aktualisieren Sie die AWS CLI Version 1 in einer virtuellen Umgebung](#)
- [Behebung von AWS CLI Installations- und Deinstallationsfehlern](#)

Voraussetzungen

- Python 3.8 oder höher. Installationsanweisungen finden Sie auf der Seite „[Python herunterladen](#)“ im Python-Anfängerhandbuch.

Matrix zur Unterstützung der Python-Version

AWS CLI Version	Unterstützte Python-Versionen
1.32.0 bis aktuell	Python 3.8 oder höher
1.27.0 bis 1.31.x	Python 3.7+
1.20.0 bis 1.26.x	Python 3.6+
1.19.0 bis 1.19.x	Python 2.7+ oder Python 3.6+
1.17 - 1.18.x	Python 2.7+, Python 3.4+
1.0 - 1.16.x	Python 2.6 und älter, Python 3.3 und älter

- pip oder pip3 ist installiert.

Installieren und aktualisieren Sie die AWS CLI Version 1 in einer virtuellen Umgebung

1. Installieren Sie `virtualenv` mit `pip`.

```
$ pip install --user virtualenv
```

2. Erstellen Sie eine virtuelle Umgebung und benennen Sie sie.

```
$ virtualenv ~/cli-ve
```

Alternativ können Sie die Option `-p` verwenden, um eine andere als die Standardversion von Python anzugeben.

```
$ virtualenv -p /usr/bin/python37 ~/cli-ve
```

3. Aktivieren Sie die neue virtuelle Umgebung.

Linux oder macOS

```
$ source ~/cli-ve/bin/activate
```

Windows

```
$ %USERPROFILE%\cli-ve\Scripts\activate
```

Die Eingabeaufforderung ändert sich und zeigt so an, dass Ihre virtuelle Umgebung aktiv ist.

```
(cli-ve)~$
```

4. Installieren oder aktualisieren Sie AWS CLI Version 1 in Ihrer virtuellen Umgebung.

```
(cli-ve)~$ pip install --upgrade awscli
```

5. Stellen Sie sicher, dass AWS CLI Version 1 korrekt installiert ist.

```
$ aws --version  
aws-cli/1.33.33 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

6. Sie können den Befehl `deactivate` verwenden, um die virtuelle Umgebung zu beenden. Wenn Sie eine neue Sitzung starten, müssen Sie die Umgebung erneut aktivieren.

Behebung von AWS CLI Installations- und Deinstallationsfehlern

Wenn Sie nach der Installation oder Deinstallation von auf Probleme stoßen AWS CLI, finden Sie die Schritte [Beheben von Fehlern](#) zur Fehlerbehebung unter. Die wichtigsten Maßnahmen zur Fehlerbehebung finden Sie unter [the section called "Fehler aufgrund eines nicht gefundenen Befehls"](#), [the section called "Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück"](#) und [the section called "Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI"](#).

Konfiguration der Einstellungen für den AWS CLI

In diesem Abschnitt wird erklärt, wie Sie die Einstellungen konfigurieren, mit denen AWS Command Line Interface (AWS CLI) interagiert AWS. Diese umfassen u. a. folgende:

- Anmeldeinformationen identifizieren, wer den anruftAPI. Zugangsdaten werden verwendet, um die Anfrage an die AWS Server zu verschlüsseln, um Ihre Identität zu bestätigen und die zugehörigen Berechtigungsrichtlinien abzurufen. Diese Berechtigungen bestimmen, welche Aktionen Sie ausführen können. Weitere Informationen zum Einrichten Ihrer Anmeldeinformationen finden Sie unter [Authentifizierung und Anmeldeinformationen](#).
- Weitere Konfigurationsdetails, die festlegen, AWS CLI wie Anfragen verarbeitet werden sollen, z. B. das Standardausgabeformat und die AWS Standardregion.

Note

AWS erfordert, dass alle eingehenden Anfragen kryptografisch signiert sind. Das AWS CLI erledigt das für Sie. Die „Signatur“ beinhaltet eine vom AWS Service date/time stamp. Therefore, you must ensure that your computer's date and time are set correctly. If you don't, and the date/time in the signature is too far off of the date/time erkannte, AWS lehnt die Anfrage ab.

Vorrang der Konfiguration und der Anmeldeinformationen

Anmeldeinformationen und Konfigurationseinstellungen befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden in der Befehlszeile explizit als Parameter deklariert. Bestimmte Speicherorte haben Vorrang vor anderen. Die AWS CLI Anmeldeinformationen und Konfigurationseinstellungen haben in der folgenden Reihenfolge Vorrang:

1. [Befehlszeilenoptionen](#) – überschreiben Einstellungen an jedem anderen Speicherort, z. B. die Parameter `--region`, `--output` und `--profile`.
2. [Umgebungsvariablen](#) – Sie können Werte in den Umgebungsvariablen Ihres Systems speichern.
3. [Rolle übernehmen](#) — Übernehmen Sie die Berechtigungen einer IAM Rolle über die Konfiguration oder den `aws sts assume-role` Befehl.

4. [Rolle mit Webidentität](#) übernehmen — Übernehmen Sie die Rechte einer IAM Rolle, die die Webidentität über die Konfiguration oder den `aws sts assume-role` Befehl verwendet.
5. [Anmeldeinformationsdatei](#) – die Dateien `credentials` und `config` werden aktualisiert, wenn Sie den Befehl `aws configure` ausführen. Die Datei `credentials` befindet sich in `~/.aws/credentials` unter Linux und in macOS oder in `C:\Users\USERNAME\.aws\credentials` unter Windows.
6. [Benutzerdefinierter Prozess](#) – rufen Sie Ihre Anmeldeinformationen von einer externen Quelle ab.
7. [Konfigurationsdatei](#) – die Dateien `credentials` und `config` werden aktualisiert, wenn Sie den Befehl `aws configure` ausführen. Die Datei `config` befindet sich in `~/.aws/config` unter Linux und in macOS oder in `C:\Users\USERNAME\.aws\config` unter Windows.
8. [Container-Anmeldeinformationen](#) — Sie können jeder Ihrer Amazon Elastic Container Service (AmazonECS) -Aufgabendefinitionen eine IAM Rolle zuordnen. Temporäre Anmeldeinformationen für diese Rolle stehen dann für die Container dieser Aufgabe zur Verfügung. Weitere Informationen finden Sie unter [IAMRollen für Aufgaben](#) im Amazon Elastic Container Service Developer Guide.
9. [Anmeldeinformationen für das EC2 Amazon-Instance-Profil](#) — Sie können jeder Ihrer Amazon Elastic Compute Cloud (AmazonEC2) -Instances eine IAM Rolle zuordnen. Temporäre Anmeldeinformationen für diese Rolle stehen dann für den Code zur Verfügung, der in dieser Instance ausgeführt wird. Die Anmeldeinformationen werden über den EC2 Amazon-Metadatenservice bereitgestellt. Weitere Informationen finden Sie unter [IAMRollen für Amazon EC2](#) im EC2Amazon-Benutzerhandbuch und [Verwenden von Instanzprofilen](#) im IAMBenutzerhandbuch.

Weitere Themen in diesem Abschnitt

- [the section called “Einstellungen für die Konfiguration und die Anmeldeinformationsdatei in der AWS CLI”](#)
- [the section called “Umgebungsvariablen”](#)
- [the section called “Befehlszeilenoptionen in der AWS CLI”](#)
- [the section called “Konfiguration der Befehlsvervollständigung in AWS CLI”](#)
- [the section called “Wiederholversuche”](#)
- [the section called “Verwendung eines HTTP Proxys für AWS CLI”](#)

Einstellungen für die Konfiguration und die Anmeldeinformationsdatei in der AWS CLI

Sie können Ihre häufig verwendeten Konfigurationseinstellungen und Anmeldeinformationen in Dateien speichern, die von der AWS CLI verwaltet werden.

Die Dateien sind in `profiles` unterteilt. Standardmäßig AWS CLI verwendet die Einstellungen des genannten Profils `default`. Wenn Sie alternative Einstellungen verwenden möchten, können Sie zusätzliche Profile erstellen und referenzieren.

Sie können eine einzelne Einstellung überschreiben, indem Sie entweder eine der unterstützten Umgebungsvariablen definieren oder einen Befehlszeilenparameter verwenden. Weitere Informationen zum Vorrang der Konfigurationseinstellungen finden Sie unter [Konfiguration der Einstellungen für den AWS CLI](#).

Note

Weitere Informationen zum Einrichten Ihrer Anmeldeinformationen finden Sie unter [Authentifizierung und Anmeldeinformationen](#).

Themen

- [Formatieren der Konfigurations- und Anmeldeinformationsdateien](#)
- [Wo werden Konfigurationseinstellungen gespeichert?](#)
- [Verwenden von benannten Profilen](#)
- [Festlegen und Anzeigen von Konfigurationseinstellungen mithilfe von Befehlen](#)
- [Befehlsbeispiele für das Festlegen neuer Konfigurationen und Anmeldeinformationen](#)
- [Unterstützte Einstellungen in der config-Datei](#)

Formatieren der Konfigurations- und Anmeldeinformationsdateien

Die Dateien `config` und `credentials` sind in Abschnitte unterteilt. Die Abschnitte lauten `profiles` und `services`. Ein Abschnitt ist eine benannte Sammlung von Einstellungen und reicht bis zur nächsten Abschnittsdefinitionszeile. Mehrere Profile und Abschnitte können in den `config`- und `credentials`-Dateien gespeichert werden.

Diese Dateien sind Klartextdateien, die folgendes Format verwenden:

- Abschnittsnamen werden in eckige Klammern [] eingeschlossen, z. B. [default], [profile *user1*] und [sso-session].
- Alle Einträge in einem Abschnitt haben das allgemeine Format `setting_name=value`.
- Zeilen können auskommentiert werden, indem die Zeile mit einem Hash-Zeichen (#) begonnen wird.

Die Dateien config und credentials enthalten die folgenden Abschnittstypen:

- [Abschnittstyp: profile](#)
- [Abschnittstyp: services](#)

Abschnittstyp: **profile**

Die AWS CLI Geschäfte

Je nach Datei verwenden Profilabschnittsnamen folgendes Format:

- Konfigurationsdatei: [default] [profile *user1*]
- Anmeldeinformationsdatei: [default] [*user1*]

Verwenden Sie beim Erstellen eines Eintrags in der credentials-Datei nicht das Wort profile.

Jedes Profil kann unterschiedliche Anmeldeinformationen und auch verschiedene AWS –Regionen und Ausgabeformate angeben. Fügen Sie beim Benennen des Profils in einer config-Datei das Präfixwort "profile," ein. Dieses darf jedoch nicht in der credentials-Datei enthalten sein.

Die folgenden Beispiele zeigen eine credentials- und eine config-Datei mit zwei angegebenen Profilen, Region und Ausgabe. Die erste Datei [default] wird verwendet, wenn Sie einen AWS CLI - Befehl ohne Profil ausführen. Die zweite wird verwendet, wenn Sie einen AWS CLI Befehl mit dem --profile user1 Parameter ausführen.

Short-term credentials

Dieses Beispiel gilt für die kurzfristigen Anmeldeinformationen von AWS Identity and Access Management. Weitere Informationen finden Sie unter [the section called "Kurzfristige Anmeldeinformationen"](#).

Konfigurationsdatei

```
[default]
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
source_profile=default
role_session_name=session_user1
region=us-east-1
output=text
```

Amazon EC2 instance metadata credentials

Dieses Beispiel bezieht sich auf die Anmeldeinformationen, die aus den Metadaten der EC2 Hosting-Amazon-Instance abgerufen wurden. Weitere Informationen finden Sie unter [the section called “Verwendung von EC2 Amazon-Instance-Metadaten als Anmeldeinformationen in der AWS CLI”](#).

Anmeldeinformationsdatei

Die `credentials`-Datei wird nicht für diese Authentifizierungsmethode verwendet.

Konfigurationsdatei

```
[default]
role_arn=arn:aws:iam::123456789012:role/defaultrole
credential_source=Ec2InstanceMetadata
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
credential_source=Ec2InstanceMetadata
region=us-east-1
output=text
```

Long-term credentials

Warning

Verwenden Sie zur Vermeidung von Sicherheitsrisiken keine IAM Benutzer zur Authentifizierung, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie [AWS IAM Identity Center](#).

Dieses Beispiel bezieht sich auf die langfristigen Anmeldeinformationen von AWS Identity and Access Management. Weitere Informationen finden Sie unter [the section called “IAMBenutzer”](#).

Anmeldeinformationsdatei

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

[user1]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

Konfigurationsdatei

```
[default]
region=us-west-2
output=json

[profile user1]
region=us-east-1
output=text
```

Weitere Informationen und zusätzliche Autorisierungs- und Anmeldeinformationsmethoden finden Sie unter [the section called “IAMBenutzer”](#).

Abschnittstyp: **services**

Bei `services` diesem Abschnitt handelt es sich um eine Gruppe von Einstellungen, mit denen benutzerdefinierte Endpunkte für Anfragen konfiguriert werden. AWS-Service Ein Profil wird dann mit einem `services`-Abschnitt verknüpft.

```
[profile dev]
services = my-services
```

Der `services`-Abschnitt ist durch `<SERVICE> =` -Zeilen in Unterabschnitte unterteilt, wobei `<SERVICE>` der AWS-Service -ID-Schlüssel ist. Der AWS-Service Bezeichner basiert auf dem API Modell, indem alle Leerzeichen `serviceId` durch Unterstriche ersetzt und alle Buchstaben klein geschrieben werden. Eine Liste aller Service-ID-Schlüssel, die im `services`-Abschnitt verwendet werden können, finden Sie unter [Verwenden von Endpunkten in der AWS CLI](#). Auf den Service-ID-Schlüssel folgen verschachtelte Einstellungen, die jeweils in einer eigenen Zeile stehen, welche durch zwei Leerzeichen eingerückt sind.

Im folgenden Beispiel wird der Endpunkt so konfiguriert, dass er für Anfragen an den Amazon DynamoDB Dienst verwendet wird `my-services` Abschnitt, der verwendet wird in `dev` Profil. Alle unmittelbar folgenden Zeilen, die eingerückt sind, sind in diesem Unterabschnitt enthalten und gelten für diesen Service.

```
[profile dev]
services = my-services

[services my-services]
dynamodb =
  endpoint_url = http://localhost:8000
```

Weitere Informationen zu servicespezifischen Endpunkten finden Sie unter [Verwenden von Endpunkten in der AWS CLI](#).

Wenn Ihr Profil über rollenbasierte Anmeldeinformationen verfügt, die über einen `source_profile` Parameter für die Funktion „Rolle IAM übernehmen“ konfiguriert wurden, werden SDK nur Dienstkonfigurationen für das angegebene Profil verwendet. Es verwendet keine Profile mit verketteten Rollen. Verwenden Sie beispielsweise die folgende freigegebene `config`-Datei:

```
[profile A]
credential_source = Ec2InstanceMetadata
```



```
endpoint_url = https://profile-a-endpoint.aws/

[profile B]
source_profile = A
role_arn = arn:aws:iam::123456789012:role/roleB
services = profileB

[services profileB]
ec2 =
    endpoint_url = https://profile-b-ec2-endpoint.aws
```

Wenn Sie das Profil verwenden B und in Ihrem Code Amazon aufrufen EC2, wird der Endpunkt als `https://profile-b-ec2-endpoint.aws` aufgelöst. Wenn Ihr Code eine Anforderung für einen anderen Service stellt, folgt die Endpunktauflösung keiner benutzerdefinierten Logik. Der Endpunkt wird nicht zu dem im Profil A definierten globalen Endpunkt aufgelöst. Damit ein globaler Endpunkt für das Profil B wirksam wird, müssten Sie `endpoint_url` direkt im Profil B festlegen.

Wo werden Konfigurationseinstellungen gespeichert?

Der AWS CLI speichert vertrauliche Anmeldeinformationen, die Sie mit `aws configure` in einer lokalen Datei mit dem Namen `credentials`, in einem Ordner mit dem Namen `.aws` in Ihrem Home-Verzeichnis. Die weniger vertraulichen Konfigurationsoptionen, die Sie mit `aws configure` angeben, werden in einer lokalen Datei namens `config` gespeichert, die sich ebenfalls im Ordner `.aws` Ihres Stammverzeichnisses befindet.

Speichern von Anmeldeinformationen in der Konfigurationsdatei

Sie können alle Ihre Profileinstellungen in einer einzigen Datei speichern, da sie Anmeldeinformationen aus der `config` Datei lesen AWS CLI können. Wenn in beiden Dateien Anmeldeinformationen für ein Profil mit demselben Namen vorhanden sind, haben die Schlüssel in der Anmeldeinformationsdatei Vorrang. Wir empfehlen, die Anmeldeinformationen in den `credentials`-Dateien zu speichern. Diese Dateien werden auch von den verschiedenen Software-Entwicklungskits für Sprachen verwendet (SDKs). Wenn Sie zusätzlich zu den eine SDKs der Optionen verwenden AWS CLI, überprüfen Sie, ob die Anmeldeinformationen in einer eigenen Datei gespeichert werden sollen.

Der Speicherort Ihres Stammverzeichnisses hängt vom Betriebssystem ab. Die folgenden Umgebungsvariablen verweisen auf dieses: `%UserProfile%` in Windows und `$HOME` oder

~ (Tilde) in Unix-basierten Systemen. Sie können auch einen Nicht-Standard-Speicherort für die Dateien angeben, indem Sie die Umgebungsvariablen `AWS_CONFIG_FILE` und `AWS_SHARED_CREDENTIALS_FILE` auf einen anderen lokalen Pfad festlegen. Details dazu finden Sie unter [Konfiguration von Umgebungsvariablen für AWS CLI](#).

Wenn Sie ein gemeinsam genutztes Profil verwenden, das eine Rolle AWS Identity and Access Management (IAM) angibt, wird der AWS STS AssumeRole Vorgang zum Abrufen temporärer Anmeldeinformationen AWS CLI aufgerufen. Diese Anmeldeinformationen werden dann (in `~/.aws/cli/cache`) gespeichert. Nachfolgende AWS CLI Befehle verwenden die zwischengespeicherten temporären Anmeldeinformationen, bis sie ablaufen, und an diesem Punkt AWS CLI werden die Anmeldeinformationen automatisch aktualisiert.

Verwenden von benannten Profilen

Wenn kein Profil explizit definiert ist, wird das default-Profil verwendet.

Um ein benanntes Profil zu verwenden, fügen Sie dem Befehl die Option `--profile profile-name` hinzu. Das folgende Beispiel listet alle Ihre EC2 Amazon-Instances mit den im `user1` Profil definierten Anmeldeinformationen und Einstellungen auf.

```
$ aws ec2 describe-instances --profile user1
```

Wenn ein benanntes Profil für mehrere Befehle verwendet werden soll, müssen Sie das Profil nicht in jedem Befehl angeben. Legen Sie stattdessen die Umgebungsvariable `AWS_PROFILE` als Standardprofil fest. Sie können diese Einstellung mithilfe des `--profile`-Parameters außer Kraft setzen.

Linux or macOS

```
$ export AWS_PROFILE=user1
```

Windows

```
C:\> setx AWS_PROFILE user1
```

Bei Verwendung von [set](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert bis zum Ende der aktuellen Eingabeaufforderungssitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert.

Wenn Sie [setx](#) zum Festlegen einer Umgebungsvariable verwenden, ändert sich der Wert in allen Befehls-Shells, die Sie nach der Ausführung des Befehls erstellen. Befehls-Shells, die zum Zeitpunkt der Befehlsausführung bereits ausgeführt werden, sind nicht betroffen. Schließen Sie die Befehls-Shell und starten Sie sie neu, um die Auswirkungen der Änderung zu sehen.

Durch die Festlegung der Umgebungsvariablen wird das Standardprofil bis zum Ende der Shell-Sitzung geändert oder bis Sie einen anderen Wert für die Variable bestimmen. Sie können Umgebungsvariablen für zukünftige Sitzungen persistent machen, indem Sie sie in das Startup-Skript Ihrer Shell stellen. Weitere Informationen finden Sie unter [Konfiguration von Umgebungsvariablen für AWS CLI](#).

Festlegen und Anzeigen von Konfigurationseinstellungen mithilfe von Befehlen

Es gibt mehrere Möglichkeiten, Ihre Konfigurationseinstellungen mithilfe von Befehlen anzuzeigen und festzulegen.

[aws configure](#)

Führen Sie diesen Befehl aus, um Ihre -Anmeldeinformationen, Ihre Region und das Ausgabeformat schnell festzulegen und anzuzeigen. Das folgende Beispiel zeigt Beispielwerte.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFicYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

[aws configure set](#)

Sie können alle Anmeldeinformationen oder Konfigurationseinstellungen mit `aws configure set` festlegen. Geben Sie das Profil an, das Sie mit der `--profile`-Einstellung anzeigen oder ändern möchten.

Mit dem folgenden Befehl beispielsweise wird die `region` in dem Profil mit dem Namen `integ` festgelegt.

```
$ aws configure set region us-west-2 --profile integ
```

Um eine Einstellung zu entfernen, löschen Sie die Einstellung in Ihren `credentials` Dateien `config` und in einem Texteditor manuell.

[aws configure get](#)

Sie können alle Anmeldeinformationen oder Konfigurationseinstellungen abrufen, die Sie mit `aws configure get` festgelegt haben. Geben Sie das Profil an, das Sie mit der `--profile`-Einstellung anzeigen oder ändern möchten.

Beispielsweise wird mit dem folgenden Befehl die `region`-Einstellung in dem Profil mit dem Namen `integ` abgerufen.

```
$ aws configure get region --profile integ
us-west-2
```

Wenn die Ausgabe leer ist, ist die Einstellung nicht explizit festgelegt und es wird der Standardwert verwendet.

[aws configure list](#)

Um Konfigurationsdaten aufzulisten, verwenden Sie den Befehl `aws configure list`. Dieser Befehl listet das Profil, den Zugriffsschlüssel, den geheimen Schlüssel und die Regionskonfigurationsinformationen auf, die für das angegebene Profil verwendet werden. Für jedes Konfigurationselement werden der Wert, der Ort, an dem der Konfigurationswert abgerufen wurde, und der Name der Konfigurationsvariablen angezeigt.

Wenn Sie beispielsweise die AWS-Region in einer Umgebungsvariablen angeben, zeigt Ihnen dieser Befehl den Namen der Region, die Sie konfiguriert haben, dass dieser Wert aus einer Umgebungsvariablen stammt, und den Namen der Umgebungsvariablen.

Bei Methoden mit temporären Anmeldeinformationen wie Rollen und IAM Identity Center zeigt dieser Befehl den temporär zwischengespeicherten Zugriffsschlüssel an, und der geheime Zugriffsschlüssel wird angezeigt.

```
$ aws configure list
  Name                Value                Type    Location
  ----                -
  profile              <not set>           None    None
  access_key          *****ABCD         shared-credentials-file
  secret_key          *****ABCD         shared-credentials-file
  region              us-west-2           env     AWS_DEFAULT_REGION
```


Amazon EC2 instance metadata credentials

Dieses Beispiel bezieht sich auf die Anmeldeinformationen, die aus den Metadaten der EC2 Hosting-Amazon-Instance abgerufen wurden. Für diesen Vorgang gibt es keinen Assistenten. Daher wird jeder Wert mit dem Befehl `aws configure set` festgelegt. Weitere Informationen finden Sie unter [the section called “Verwendung von EC2 Amazon-Instance-Metadaten als Anmeldeinformationen in der AWS CLI”](#).

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
$ aws configure set credential_source Ec2InstanceMetadata
$ aws configure set region us-west-2
$ aws configure set output json
```

Long-term credentials

Warning

Verwenden Sie zur Vermeidung von Sicherheitsrisiken keine IAM Benutzer zur Authentifizierung, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie [AWS IAM Identity Center](#).

Dieses Beispiel bezieht sich auf die langfristigen Anmeldeinformationen von AWS Identity and Access Management. Weitere Informationen finden Sie unter [the section called “IAMBenutzer”](#).

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Unterstützte Einstellungen in der `config`-Datei

Themen

- [Globale Einstellungen](#)
- [Einstellungen für benutzerdefinierte S3-Befehle](#)

Die folgenden Einstellungen werden in der `config`-Datei unterstützt. Es werden die Werte in dem angegebenen Profil (oder dem Standardprofil) verwendet, es sei denn, sie werden durch eine gleichnamige Umgebungsvariable oder eine gleichnamige Befehlszeilenoption überschrieben. Weitere Informationen darüber, welche Rangfolgeeinstellungen Vorrang haben, finden Sie unter [Konfiguration der Einstellungen für den AWS CLI](#)

Globale Einstellungen

api_versions

Einige AWS Dienste verfügen über mehrere API Versionen, um die Abwärtskompatibilität zu unterstützen. Standardmäßig verwenden AWS CLI Befehle die neueste verfügbare API Version. Sie können eine API Version angeben, die für ein Profil verwendet werden soll, indem Sie die `api_versions` Einstellung in die `config` Datei aufnehmen.

Dies ist eine „verschachtelte“ Einstellung, auf die eine oder mehrere eingerückte Zeilen folgen, die jeweils einen AWS Dienst und die zu verwendende API Version angeben. In der Dokumentation zu den einzelnen Diensten finden Sie Informationen darüber, welche API Versionen verfügbar sind.

Das folgende Beispiel zeigt, wie eine API Version für zwei AWS Dienste angegeben wird. Diese API Versionen werden nur für Befehle verwendet, die unter dem Profil ausgeführt werden, das diese Einstellungen enthält.

```
api_versions =  
  ec2 = 2015-03-01  
  cloudfront = 2015-09-017
```

Diese Einstellung verfügt nicht über eine Umgebungsvariable oder einen entsprechenden Befehlszeilenparameter.

aws_access_key_id

Gibt den AWS Zugriffsschlüssel an, der als Teil der Anmeldeinformationen zur Authentifizierung der Befehlsanforderung verwendet wird. Zwar kann dieses in der `config`-Datei gespeichert werden, wir empfehlen Ihnen jedoch, es in der `credentials`-Datei zu speichern.

Kann von der Umgebungsvariablen `AWS_ACCESS_KEY_ID` überschrieben werden. Es ist nicht möglich, die Zugriffsschlüssel-ID als Befehlszeilenoption anzugeben.

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

aws_secret_access_key

Gibt den AWS geheimen Schlüssel an, der als Teil der Anmeldeinformationen zur Authentifizierung der Befehlsanforderung verwendet wird. Zwar kann dieses in der config-Datei gespeichert werden, wir empfehlen Ihnen jedoch, es in der credentials-Datei zu speichern.

Kann von der Umgebungsvariablen `AWS_SECRET_ACCESS_KEY` überschrieben werden. Es ist nicht möglich, den geheimen Zugriffsschlüssel als Befehlszeilenoption anzugeben.

```
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

aws_session_token

Gibt ein AWS Sitzungstoken an. Ein Sitzungs-Token ist nur erforderlich, wenn Sie manuell temporäre Anmeldeinformationen angeben. Zwar kann dieses in der config-Datei gespeichert werden, wir empfehlen Ihnen jedoch, es in der credentials-Datei zu speichern.

Kann von der Umgebungsvariablen `AWS_SESSION_TOKEN` überschrieben werden. Es ist nicht möglich, das Sitzungs-Token als Befehlszeilenoption anzugeben.

```
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwnqKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

ca_bundle

Gibt ein CA-Zertifikatspaket (eine Datei mit der .pem Erweiterung) an, das zur Überprüfung von SSL Zertifikaten verwendet wird.

Kann von der Umgebungsvariablen [AWS_CA_BUNDLE](#) oder mit der Befehlszeilenoption [--ca-bundle](#) überschrieben werden.

```
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

cli_follow_urlparam

Gibt an, ob AWS CLI versucht wird, URL Links in Befehlszeilenparametern zu folgen, die mit `http://` oder `beginnenhttps://`. Wenn diese Option aktiviert ist, wird der abgerufene Inhalt anstelle von als Parameterwert verwendetURL.

- `true` (wahr) – Dies ist der Standardwert. Wenn angegeben, werden alle Zeichenfolgeparameter abgerufen, die mit `http://` oder `https://` beginnen, und die heruntergeladenen Inhalte werden als Parameterwert für den Befehl verwendet.
- `false` — Falls angegeben, behandelt der AWS CLI keine Parameterzeichenfolgenwerte, die mit anderen Zeichenfolgen beginnen `http://` `https://` oder sich von diesen unterscheiden.

Zu diesem Eintrag gibt es keine entsprechende Umgebungsvariable oder Befehlszeilenoption.

```
cli_follow_urlparam = false
```

cli_history

Standardmäßig deaktiviert. Diese Einstellung aktiviert den Befehlsverlauf für die AWS CLI. Nach dem Aktivieren dieser Einstellung AWS CLI zeichnet der den Verlauf der `aws` Befehle auf.

```
cli_history = enabled
```

Sie können Ihren Verlauf mit dem Befehl `aws history list` auflisten und die resultierenden `command_ids` im Befehl `aws history show` verwenden, um Details abzurufen. Weitere Informationen finden Sie unter [aws history](#) im AWS CLI -Referenzhandbuch.

cli_timestamp_format

Gibt das Format der in der Ausgabe enthaltenen Zeitstempelwerte an. Sie können einen der folgenden Werte angeben:

- `iso8601` — Der Standardwert für Version 2. AWS CLI [Falls angegeben, werden alle Zeitstempel gemäß 8601 AWS CLI neu formatiert. ISO](#)

ISO8601 formatierte Zeitstempel sehen wie in den folgenden Beispielen aus. Das erste Beispiel zeigt die Zeit in der [koordinierten Weltzeit \(UTC\)](#), indem ein Zeichen `Z` nach der Uhrzeit angegeben wird. Datum und Uhrzeit werden durch ein `T` getrennt.

```
2019-10-31T22:21:41Z
```

Um eine andere Zeitzone anzugeben, geben Sie statt der ein `+` oder `-` und die Anzahl der Stunden `Z`, vor oder hinter der die gewünschte Zeitzone liegt UTC, als zweistelligen Wert an. Das folgende Beispiel zeigt dieselbe Uhrzeit wie das vorherige Beispiel, jedoch angepasst an die Pacific Standard Time, die acht Stunden dahinter UTC liegt.

```
2019-10-31T14:21:41-08
```

- `wire` — Der Standardwert für AWS CLI Version 1. Falls angegeben, AWS CLI werden alle Zeitstempelwerte genau so angezeigt, wie sie in der HTTP Abfrageantwort empfangen wurden.

Zu diesem Eintrag gibt es keine entsprechende Umgebungsvariable oder Befehlszeilenoption.

```
cli_timestamp_format = iso8601
```

credential_process

Gibt einen externen Befehl an, der AWS CLI ausgeführt wird, um die für diesen Befehl zu verwendenden Anmeldeinformationen zu generieren oder abzurufen. Der Befehl muss die Anmeldeinformationen in einem bestimmten Format zurückgeben. Weitere Informationen zur Verwendung dieser Einstellung finden Sie unter [Beschaffung von Referenzen mit einem externen Prozess in der AWS CLI](#).

Zu diesem Eintrag gibt es keine entsprechende Umgebungsvariable oder Befehlszeilenoption.

```
credential_process = /opt/bin/awscreds-retriever --username susan
```

credential_source

Wird innerhalb von EC2 Amazon-Instances oder Containern verwendet, um anzugeben, wo sie Anmeldeinformationen finden AWS CLI können, die sie verwenden können, um die Rolle anzunehmen, die Sie mit dem `role_arn` Parameter angegeben haben. Sie können `source_profile` und `credential_source` nicht im selben Profil angeben.

Dieser Parameter kann einen von drei Werten haben:

- `Umgebung` — Gibt an, dass Quellanmeldedaten aus Umgebungsvariablen abgerufen werden sollen. AWS CLI
- `Ec2 InstanceMetadata` — Gibt an, dass die dem AWS CLI [EC2Instanzprofil](#) zugeordnete IAM Rolle zum Abrufen der Quellanmeldeinformationen verwendet werden soll.
- `EcsContainer`— Gibt an, dass die dem AWS CLI ECS Container zugeordnete IAM Rolle als Quellanmeldedaten verwendet werden soll.

```
credential_source = Ec2InstanceMetadata
```

duration_seconds

Gibt die maximale Dauer der Rollensitzung in Sekunden an. Der Wert kann zwischen 900 Sekunden (15 Minuten) und der maximalen Sitzungsdauer für die Rolle liegen maximal 43 200). Dieser Parameter ist optional. Standardmäßig ist der Wert auf 3 600 Sekunden festgelegt.

endpoint_url

Gibt den Endpunkt an, der für alle Serviceanforderungen verwendet wird. Wenn diese Einstellung im [services](#)-Abschnitt der config-Datei verwendet wird, wird der Endpunkt nur für den angegebenen Service genutzt.

Im folgenden Beispiel wird der globale Endpunkt `http://localhost:1234` und ein servicespezifischer Endpunkt namens `http://localhost:4567` für Amazon S3 verwendet.

```
[profile dev]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
    endpoint_url = http://localhost:4567
```

Die Einstellungen für die Endpunktkonfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktkonfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT](#)-, [AWS_USE_FIPS_ENDPOINT](#)- und [AWS_ENDPOINT_URL](#)-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem [services](#)-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem [profile](#) der freigegebenen config-Datei bereitgestellt wird

7. [use_dualstack_endpoint](#)-, [use_fips_endpoint](#)- und [endpoint_url](#)-Einstellungen.
8. Jeder Standard-Endpoint URL für den jeweiligen Endpoint AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

ignore_configure_endpoint_urls

Wenn diese Option aktiviert ist, AWS CLI ignoriert sie alle in der config Datei angegebenen benutzerdefinierten Endpunktkonfigurationen. Gültige Werte sind **true** und **false**.

```
ignore_configure_endpoint_urls = true
```

Die Einstellungen für die Endpunktkonfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktkonfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpoint-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT](#)-, [AWS_USE_FIPS_ENDPOINT](#)- und [AWS_ENDPOINT_URL](#)-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint](#)-, [use_fips_endpoint](#)- und [endpoint_url](#)-Einstellungen.
8. Jeder Standard-Endpoint URL für den jeweiligen Endpoint AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

external_id

Gibt eine eindeutige Kennung an, die von Dritten verwendet wird, um eine Rolle in den Konten ihrer Kunden zu übernehmen. Dies entspricht dem Parameter `ExternalId` in der Operation `AssumeRole`. Dieser Parameter ist nur erforderlich, wenn die Vertrauensrichtlinie für die Rolle einen Wert für `ExternalId` angibt. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [So verwenden Sie eine externe ID, wenn Sie Dritten Zugriff auf Ihre AWS Ressourcen gewähren](#).

max_attempts

Gibt einen Wert für die maximale Anzahl von Wiederholungsversuchen an, die der AWS CLI Wiederholungshandler verwendet, wobei der erste Aufruf auf den von Ihnen angegebenen `max_attempts` Wert angerechnet wird.

Sie können diesen Wert überschreiben, indem Sie die Umgebungsvariable `AWS_MAX_ATTEMPTS` verwenden.

```
max_attempts = 3
```

mfa_serial

Die Identifikationsnummer eines MFA Geräts, das verwendet werden soll, wenn eine Rolle übernommen wird. Dies ist nur dann verpflichtend, wenn die Vertrauensrichtlinie der Rolle, die übernommen wird, eine Bedingung beinhaltet, die eine MFA Authentifizierung erfordert. Der Wert kann entweder eine Seriennummer für ein Hardwaregerät (z. B. GAHT12345678) oder ein Amazon-Ressourcenname (ARN) für ein virtuelles MFA Gerät (z. B. `arn:aws:iam::123456789012:mfa/user`) sein.

output

Gibt das Standardausgabeformat für Befehle an, die mit diesem Profil angefordert wurden. Sie können alle folgenden Werte angeben:

- **json**— Die Ausgabe ist als [JSON](#) Zeichenfolge formatiert.
- **text** – Die Ausgabe wird als mehrere Zeilen mit tabulatorgetrennten Zeichenfolgenwerten formatiert. Dies kann nützlich sein, um die Ausgabe an einen Textprozessor wie `grep`, `sed` oder `awk` zu übergeben.
- **table** – Die Ausgabe erfolgt in Form einer Tabelle mit den Zeichen `+|-`, um die Zellenrahmen zu bilden. Normalerweise wird die Information in einem benutzerfreundlichen Format

wiedergegeben, das viel einfacher zu lesen ist als die anderen, jedoch programmatisch nicht so nützlich ist.

Kann von der Umgebungsvariablen `AWS_DEFAULT_OUTPUT` oder mit der Befehlszeilenoption `--output` überschrieben werden.

```
output = table
```

parameter_validation

Gibt an, ob der AWS CLI Client versucht, Parameter zu überprüfen, bevor er sie an den AWS Dienstendpunkt sendet.

- `true` (wahr) – Dies ist der Standardwert. Falls angegeben, AWS CLI führt der eine lokale Überprüfung der Befehlszeilenparameter durch.
- `false` — Falls angegeben, überprüft der die Befehlszeilenparameter AWS CLI nicht, bevor sie an den AWS Dienstendpunkt gesendet werden.

Zu diesem Eintrag gibt es keine entsprechende Umgebungsvariable oder Befehlszeilenoption.

```
parameter_validation = false
```

region

Gibt an AWS-Region , an welche Anfragen für Befehle gesendet werden sollen, die mit diesem Profil angefordert wurden.

- Sie können einen der für den ausgewählten Service verfügbaren Regionscodes angeben, die unter [AWS -Regionen und -Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz aufgeführt sind.
- `aws_global` ermöglicht es Ihnen, den globalen Endpunkt für Services anzugeben, die zusätzlich zu regionalen Endpunkten wie AWS Security Token Service (AWS STS) und Amazon Simple Storage Service (Amazon S3) einen globalen Endpunkt unterstützen.

Sie können diesen Wert überschreiben, indem Sie die die `AWS_DEFAULT_REGION`-Umgebungsvariable oder die Befehlszeilenoption `--region` verwenden.

```
region = us-west-2
```

retry_mode

Gibt an, welcher Wiederholungsmodus verwendet wird AWS CLI . Es stehen drei Wiederholungsmodi zur Verfügung: Legacy (Standard), Standard und Adaptiv. Weitere Informationen zu Wiederholungsversuchen finden Sie unter [AWS CLI Wiederholungen in der AWS CLI](#).

Sie können diesen Wert überschreiben, indem Sie die Umgebungsvariable `AWS_RETRY_MODE` verwenden.

```
retry_mode = standard
```

role_arn

Gibt den Amazon-Ressourcennamen (ARN) einer IAM Rolle an, die Sie zur Ausführung der AWS CLI Befehle verwenden möchten. Sie müssen auch einen der folgenden Parameter angeben, um die Anmeldeinformationen zu identifizieren, die berechtigt sind, diese Rolle zu übernehmen:

- `source_profile`
- `credential_source`

```
role_arn = arn:aws:iam::123456789012:role/role-name
```

Diese Umgebungsvariable überschreibt die [AWS_ROLE_ARN](#)-Einstellung.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called "Übernehmen einer Rolle mit Web-Identität"](#).

role_session_name

Gibt den Namen an, der der Rollensitzung zugeordnet werden soll. Dieser Wert wird dem `RoleSessionName` Parameter beim AWS CLI Aufrufen der `AssumeRole` Operation zur Verfügung gestellt und wird Teil der angenommenen Rolle `userARN`: `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Dieser Parameter ist optional. Wenn Sie diesen Wert nicht angeben, wird automatisch ein Sitzungsname generiert. Dieser Name wird in den AWS CloudTrail -Protokollen für Einträge angezeigt, die dieser Sitzung zugeordnet sind.

```
role_session_name = maria_garcia_role
```

Die Umgebungsvariable [AWS_ROLE_SESSION_NAME](#) überschreibt diese Einstellung.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called “Übernehmen einer Rolle mit Web-Identität”](#).

services

Gibt die Servicekonfiguration an, die für Ihr Profil verwendet werden soll.

```
[profile dev-s3-specific-and-global]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
  endpoint_url = http://localhost:4567
```

Weitere Informationen zum Abschnitt `services` finden Sie unter [the section called “services”](#).

Die Umgebungsvariable [AWS_ROLE_SESSION_NAME](#) überschreibt diese Einstellung.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called “Übernehmen einer Rolle mit Web-Identität”](#).

sdk_ua_app_id

Eine einzelne AWS-Konto kann von mehreren Kundenanwendungen verwendet werden, um Anrufe zu tätigen AWS-Services. Die Anwendungs-ID identifiziert, welche Quellanwendung eine Reihe von Aufrufen mithilfe von getätigt hat AWS-Service. AWS SDKs und Dienste verwenden oder interpretieren diesen Wert nur, um ihn in der Kundenkommunikation wieder zum Vorschein zu bringen. Dieser Wert kann beispielsweise in betriebliche E-Mails aufgenommen werden, um eindeutig zu identifizieren, welche Ihrer Anwendungen mit der Benachrichtigung verknüpft ist.

Die Anwendungs-ID ist eine Zeichenfolge mit einer maximalen Länge von 50 Zeichen.

Buchstaben, Zahlen und die folgenden Sonderzeichen sind zulässig: ! \$ % & * + - . , ^ _ ` | ~ Standardmäßig wird kein Wert zugewiesen.

```
sdk_ua_app_id = prod1
```

Diese Einstellung kann mit der Umgebungsvariablen [AWS_SDK_UA_APP_ID](#) überschrieben werden. Sie können diesen Wert nicht als Befehlszeilenparameter festlegen.

source_profile

Gibt ein benanntes Profil mit langfristigen Anmeldeinformationen an, die die AWS CLI verwenden kann, um eine Rolle zu übernehmen, die Sie mit dem Parameter `role_arn` angegeben haben. Sie können `source_profile` und `credential_source` nicht im selben Profil angeben.

```
source_profile = production-profile
```

sts_regional_endpoints

Gibt an, wie der den AWS Dienstendpunkt AWS CLI bestimmt, den der AWS CLI Client für die Kommunikation mit dem AWS Security Token Service (AWS STS) verwendet. Der Standardwert für AWS CLI Version 1 ist `legacy`.

Sie können einen von zwei Werten angeben:

- **legacy**— Verwendet den globalen STS Endpunkt `sts.amazonaws.com`, für die folgenden AWS Regionen: `ap-northeast-1`, `ap-south-1`, `ap-southeast-1`, `ap-southeast-2`, `aws-global`, `ca-central-1`, `eu-central-1`, `eu-north-1`, `eu-west-1`, `eu-west-2`, `eu-west-3`, `sa-east-1`, `us-east-1`, `us-east-2`, `us-west-1`, und `us-west-2`. Für alle anderen Regionen wird automatisch deren jeweiliger regionaler Endpunkt verwendet.
- **regional**— Der verwendet AWS CLI immer den AWS STS Endpunkt für die aktuell konfigurierte Region. Wenn der Client beispielsweise für die Verwendung konfiguriert ist `us-west-2`, AWS STS werden alle Aufrufe an den regionalen Endpunkt `sts.us-west-2.amazonaws.com` statt an den globalen `sts.amazonaws.com` Endpunkt getätigt. Um eine Anforderung an den globalen Endpunkt zu senden, während diese Einstellung aktiviert ist, können Sie die Region auf `aws-global` festlegen.

Diese Einstellung kann mit der Umgebungsvariablen `AWS_STS_REGIONAL_ENDPOINTS` überschrieben werden. Sie können diesen Wert nicht als Befehlszeilenparameter festlegen.

use_dualstack_endpoint

Ermöglicht die Verwendung von Dual-Stack-Endpunkten zum Senden AWS von Anfragen. Weitere Informationen zu Dual-Stack-Endpunkten, die IPv4 sowohl IPv6 Datenverkehr als auch unterstützen, finden Sie unter [Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten](#) im Amazon

Simple Storage Service-Benutzerhandbuch. Dual-Stack-Endpunkte sind für einige Services in einigen Regionen verfügbar. Wenn kein Dual-Stack-Endpunkt für den Service oder AWS-Region existiert, schlägt die Anfrage fehl. Diese ist standardmäßig deaktiviert.

Diese Einstellung und die Einstellung `use_accelerate_endpoint` schließen sich gegenseitig aus.

Die Einstellungen für die Endpunktconfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktconfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption `--endpoint-url`
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` oder die Profileinstellung `ignore_configure_endpoint_urls` zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen `AWS_ENDPOINT_URL_<SERVICE>` bereitgestellt wird, z. B. `AWS_ENDPOINT_URL_DYNAMODB`
4. Die von den `AWS_USE_DUALSTACK_ENDPOINT`-, `AWS_USE_FIPS_ENDPOINT`- und `AWS_ENDPOINT_URL`-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung `endpoint_url` in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung `endpoint_url` in einem profile der freigegebenen config-Datei bereitgestellt wird
7. `use_dualstack_endpoint`-, `use_fips_endpoint`- und `endpoint_url`-Einstellungen.
8. Jeder Standard-Endpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

use_fips_endpoint

Einige AWS Dienste bieten Endgeräte, die in einigen Fällen den [Federal Information Processing Standard \(FIPS\) 140-2](#) unterstützen. AWS-Regionen Wenn der AWS Dienst dies unterstützt FIPS, gibt diese Einstellung an, welcher FIPS Endpunkt verwendet werden soll. AWS CLI Im Gegensatz zu AWS Standardendpunkten verwenden FIPS Endpunkte eine TLS Softwarebibliothek, die 140-2 entspricht FIPS. Diese Endpunkte können von Unternehmen erfordert werden, die mit der US-Regierung interagieren.

Wenn diese Einstellung aktiviert ist, aber kein FIPS Endpunkt für den Dienst in Ihrer Region vorhanden ist, schlägt der Befehl möglicherweise fehl. Geben Sie in diesem Fall mithilfe der `--endpoint-url`-Option manuell den Endpunkt an, der im Befehl verwendet werden soll, oder verwenden Sie [servicespezifische Endpunkte](#).

Weitere Informationen zur Angabe von FIPS Endpunkten finden Sie in der AWS-Region unter [FIPSEndpunkte nach Dienst](#).

Die Einstellungen für die Endpunkt-Konfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunkt-Konfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption `--endpoint-url`
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` oder die Profileinstellung `ignore_configure_endpoint_urls` zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen `AWS_ENDPOINT_URL_<SERVICE>` bereitgestellt wird, z. B. `AWS_ENDPOINT_URL_DYNAMODB`
4. Die von den `AWS_USE_DUALSTACK_ENDPOINT`-, `AWS_USE_FIPS_ENDPOINT`- und `AWS_ENDPOINT_URL`-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunkt-Wert, der durch die Einstellung `endpoint_url` in einem `services`-Abschnitt der freigegebenen `config`-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung `endpoint_url` in einem `profile` der freigegebenen `config`-Datei bereitgestellt wird
7. `use_dualstack_endpoint`-, `use_fips_endpoint`- und `endpoint_url`-Einstellungen.
8. Jeder Standard-Endpoint-URL für den jeweiligen Endpunkt-AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeinen Amazon Web Services-Referenz.

[web_identity_token_file](#)

Gibt den Pfad zu einer Datei an, die ein OAuth 2.0-Zugriffstoken oder ein OpenID Connect-ID-Token enthält, das von einem Identitätsanbieter bereitgestellt wird. Die AWS CLI lädt den Inhalt dieser Datei und übergibt ihn als `WebIdentityToken`-Argument an die Operation `AssumeRoleWithWebIdentity`.

Diese Umgebungsvariable überschreibt die [AWS_WEB_IDENTITY_TOKEN_FILE](#)-Einstellung.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called “Übernehmen einer Rolle mit Web-Identität”](#).

tcp_keepalive

Gibt an, ob der AWS CLI Client TCP Keep-Alive-Pakete verwendet.

Zu diesem Eintrag gibt es keine entsprechende Umgebungsvariable oder Befehlszeilenoption.

```
tcp_keepalive = false
```

Einstellungen für benutzerdefinierte S3-Befehle

Amazon S3 unterstützt mehrere Einstellungen, mit denen konfiguriert wird, wie Amazon AWS CLI S3-Operationen ausgeführt werden. Einige gelten für alle S3-Befehle in den `s3`-Namespaces und den `s3api`-Namespaces. Andere sind speziell für die „benutzerdefinierten“ S3-Befehle vorgesehen, die allgemeine Operationen abstrahieren und mehr als nur eine one-to-one Zuordnung zu einer API Operation ermöglichen. Für die `aws s3`-Übertragungsbefehle `cp`, `sync`, `mv` und `rm` gibt es zusätzliche Einstellungen, die Sie zur Steuerung von S3-Übertragungen verwenden können.

Alle diese Optionen können durch Angabe der verschachtelten Einstellung `s3` in Ihrer `config`-Datei konfiguriert werden. Jede Einstellung wird dann in einer eigenen Zeile eingerückt.

Note

Diese Einstellungen sind völlig optional. Die `aws s3`-Übertragungsbefehle müssten auch ohne Konfiguration dieser Einstellungen erfolgreich ausgeführt werden können. Die Einstellungen werden bereitgestellt, damit Sie die Leistung verbessern oder die besondere Umgebung berücksichtigen können, in der Sie diese `aws s3`-Befehle ausführen.

Diese Einstellungen werden alle unter einem `s3`-Schlüssel der obersten Ebene in der `config`-Datei festgelegt, wie im folgenden Beispiel für das `development`-Profil dargestellt.

```
[profile development]
s3 =
    max_concurrent_requests = 20
```

```
max_queue_size = 10000
multipart_threshold = 64MB
multipart_chunksize = 16MB
max_bandwidth = 50MB/s
use_accelerate_endpoint = true
addressing_style = path
```

Die folgenden Einstellungen gelten für alle S3-Befehle in den `s3-` oder `s3api-`Namespaces.

addressing_style

Gibt an, welcher Adressierungsstil verwendet werden soll. Dadurch wird gesteuert, ob der Bucket-Name im Hostnamen oder Teil des URL ist. Gültige Werte sind: `path`, `virtual` und `auto`. Der Standardwert ist `auto`.

Es gibt zwei Möglichkeiten, einen Amazon-S3-Endpunkt zu erstellen. Die erste wird als `virtual` bezeichnet und beinhaltet den Bucket-Namen als Teil des Hostnamens. Beispiel: `https://bucketname.s3.amazonaws.com`. Alternativ behandeln Sie mit dem `path` Stil den Bucket-Namen so, als ob es sich um einen Pfad in der handeltURI; zum Beispiel `https://s3.amazonaws.com/bucketname`. Der Standardwert in der CLI ist `to use`. Dabei wird versucht `auto`, den `virtual` Stil zu verwenden, wo es möglich ist, aber bei Bedarf auf den `path` Stil zurückgreift. Wenn Ihr Bucket-Name beispielsweise nicht DNS kompatibel ist, kann der Bucket-Name nicht Teil des Hostnamens sein und muss im Pfad enthalten sein. Mit `auto` CLI erkennt der diesen Zustand und wechselt automatisch zum `path` Stil für Sie. Wenn Sie den Adressierungsstil auf `einstellenpath`, müssen Sie sicherstellen, dass die AWS Region, die Sie in der konfiguriert haben, AWS CLI mit der Region Ihres Buckets übereinstimmt.

payload_signing_enabled

Gibt an, ob Sigv4-Payloads SHA256 signiert werden sollen. Standardmäßig ist dies für Streaming-Uploads (`UploadPart` und `PutObject`) bei Verwendung deaktiviert. HTTPS Standardmäßig ist dies `false` für Streaming-Uploads (`UploadPart` und `PutObject`) aktiviert, aber nur, wenn a vorhanden `ContentMD5` ist (es wird standardmäßig generiert) und der Endpunkt verwendet. HTTPS

Wenn dieser Wert auf `true` gesetzt ist, erhalten S3-Anfragen eine zusätzliche Inhaltsvalidierung in Form einer SHA256 Prüfsumme, die für Sie berechnet und in der Anforderungssignatur enthalten ist. Wenn „`false`“ festgelegt ist, wird die Prüfsumme nicht berechnet. Eine Deaktivierung dieser Einstellung kann nützlich sein, um den durch die Prüfsummenberechnung entstandenen Leistungsaufwand zu reduzieren.

use_accelerate_endpoint

Verwendet den Amazon-S3-Accelerate-Endpunkt für alle `s3-` und `s3api-`Befehle. Der Standardwert ist "false". Diese Einstellung und die Einstellung `use_dualstack_endpoint` schließen sich gegenseitig aus.

Wenn auf `true` gesetzt, AWS CLI leitet alle Amazon S3 S3-Anfragen an den S3 Accelerate Endpunkt unter `weitere3-accelerate.amazonaws.com`. Um diesen Endpunkt zu verwenden, müssen Sie Ihrem Bucket die Verwendung von S3 Accelerate ermöglichen. Alle Anforderungen werden unter Verwendung der virtuellen Bucket-Adressierungsform *my-bucket*.s3-accelerate.amazonaws.com gesendet. `ListBuckets-`, `CreateBucket-` und `DeleteBucket` -Anfragen werden nicht an den S3-Accelerate-Endpunkt gesendet, da dieser Endpunkt diese Operationen nicht unterstützt. Dieses Verhalten kann auch festgelegt werden, wenn der Parameter `--endpoint-url` für einen `s3-` oder `s3api-`Befehl auf `https://s3-accelerate.amazonaws.com` oder `http://s3-accelerate.amazonaws.com` gesetzt ist.

Die folgenden Einstellungen gelten nur für Befehle im Befehlssatz für den `s3-`Namespace.

max_bandwidth

Gibt die maximale Bandbreite an, die zum Hoch- und Herunterladen von Daten in und aus Amazon S3 verbraucht werden kann. Standardmäßig ist kein Grenzwert festgelegt.

Diese Einstellung begrenzt die maximale Bandbreite, die S3-Befehle zur Übertragung von Daten an und aus Amazon S3 nutzen können. Der Wert gilt nur für Uploads und Downloads, nicht für Kopien oder Löschvorgänge. Der Wert wird in Bytes pro Sekunde ausgedrückt. Der Wert kann wie folgt angegeben werden:

- Als ganze Zahl. Bei Angabe von `1048576` wird für die maximale Bandbreitennutzung beispielsweise 1 Megabyte pro Sekunde festgelegt.
- Als ganze Zahl, gefolgt von einem Suffix für die Rate. Sie können Suffixe unter Verwendung von `KB/s`, `MB/s` oder `GB/s` angeben. Zum Beispiel `300KB/s`, `10MB/s`.

Im Allgemeinen empfehlen wir, zunächst durch Verringerung des Werts für `max_concurrent_requests` eine niedrigere Bandbreitennutzung anzugeben. Wenn der Bandbreitenverbrauch dadurch nicht angemessen auf die gewünschte Rate begrenzt werden kann, können Sie die Einstellung `max_bandwidth` verwenden, um den Bandbreitenverbrauch weiter zu begrenzen. Dies liegt daran, dass `max_concurrent_requests` steuert, wie viele Threads zurzeit ausgeführt werden. Wenn Sie stattdessen zuerst `max_bandwidth` senken, es

aber bei einer hohen `max_concurrent_requests`-Einstellung belassen, kann dies dazu führen, dass Threads unnötig warten müssen. Dies kann zu einem übermäßigen Ressourcenverbrauch und Verbindungszeitüberschreitungen führen.

max_concurrent_requests

Gibt die maximale Anzahl gleichzeitiger Anforderungen an. Der Standardwert lautet 10.

Die `aws s3`-Übertragungsbefehle sind Multithread-Befehle. Zu jedem Zeitpunkt können mehrere Amazon-S3-Anforderungen ausgeführt werden. Wenn Sie beispielsweise den Befehl verwenden, `aws s3 cp localdir s3://bucket/ --recursive` um Dateien in einen S3-Bucket hochzuladen, AWS CLI können sie die Dateien `localdir/file1`, `localdir/file2`, und `localdir/file3` parallel hochladen. Die Einstellung `max_concurrent_requests` gibt die maximale Anzahl von Übertragungsoperationen an, die gleichzeitig ausgeführt werden können.

Unter Umständen müssen Sie diesen Wert aus einem der folgenden Gründe ändern:

- Verringerung dieses Werts – in einigen Umgebungen kann der Standardwert von 10 gleichzeitigen Anforderungen zu einer Überlastung des Systems führen. Die Folge können Zeitüberschreitungen bei der Verbindung oder eine herabgesetzte Reaktionsfähigkeit des Systems sein. Wenn Sie diesen Wert senken, sind die S3-Übertragungsbefehle weniger ressourcenintensiv. Der Nachteil ist jedoch, dass S3-Übertragungen länger dauern können. Eine Senkung dieses Wertes könnte erforderlich sein, wenn Sie ein Tool zur Begrenzung der Bandbreite verwenden.
- Erhöhung dieses Werts – in einigen Fällen kann es sinnvoll sein, die Amazon-S3-Übertragungen so schnell wie möglich durchzuführen und dabei so viel Netzwerkbandbreite wie nötig zu beanspruchen. In einem solchen Fall reicht die standardmäßige Anzahl gleichzeitiger Anforderungen möglicherweise nicht aus, um die gesamte verfügbare Netzwerkbandbreite zu nutzen. Eine Erhöhung dieses Werts kann dazu führen, dass sich die Zeit für die Durchführung einer Amazon-S3-Übertragung verkürzt.

max_queue_size

Gibt die maximale Anzahl von Aufgaben in der Aufgabenwarteschlange an. Der Standardwert lautet 1000.

Das verwendet AWS CLI intern ein Modell, bei dem Amazon S3 S3-Aufgaben in eine Warteschlange gestellt werden, die dann von Verbrauchern ausgeführt werden, deren Anzahl begrenzt ist durch `max_concurrent_requests`. Eine Aufgabe entspricht im Allgemeinen einer einzelnen Amazon-S3-Operation. Eine mögliche Aufgabe könnte beispielsweise `PutObjectTask`, `GetObjectTask` oder `UploadPartTask` sein. Die Aufgaben

können der Warteschlange viel schneller hinzugefügt werden, als die Konsumenten die Aufgaben abschließen. Um ein unbegrenztes Wachstum zu vermeiden, ist die Größe der Aufgabenwarteschlange begrenzt. Diese Einstellung ändert den Wert dieser maximalen Anzahl.

Im Allgemeinen müssen Sie diese Einstellung nicht ändern. Diese Einstellung entspricht auch der Anzahl der Aufgaben, von denen sie weiß, dass sie ausgeführt werden müssen. AWS CLI Das bedeutet, dass sie standardmäßig nur 1000 Aufgaben vor sich sehen AWS CLI können. Wenn dieser Wert erhöht wird, AWS CLI können sie schneller die Gesamtzahl der benötigten Aufgaben ermitteln, vorausgesetzt, dass die Warteschlangenrate schneller ist als die Geschwindigkeit, mit der die Aufgaben abgeschlossen werden. Der Nachteil besteht darin, dass für eine größere maximale Warteschlangengröße mehr Speicher benötigt wird.

multipart_chunksize

Gibt die Blockgröße an, die für mehrteilige Übertragungen einzelner Dateien AWS CLI verwendet wird. Der Standardwert ist 8 MB und mindestens 5 MB.

Wenn eine Datei den `multipart_threshold`-Wert überschreitet, teilt die AWS CLI die Datei in Blöcke dieser Größe. Dieser Wert kann mit derselben Syntax wie `multipart_threshold` angegeben werden, also entweder als ganzzahlige Bytezahl oder unter Verwendung einer Größe und eines Suffixes.

multipart_threshold

Gibt den Größenschwellenwert an, der für mehrteilige Übertragungen einzelner Dateien AWS CLI verwendet wird. Der Standardwert ist 8 MB.

Beim Hochladen, Herunterladen oder Kopieren einer Datei gehen die Amazon-S3-Befehle zu mehrteiligen Operationen über, wenn die Datei diese Größe überschreitet. Sie können diesen Wert auf zwei Arten angeben:

- Dateigröße in Bytes. Beispiel, 1048576.
- Dateigröße mit einem Größensuffix. Sie können KB, MB, GB oder TB verwenden. Zum Beispiel: 10MB, 1GB

Note

S3 kann Beschränkungen für gültige Werte anwenden, die für mehrteilige Operationen verwendet werden können. Weitere Informationen finden Sie in der [S3-Dokumentation zu mehrteiligen Uploads](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Konfiguration von Umgebungsvariablen für AWS CLI

Umgebungsvariablen bieten eine weitere Möglichkeit, Konfigurationsoptionen und Anmeldeinformationen anzugeben, und können für die Skripterstellung nützlich sein.

Vorrang von Optionen

- Wenn Sie eine Option mit einer der in diesem Thema beschriebenen Umgebungsvariablen angeben, setzt sie jeden Wert außer Kraft, der aus einem Profil in der Konfigurationsdatei geladen wurde.
- Wenn Sie eine Option mithilfe eines Parameters in der AWS CLI Befehlszeile angeben, überschreibt sie jeden Wert aus der entsprechenden Umgebungsvariablen oder einem Profil in der Konfigurationsdatei.

Weitere Hinweise zur Rangfolge und dazu, wie AWS CLI festgelegt wird, welche Anmeldeinformationen verwendet werden sollen, finden Sie unter [Konfiguration der Einstellungen für den AWS CLI](#)

Themen

- [Festlegen von Umgebungsvariablen](#)
- [AWS CLI unterstützte Umgebungsvariablen](#)

Festlegen von Umgebungsvariablen

Die folgenden Beispiele zeigen, wie Sie Umgebungsvariablen für den Standardbenutzer konfigurieren können.

Linux or macOS

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
$ export AWS_DEFAULT_REGION=us-west-2
```

Durch die Festlegung der Umgebungsvariablen wird der verwendete Wert bis zum Ende der Shell-Sitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert. Sie können Variablen für zukünftige Sitzungen persistent machen, indem Sie sie im Startup-Skript Ihrer Shell festlegen.

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> setx AWS_DEFAULT_REGION us-west-2
```

Bei Verwendung von [setx](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert in der aktuellen Eingabeaufforderungssitzung und allen nach Ausführung des Befehls erstellten Eingabeaufforderungssitzungen geändert. Andere Befehls-Shells, die zum Zeitpunkt der Befehlsausführung bereits ausgeführt werden, sind hiervon nicht betroffen. Möglicherweise müssen Sie Ihr Terminal neu starten, damit die Einstellungen geladen werden.

Einrichten nur für die aktuelle Sitzung

Bei Verwendung von [set](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert bis zum Ende der aktuellen Eingabeaufforderungssitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert.

```
C:\> set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
C:\> set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> set AWS_DEFAULT_REGION=us-west-2
```

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
PS C:\> $Env:AWS_DEFAULT_REGION="us-west-2"
```

Wenn Sie an der PowerShell Eingabeaufforderung eine Umgebungsvariable festlegen, wie in den vorherigen Beispielen gezeigt, wird der Wert nur für die Dauer der aktuellen Sitzung gespeichert. Um die Einstellung der Umgebungsvariablen für alle Sitzungen PowerShell und Befehlszeilensitzungen beizubehalten, speichern Sie sie mithilfe der Systemanwendung in der Systemsteuerung. Alternativ können Sie die Variable für alle future PowerShell Sitzungen festlegen, indem Sie sie zu Ihrem PowerShell Profil hinzufügen. Weitere Informationen zum Speichern von Umgebungsvariablen oder deren Beibehaltung über mehrere Sitzungen hinweg finden Sie in der [PowerShell Dokumentation](#).

AWS CLI unterstützte Umgebungsvariablen

Das AWS CLI unterstützt die folgenden Umgebungsvariablen.

AWS_ACCESS_KEY_ID

Gibt einen AWS Zugriffsschlüssel an, der einem IAM Konto zugeordnet ist.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `aws_access_key_id`. Es ist nicht möglich, die Zugriffsschlüssel-ID mithilfe einer Befehlszeilenoption anzugeben.

AWS_CA_BUNDLE

Gibt den Pfad zu einem Zertifikatspaket an, das für die HTTPS Zertifikatsvalidierung verwendet werden soll.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung [ca_bundle](#). Sie können diese Umgebungsvariable mithilfe des [--ca-bundle](#)-Befehlszeilenparameters überschreiben.

AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS

Wenn die Quell- und Ziel-Buckets identisch sind, wenn Sie den `s3 mv` Befehl Benutzerdefiniert verwenden, kann die Quelldatei oder das Quellobjekt auf sich selbst verschoben werden, was zu einem versehentlichen Löschen Ihrer Quelldatei oder Ihres Quellobjekts führen kann. Die `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` Umgebungsvariable und `--validate-same-s3-paths` -option geben an, ob Ihr Access Point ARNs oder Ihre Access Point-Aliase in Ihrer Amazon S3-Quelle oder Ihrem Amazon S3-Ziel URIs validiert werden sollen.

Note

Die Pfadvalidierung für `s3 mv` erfordert zusätzliche API Aufrufe.

AWS_CONFIG_FILE

Gibt den Speicherort der Datei an, die zum Speichern von Konfigurationsprofilen AWS CLI verwendet wird. Der Standardpfad ist `~/.aws/config`.

Sie können diesen Wert nicht in einer benannten Profileinstellung oder mithilfe eines Befehlszeilenparameters angeben.

AWS_DATA_PATH

Eine Liste zusätzlicher Verzeichnisse, die `~/ .aws/models` beim Laden von AWS CLI Daten außerhalb des integrierten Suchpfads überprüft werden sollen. Durch die Festlegung dieser Umgebungsvariable werden zusätzliche Verzeichnisse angegeben, die zuerst überprüft werden, bevor auf die integrierten Suchpfade zurückgegriffen wird. Mehrere Einträge sollten mit dem `os.pathsep`-Zeichen getrennt werden (unter Linux oder macOS : und unter Windows ;).

AWS_DEFAULT_OUTPUT

Gibt das zu verwendende [Ausgabeformat](#) an.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `output`. Sie können diese Umgebungsvariable mithilfe des `--output`-Befehlszeilenparameters überschreiben.

AWS_DEFAULT_REGION

Der `Default region name` identifiziert die AWS Region, an deren Server Sie Ihre Anfragen standardmäßig senden möchten. Dies ist in der Regel die nächstgelegene Region, aber jede Region ist zulässig. Sie können beispielsweise `us-west-2` eingeben, um USA West (Oregon) zu verwenden. Dies ist die Region, an die alle späteren Anfragen gesendet werden, es sei denn, Sie geben in einem Befehl etwas anderes an.

Note

Sie müssen eine AWS Region angeben, wenn Sie die verwenden AWS CLI, entweder explizit oder indem Sie eine Standardregion festlegen. Eine Liste der verfügbaren Regionen finden Sie unter [Regionen und Endpunkte](#). Die von der verwendeten Regionsbezeichner AWS CLI sind dieselben Namen, die Sie in den Dienstendpunkten AWS Management Console URLs sehen.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `region`. Sie können diese Umgebungsvariable überschreiben, indem Sie den `--region` Befehlszeilenparameter Umgebungsvariable verwenden.

AWS_EC2_METADATA_DISABLED

Deaktiviert die Verwendung des EC2 Amazon-Instance-Metadatendienstes (IMDS).

Wenn dieser Wert auf „true“ gesetzt ist, werden keine Benutzeranmeldedaten oder Konfigurationen (wie die Region) angefordertIMDS.

AWS_ENDPOINT_URL

Gibt den Endpunkt an, der für alle Serviceanforderungen verwendet wird.

Die Einstellungen für die Endpunktkonfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktkonfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT](#)-, [AWS_USE_FIPS_ENDPOINT](#)- und [AWS_ENDPOINT_URL](#)-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint](#)-, [use_fips_endpoint](#)- und [endpoint_url](#)-Einstellungen.
8. Jeder Standardendpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

AWS_ENDPOINT_URL_<SERVICE>

Gibt einen benutzerdefinierten Endpunkt an, der für einen bestimmten Dienst verwendet <SERVICE> wird und der durch den AWS-Service Bezeichner ersetzt wird. Amazon DynamoDB Hat zum Beispiel ein serviceId von [DynamoDB](#). Für diesen Dienst lautet die URL Umgebungsvariable für den Endpunkt [AWS_ENDPOINT_URL_DYNAMODB](#).

Eine Liste aller servicespezifischen Umgebungsvariablen finden Sie unter [Liste der servicespezifischen Kennungen](#).

Die Einstellungen für die Endpunktconfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktconfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT-](#), [AWS_USE_FIPS_ENDPOINT-](#) und [AWS_ENDPOINT_URL-](#)Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint-](#), [use_fips_endpoint-](#) und [endpoint_url](#)-Einstellungen.
8. Jeder Standardendpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

AWS_IGNORE_CONFIGURED_ENDPOINT_URLS

Wenn diese Option aktiviert ist, AWS CLI ignoriert sie alle benutzerdefinierten Endpunktconfigurationen. Gültige Werte sind **true** und **false**.

Die Einstellungen für die Endpunktconfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktconfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten

3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. `AWS_ENDPOINT_URL_DYNAMODB`
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT](#)-, [AWS_USE_FIPS_ENDPOINT](#)- und [AWS_ENDPOINT_URL](#)-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint](#)-, [use_fips_endpoint](#)- und [endpoint_url](#)-Einstellungen.
8. Jeder Standard-Endpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

AWS_MAX_ATTEMPTS

Gibt einen Wert für die maximale Anzahl von Wiederholungsversuchen an, die der AWS CLI Wiederholungshandler verwendet, wobei der erste Aufruf auf den von Ihnen angegebenen Wert angerechnet wird. Weitere Informationen zu Wiederholungsversuchen finden Sie unter [AWS CLI Wiederholungen in der AWS CLI](#).

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `max_attempts`.

AWS_METADATA_SERVICE_NUM_ATTEMPTS

Beim Versuch, Anmeldeinformationen für eine EC2 Amazon-Instance abzurufen, die mit einer IAM Rolle konfiguriert wurde, wird AWS CLI versucht, Anmeldeinformationen einmal vom Instance-Metadaten-Service abzurufen, bevor der Vorgang beendet wird. Wenn Sie wissen, dass Ihre Befehle auf einer EC2 Amazon-Instance ausgeführt werden, können Sie diesen Wert erhöhen, um AWS CLI es mehrmals zu wiederholen, bevor Sie aufgeben.

AWS_METADATA_SERVICE_TIMEOUT

Die Anzahl der Sekunden, bevor bei einer Verbindung zum Instance-Metadaten-Service eine Zeitüberschreitung eintritt. Beim Versuch, Anmeldeinformationen auf einer EC2 Amazon-Instance abzurufen, die mit einer IAM Rolle konfiguriert ist, wird eine Verbindung zum Instance-Metadaten-Service standardmäßig nach 1 Sekunde unterbrochen. Wenn Sie wissen, dass Sie auf einer EC2 Amazon-Instance mit einer konfigurierten IAM Rolle arbeiten, können Sie diesen Wert bei Bedarf erhöhen.

AWS_PROFILE

Gibt den Namen des AWS CLI Profils mit den zu verwendenden Anmeldeinformationen und Optionen an. Dabei kann es sich um den Namen eines Profils handeln, das in einer `credentials`- oder `config`-Datei gespeichert ist, oder um den Wert `default` zur Nutzung des Standardprofils.

Ist sie definiert, überschreibt diese Umgebungsvariable das Verhalten, bei dem das Profil mit dem Namen `[default]` in der Konfigurationsdatei verwendet wird. Sie können diese Umgebungsvariable mithilfe des `--profile`-Befehlszeilenparameters überschreiben.

AWS_RETRY_MODE

Gibt an, welcher Wiederholungsmodus AWS CLI verwendet wird. Es stehen drei Wiederholungsmodi zur Verfügung: Legacy (Standard), Standard und Adaptiv. Weitere Informationen zu Wiederholungsversuchen finden Sie unter [AWS CLI Wiederholungen in der AWS CLI](#).

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `retry_mode`.

AWS_ROLE_ARN

Gibt den Amazon-Ressourcennamen (ARN) einer IAM Rolle mit einem Web-Identitätsanbieter an, die Sie zur Ausführung der AWS CLI Befehle verwenden möchten.

Wird mit `AWS_WEB_IDENTITY_TOKEN_FILE`- und `AWS_ROLE_SESSION_NAME`-Umgebungsvariablen genutzt.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung [role_arn](#). Sie können keinen Rollensitzungsnamen als Befehlszeilenparameter angeben.

Note

Diese Umgebungsvariable gilt nur für eine Übernahmerolle mit dem Web-Identitätsanbieter, sie gilt nicht für die allgemeine Konfiguration des Rollenanbieters.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called "Übernehmen einer Rolle mit Web-Identität"](#).

AWS_ROLE_SESSION_NAME

Gibt den Namen an, der der Rollensitzung zugeordnet werden soll. Dieser Wert wird dem `RoleSessionName` Parameter übergeben, wenn der die `AssumeRole` Operation AWS CLI aufruft, und wird Teil der angenommenen Rolle `userARN`: `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Dieser Parameter ist optional. Wenn Sie diesen Wert nicht angeben, wird automatisch ein Sitzungsname generiert. Dieser Name erscheint in den AWS CloudTrail Protokollen für Einträge, die mit dieser Sitzung verknüpft sind.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung [role_session_name](#).

Wird mit `AWS_ROLE_ARN`- und `AWS_WEB_IDENTITY_TOKEN_FILE`-Umgebungsvariablen genutzt.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called “Übernehmen einer Rolle mit Web-Identität”](#).

Note

Diese Umgebungsvariable gilt nur für eine Übernahmerolle mit dem Web-Identitätsanbieter, sie gilt nicht für die allgemeine Konfiguration des Rollenanbieters.

AWS_SDK_UA_APP_ID

Eine einzelne AWS-Konto kann von mehreren Kundenanwendungen verwendet werden, um Anrufe zu tätigen AWS-Services. Die Anwendungs-ID identifiziert, welche Quellanwendung eine Reihe von Aufrufen mithilfe von getätigt hat AWS-Service. AWS SDKs und Dienste verwenden oder interpretieren diesen Wert nur, um ihn in der Kundenkommunikation wieder zum Vorschein zu bringen. Dieser Wert kann beispielsweise in betriebliche E-Mails aufgenommen werden, um eindeutig zu identifizieren, welche Ihrer Anwendungen mit der Benachrichtigung verknüpft ist.

Standardmäßig gibt es keinen Wert.

Die Anwendungs-ID ist eine Zeichenfolge mit einer maximalen Länge von 50 Zeichen. Buchstaben, Zahlen und die folgenden Sonderzeichen sind zulässig:

! \$ % & * + - . , ^ _ ` | ~

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung [sdk_ua_app_id](#). Sie können die Anwendungs-ID nicht als Befehlszeilenoption angeben.

AWS_SECRET_ACCESS_KEY

Gibt den geheimen Schlüssel an, der mit dem Zugriffsschlüssel verknüpft ist. Dies ist im Wesentlichen das "Passwort" für den Zugriffsschlüssel.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `aws_secret_access_key`. Es ist nicht möglich, die geheime Zugriffsschlüssel-ID als Befehlszeilenoption anzugeben.

AWS_SESSION_TOKEN

Gibt den Sitzungstokenwert an, der erforderlich ist, wenn Sie temporäre Sicherheitsanmeldeinformationen verwenden, die Sie direkt aus AWS STS -Operationen abgerufen haben. Weitere Informationen finden Sie im [Ausgabeabschnitt des Befehls `assume-role`](#) im AWS CLI -Befehlsverweis.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `aws_session_token`.

AWS_SHARED_CREDENTIALS_FILE

Gibt den Speicherort der Datei an, die zum Speichern von Zugriffsschlüsseln AWS CLI verwendet wird. Der Standardpfad ist `~/.aws/credentials`.

Sie können diesen Wert nicht in einer benannten Profileinstellung oder mithilfe eines Befehlszeilenparameters angeben.

AWS_STS_REGIONAL_ENDPOINTS

Gibt an, wie der den AWS Dienstendpunkt AWS CLI bestimmt, den der AWS CLI Client für die Kommunikation mit dem AWS Security Token Service (AWS STS) verwendet. Für den Standardwert ist für AWS CLI Version 1 ist `legacy`.

Sie können einen von zwei Werten angeben:

- **legacy**— Verwendet den globalen STS Endpunkt `sts.amazonaws.com`, für die folgenden AWS Regionen: `ap-northeast-1` `ap-south-1` `ap-southeast-1` `ap-southeast-2`, `aws-global`, `ca-central-1`, `eu-central-1`, `eu-north-1`, `eu-west-1` `eu-west-2`, `eu-west-3`, `sa-east-1`, `us-east-1`, `us-east-2`, `us-west-1`, und `us-west-2`. Für alle anderen Regionen wird automatisch deren jeweiliger regionaler Endpunkt verwendet.

- **regional**— Der verwendet AWS CLI immer den AWS STS Endpunkt für die aktuell konfigurierte Region. Wenn der Client beispielsweise für die Verwendung konfiguriert ist `us-west-2`, AWS STS werden alle Aufrufe an den regionalen Endpunkt `sts.us-west-2.amazonaws.com` statt an den globalen `sts.amazonaws.com` Endpunkt getätigt. Um eine Anforderung an den globalen Endpunkt zu senden, während diese Einstellung aktiviert ist, können Sie die Region auf `aws-global` festlegen.

AWS_USE_DUALSTACK_ENDPOINT

Ermöglicht die Verwendung von Dual-Stack-Endpunkten zum Senden AWS von Anfragen. Weitere Informationen zu Dual-Stack-Endpunkten, die IPv4 sowohl IPv6 Datenverkehr als auch unterstützen, finden Sie unter [Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten](#) im Amazon Simple Storage Service-Benutzerhandbuch. Dual-Stack-Endpunkte sind für einige Services in einigen Regionen verfügbar. Wenn kein Dual-Stack-Endpunkt für den Service oder AWS-Region existiert, schlägt die Anfrage fehl. Diese ist standardmäßig deaktiviert.

Die Einstellungen für die Endpunktconfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktconfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. `AWS_ENDPOINT_URL_DYNAMODB`
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT](#)-, [AWS_USE_FIPS_ENDPOINT](#)- und [AWS_ENDPOINT_URL](#)-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint](#)-, [use_fips_endpoint](#)- und [endpoint_url](#)-Einstellungen.
8. Jeder Standard-Endpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen

verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

AWS_USE_FIPS_ENDPOINT

Einige AWS Dienste bieten Endgeräte, die in einigen Fällen den [Federal Information Processing Standard \(FIPS\) 140-2](#) unterstützen. AWS-Regionen Wenn der AWS Dienst dies unterstützt FIPS, gibt diese Einstellung an, welcher FIPS Endpunkt verwendet werden soll. AWS CLI Im Gegensatz zu AWS Standardendpunkten verwenden FIPS Endpunkte eine TLS Softwarebibliothek, die 140-2 entspricht FIPS. Diese Endpunkte können von Unternehmen erfordert werden, die mit der US-Regierung interagieren.

Wenn diese Einstellung aktiviert ist, aber kein FIPS Endpunkt für den Dienst in Ihrem vorhanden ist AWS-Region, schlägt der Befehl möglicherweise fehl. AWS Geben Sie in diesem Fall mithilfe der [--endpoint-url](#)-Option manuell den Endpunkt an, der im Befehl verwendet werden soll, oder verwenden Sie [servicespezifische Endpunkte](#).

Weitere Informationen zur Angabe von FIPS Endpunkten nach finden Sie AWS-Region unter [FIPSEndpunkte nach Dienst](#).

Die Einstellungen für die Endpunktconfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktconfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT](#)-, [AWS_USE_FIPS_ENDPOINT](#)- und [AWS_ENDPOINT_URL](#)-Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint](#)-, [use_fips_endpoint](#)- und [endpoint_url](#)-Einstellungen.

8. Jeder Standard-Endpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

[AWS_WEB_IDENTITY_TOKEN_FILE](#)

Gibt den Pfad zu einer Datei an, die ein OAuth 2.0-Zugriffstoken oder ein OpenID Connect-ID-Token enthält, das von einem Identitätsanbieter bereitgestellt wird. Die AWS CLI lädt den Inhalt dieser Datei und übergibt ihn als `WebIdentityToken`-Argument an die Operation `AssumeRoleWithWebIdentity`.

Wird mit `AWS_ROLE_ARN`- und `AWS_ROLE_SESSION_NAME`-Umgebungsvariablen genutzt.

Wenn diese Umgebungsvariable definiert ist, überschreibt sie den Wert der Profileinstellung `web_identity_token_file`.

Weitere Informationen zur Verwendung von Webidentitäten finden Sie unter [the section called "Übernehmen einer Rolle mit Web-Identität"](#).

Note

Diese Umgebungsvariable gilt nur für eine Übernahmerolle mit dem Web-Identitätsanbieter, sie gilt nicht für die allgemeine Konfiguration des Rollenanbieters.

Befehlszeilenoptionen in der AWS CLI

Bei den Befehlszeilenoptionen handelt es sich um globale Parameter AWS CLI, die Sie verwenden können, um die Standardkonfigurationseinstellungen, alle entsprechenden Profileinstellungen oder Umgebungsvariableneinstellungen für diesen einzelnen Befehl außer Kraft zu setzen. Sie können mit Befehlszeilenoptionen nicht direkt Anmeldeinformationen angeben. Sie können jedoch angeben, welches Profil verwendet werden soll.

Themen

- [Verwenden von Befehlszeilenoptionen](#)
- [AWS CLI unterstützte globale Befehlszeilenoptionen](#)
- [Häufige Verwendungsweisen von Befehlszeilenoptionen](#)

Verwenden von Befehlszeilenoptionen

Die meisten Befehlszeilenoptionen sind einfache Zeichenfolgen, wie der Profilname `profile1` im folgenden Beispiel:

```
$ aws s3 ls --profile profile1
amzn-s3-demo-bucket1
amzn-s3-demo-bucket2
...
```

Bei jeder Option, für die ein Argument angegeben wird, muss das Argument mit einem Leerzeichen oder Gleichheitszeichen (=) vom Namen der Option getrennt werden. Falls es sich bei dem Argumentwert um eine Zeichenfolge mit einem Leerzeichen handelt, muss das Argument in Anführungszeichen gesetzt werden. Weitere Informationen zu Argumenttypen und zur Formatierung für Parameter finden Sie unter [Angeben von Parameterwerten in der AWS CLI](#).

AWS CLI unterstützte globale Befehlszeilenoptionen

In der können AWS CLI Sie die folgenden Befehlszeilenoptionen verwenden, um die Standardkonfigurationseinstellungen, alle entsprechenden Profileinstellungen oder Umgebungsvariableneinstellungen für diesen einzelnen Befehl zu überschreiben.

`--ca-bundle` *<string>*

Gibt das Zertifikatspaket der Zertifizierungsstelle (CA) an, das bei der Überprüfung SSL von Zertifikaten verwendet werden soll.

Falls definiert, überschreibt diese Option den Wert für die Profileinstellung [ca_bundle](#) und die Umgebungsvariable [AWS_CA_BUNDLE](#).

`--cli-connect-timeout` *<integer>*

Gibt die maximale Socket-Verbindungszeit in Sekunden an. Wenn als Wert Null (0) festgelegt ist, wartet der Socket-Verbindungsvorgang unbegrenzt (blockiert) und es erfolgt keine Zeitüberschreitung.

`--cli-read-timeout` *<integer>*

Gibt die maximale Socket-Lesezeit in Sekunden an. Wenn als Wert Null (0) festgelegt ist, wartet der Socket-Lesevorgang unbegrenzt (blockiert) und es erfolgt keine Zeitüberschreitung.

--color <string>

Gibt Unterstützung für die Farbausgabe an. Gültige Werte sind on, off und auto. Der Standardwert ist auto.

--debug

Ein boolescher Schalter, der die Debug-Protokollierung ermöglicht. AWS CLI Standardmäßig werden in der Befehlsausgabe bereinigte Informationen zu Erfolgen oder Fehlschlägen in Bezug auf die Befehlsergebnisse angezeigt. Die --debug-Option stellt die vollständigen Python-Protokolle bereit. Dazu gehören zusätzliche diagnostische stderr-Informationen über die Befehlsausführung, die bei der Fehlerbehebung nützlich sein können, um herauszufinden, warum ein Befehl unerwartete Ergebnisse liefert. Um Debug-Protokolle einfach anzuzeigen, empfehlen wir, die Protokolle an eine Datei zu senden, um die Informationen einfacher zu durchsuchen. Sie können dies mit einer der folgenden Methoden durchführen.

Um nur die stderr-Diagnoseinformationen zu senden, fügen Sie 2> debug.txt an, wobei debug.txt der Name ist, den Sie für Ihre Debug-Datei verwenden möchten:

```
$ aws servicename commandname options --debug 2> debug.txt
```

Um beide, die Ausgangs- und stderr-Diagnoseinformationen zu senden, fügen Sie &> debug.txt an, wobei debug.txt der Name ist, den Sie für Ihre Debug-Datei verwenden möchten:

```
$ aws servicename commandname options --debug &> debug.txt
```

--endpoint-url <string>

Gibt an, an welche URL die Anfrage gesendet werden soll. Bei den meisten Befehlen bestimmt das AWS CLI automatisch auf der URL Grundlage des ausgewählten Dienstes und der angegebenen AWS Region. Bei einigen Befehlen müssen Sie jedoch einen kontospezifischen Wert angeben. URL Sie können einige AWS Dienste auch so konfigurieren, dass sie [einen Endpunkt direkt in Ihrem privaten Bereich hosten VPC](#), der dann möglicherweise angegeben werden muss.

Das folgende Befehlsbeispiel verwendet einen benutzerdefinierten Amazon S3 S3-EndpointURL.

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

Die Einstellungen für die Endpunktconfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktconfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT-](#), [AWS_USE_FIPS_ENDPOINT-](#) und [AWS_ENDPOINT_URL-](#)Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint-](#), [use_fips_endpoint-](#) und [endpoint_url](#)-Einstellungen.
8. Jeder Standard-Endpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

--no-paginate

Ein boolescher Schalter, der mehrere Aufrufe deaktiviert, die er automatisch durchführt, um alle Befehlsergebnisse zu empfangen, AWS CLI wodurch eine Seitennummerierung der Ausgabe erzeugt wird. Dies bedeutet, dass nur die erste Seite Ihrer Ausgabe angezeigt wird.

--no-sign-request

Ein boolescher Schalter, der das Signieren von Anfragen an den Dienstendpunkt deaktiviert. HTTP AWS Dadurch wird verhindert, dass Anmeldeinformationen geladen werden.

--no-verify-ssl

SSL Wird standardmäßig bei der Kommunikation mit AWS CLI Diensten verwendet. AWS Für jede SSL Verbindung und jeden Anruf AWS CLI überprüft der die SSL Zertifikate. Durch die

Verwendung dieser Option wird das Standardverhalten bei der Überprüfung SSL von Zertifikaten außer Kraft gesetzt.

⚠ Warning

Diese Option stellt keine bewährte Methode dar. Wenn Sie diese Option verwenden --no-verify-ssl, ist Ihr Datenverkehr zwischen Ihrem Client und den AWS Diensten nicht mehr gesichert. Das bedeutet, dass Ihr Datenverkehr ein Sicherheitsrisiko darstellt und anfällig für man-in-the-middle Exploits ist. Wenn Sie Probleme mit Zertifikaten haben, sollten Sie diese Probleme stattdessen lösen. Informationen zur Behebung von Zertifikatsfehlern finden Sie unter [the section called "SSLZertifikatsfehler"](#).

--output *<string>*

Gibt das Ausgabeformat an, das für diesen Befehl verwendet werden soll. Sie können alle folgenden Werte angeben:

- **[json](#)**— Die Ausgabe ist als Zeichenfolge formatiert. [JSON](#)
- **[text](#)** – Die Ausgabe wird als mehrere Zeilen mit tabulatorgetrennten Zeichenfolgenwerten formatiert. Dies kann nützlich sein, um die Ausgabe an einen Textprozessor wie `grep`, `sed` oder `awk` zu übergeben.
- **[table](#)** – Die Ausgabe erfolgt in Form einer Tabelle mit den Zeichen +|-, um die Zellenrahmen zu bilden. Normalerweise wird die Information in einem benutzerfreundlichen Format wiedergegeben, das viel einfacher zu lesen ist als die anderen, jedoch programmatisch nicht so nützlich ist.

--Profil *<string>*

Gibt das [benannte Profil](#) an, das für diesen Befehl verwendet werden soll. Zum Einrichten weiterer benannter Profile können Sie den Befehl `aws configure` mit der Option `--profile` verwenden.

```
$ aws configure --profile <profilename>
```

--query *<string>*

Gibt eine [JMESPathAbfrage](#) an, die beim Filtern der Antwortdaten verwendet werden soll. Weitere Informationen finden Sie unter [Ausgabe filtern in AWS CLI](#).

--Region <string>

Gibt an, an welche AWS Region die AWS Anforderung dieses Befehls gesendet werden soll. Eine Liste aller Regionen, die Sie angeben können, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

--version

Ein boolescher Schalter, der die aktuelle Version des AWS CLI Programms anzeigt, das gerade ausgeführt wird.

Häufige Verwendungsweisen von Befehlszeilenoptionen

Befehlszeilenoptionen werden häufig für die Überprüfung Ihrer Ressourcen in mehreren AWS -Regionen und zur Änderung des Ausgabeformats für bessere Lesbarkeit oder zur einfacheren Skripterstellung verwendet. In den folgenden Beispielen führen wir den Befehl `describe-instances` gegen jede Region aus, bis wir herausfinden, in welcher Region sich unsere Instance befindet.

```
$ aws ec2 describe-instances --output table --region us-west-1
-----
|DescribeInstances|
+-----+
$ aws ec2 describe-instances --output table --region us-west-2
-----
|                               DescribeInstances                               |
+-----+
||                               Reservations                               ||
|+-----+-----+
||  OwnerId                       |  012345678901                       ||
||  ReservationId                  |  r-abcdefgh                       ||
|+-----+-----+
|||                               Instances                               |||
||+-----+-----+
|||  AmiLaunchIndex                |  0                                 |||
|||  Architecture                  |  x86_64                           |||
...

```

Konfiguration der Befehlsvervollständigung in AWS CLI

Das AWS Command Line Interface (AWS CLI) beinhaltet eine Bash-kompatible Funktion zur Befehlsvervollständigung, mit der Sie die Tabulatortaste verwenden können, um einen teilweise

einggegebenen Befehl abzuschließen. Auf den meisten Systemen müssen Sie dies manuell konfigurieren.

Themen

- [Funktionsweise](#)
- [Konfigurieren der Befehlsvervollständigung unter Linux oder macOS](#)
- [Konfigurieren der Befehlsvervollständigung unter Windows](#)

Funktionsweise

Wenn Sie einen Befehl, einen Parameter oder eine Option teilweise eingeben, wird der Befehl von der Befehlsvervollständigungsfunktion entweder automatisch vervollständigt oder von dieser eine Liste mit vorgeschlagenen Befehlen angezeigt. Um die Befehlsvervollständigung aufzufordern, geben Sie einen Befehl teilweise ein und drücken dann die Vervollständigungstaste, was in der Regel **Tab** in den meisten Shells.

Die folgenden Beispiele zeigen verschiedene Möglichkeiten, wie Sie die Befehlsvervollständigung nutzen können:

- Geben Sie teilweise einen Befehl ein und drücken Sie **Tab** um eine vorgeschlagene Liste von Befehlen anzuzeigen.

```
$ aws dynamodb dTAB
delete-backup                describe-global-table
delete-item                  describe-global-table-settings
delete-table                 describe-limits
describe-backup              describe-table
describe-continuous-backups describe-table-replica-auto-scaling
describe-contributor-insights describe-time-to-live
describe-endpoints
```

- Geben Sie teilweise einen Parameter ein und drücken Sie **Tab** um eine vorgeschlagene Liste von Parametern anzuzeigen.

```
$ aws dynamodb delete-table --TAB
--ca-bundle                --endpoint-url          --profile
--cli-connect-timeout     --generate-cli-skeleton --query
--cli-input-json          --no-paginate           --region
--cli-read-timeout        --no-sign-request       --table-name
```

```
--color          --no-verify-ssl    --version
--debug          --output
```

- Geben Sie einen Parameter ein und drücken Sie **Tab**, um eine vorgeschlagene Liste von Ressourcenwerten anzuzeigen. Diese Funktion ist nur in der AWS CLI Version 2 verfügbar.

```
$ aws dynamodb db delete-table --table-name TAB
Table 1          Table 2          Table 3
```

Konfigurieren der Befehlsvervollständigung unter Linux oder macOS

Um die Befehlsvervollständigung unter Linux oder macOS zu konfigurieren, müssen Sie den Namen der verwendeten Shell und den Speicherort des `aws_completer`-Skripts kennen.

Note

Die Befehlsvervollständigung wird auf EC2 Amazon-Instances, auf denen Amazon Linux ausgeführt wird, standardmäßig automatisch konfiguriert und aktiviert.

Themen

- [Bestätigen Sie, dass sich der Ordner der Vervollständigung in Ihrem Pfad befindet](#)
- [Aktivieren der Befehlsvervollständigung](#)
- [Verifizieren der Befehlsvervollständigung](#)

Bestätigen Sie, dass sich der Ordner der Vervollständigung in Ihrem Pfad befindet

Damit der AWS Completer erfolgreich funktioniert, `aws_completer` muss er sich im Pfad Ihrer Shell befinden. Der `which`-Befehl kann überprüfen, ob sich die Vervollständigung in Ihrem Pfad befindet.

```
$ which aws_completer
/usr/local/bin/aws_completer
```

Wenn der Befehl die Vervollständigung nicht finden kann, führen Sie die folgenden Schritte aus, um den Ordner der Vervollständigung zu Ihrem Pfad hinzuzufügen.

Schritt 1: Suchen Sie den Completer AWS

Der Standort des AWS Completers kann je nach verwendeter Installationsmethode variieren.

- Paketmanager — Programme wie `pip`, `yumbrew`, und installieren den AWS Completer (oder einen Symlink dazu) `apt-get` normalerweise an einem Standardpfad.
 - Wenn Sie `pip` ohne den Parameter `--user` verwendet haben, ist der Standardpfad `/usr/local/bin/aws_completer`.
 - Wenn Sie `pip` ohne den Parameter `--user` verwendet haben, ist der Standardpfad `/home/username/.local/bin/aws_completer`.
- Gebündeltes Installationsprogramm – Wenn Sie das mitgelieferte Installationsprogramm verwendet haben, ist der Standardpfad `/usr/local/bin/aws_completer`.

Wenn alles andere fehlschlägt, können Sie den `find` Befehl verwenden, um Ihr Dateisystem nach dem AWS Completer zu durchsuchen.

```
$ find / -name aws_completer
/usr/local/bin/aws_completer
```

Schritt 2: Identifizieren Ihrer Shell

Um zu ermitteln, welche Shell Sie verwenden, können Sie einen der folgenden Befehle verwenden.

- `echo $SHELL` — Zeigt den Namen der Programmdatei der Shell an. Dieser wird in der Regel für die verwendete Shell korrekt sein, sofern Sie keine andere Shell nach dem Anmelden gestartet haben.

```
$ echo $SHELL
/bin/bash
```

- `ps` – zeigt die für den aktuellen Benutzer ausgeführten Prozesse an. Einer von ihnen ist die Shell.

```
$ ps
  PID TTY          TIME CMD
 2148 pts/1    00:00:00 bash
 8756 pts/1    00:00:00 ps
```

Schritt 3: Fügen Sie die Vervollständigung zu Ihrem Pfad hinzu

1. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner.

```
$ ls -a ~/
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
 - Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc` oder `.login`
2. Fügen Sie einen Exportbefehl an das Ende Ihres Profilskripts hinzu und orientieren Sie sich dabei an folgendem Beispiel. Ersetzen Sie `/usr/local/bin/` durch den Ordner, den Sie im vorherigen Abschnitt entdeckt haben.

```
export PATH=/usr/local/bin/:$PATH
```

3. Laden Sie das Profil erneut in die aktuelle Sitzung, damit die Änderungen wirksam werden. Ersetzen Sie `.bash_profile` durch den Namen des Shell-Skripts, das Sie im ersten Abschnitt entdeckt haben.

```
$ source ~/.bash_profile
```

Aktivieren der Befehlsvervollständigung

Nachdem Sie bestätigt haben, dass sich die Vervollständigung in Ihrem Pfad befindet, aktivieren Sie die Befehlsvervollständigung, indem Sie den entsprechenden Befehl für die von Ihnen verwendete Shell ausführen. Sie können den Befehl des Profils Ihrer Shell hinzufügen, sodass er immer ausgeführt wird, wenn Sie eine neue Shell öffnen. Ersetzen Sie in jedem Befehl den `/usr/local/bin/` Pfad durch den Pfad, den Sie auf Ihrem System gefunden haben, in [Bestätigen Sie, dass sich der Ordner der Vervollständigung in Ihrem Pfad befindet](#).

- **bash** – Verwenden Sie den integrierten Befehl `complete`.

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

Fügen Sie den vorherigen Befehl `~/ .bashrc` hinzu, sodass er immer ausgeführt wird, wenn Sie eine neue Shell öffnen. Ihr `~/ .bash_profile` sollte `~/ .bashrc` bereitstellen, um sicherzustellen, dass der Befehl auch in Anmelde-Shell ausgeführt wird.

- **zsh** – um die Befehlsvervollständigung auszuführen, müssen Sie `bashcompinit` ausführen, indem Sie die folgende Autoload-Zeile am Ende Ihres `~/ .zshrc`-Profilskripts hinzufügen.

```
$ autoload bashcompinit && bashcompinit
$ autoload -Uz compinit && compinit
```

Um die Befehlsvervollständigung zu aktivieren, verwenden Sie den integrierten Befehl `complete`.

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

Fügen Sie die vorherigen Befehle `~/ .zshrc` hinzu, sodass sie immer ausgeführt werden, wenn Sie eine neue Shell öffnen.

- **tcsh** – Die Vervollständigung für `tcsh` erwartet einen Worttyp und ein Muster, um das Vervollständigungsverhalten zu definieren.

```
> complete aws 'p/*/'`aws_completer`/'
```

Fügen Sie den vorherigen Befehl `~/ .tcshrc` hinzu, sodass er immer ausgeführt wird, wenn Sie eine neue Shell öffnen.

Nachdem Sie die Befehlsvervollständigung aktiviert haben, funktioniert [Verifizieren der Befehlsvervollständigung](#).

Verifizieren der Befehlsvervollständigung

Nachdem Sie die Befehlsvervollständigung aktiviert haben, laden Sie Ihre Shell neu, geben Sie einen Teilbefehl ein und drücken Sie die Tab-Taste, um die verfügbaren Befehle anzuzeigen.

```
$ aws sTAB
s3          ses          sqs          sts          swf
s3api       sns          storagegateway support
```

Konfigurieren der Befehlsvervollständigung unter Windows

Note

Informationen dazu, wie PowerShell mit ihrer Vervollständigung umgegangen wird, einschließlich ihrer verschiedenen Vervollständigungsschlüssel, finden Sie unter [about_Tab_Expansion](#) in den Microsoft Docs. PowerShell

Um die Befehlsvervollständigung für PowerShell unter Windows zu aktivieren, führen Sie die folgenden Schritte unter aus. PowerShell

1. Öffnen Sie Ihr \$PROFILE mit dem folgenden Befehl.

```
PS C:\> Notepad $PROFILE
```

Wenn Sie kein \$PROFILE haben, erstellen Sie mit dem folgenden Befehl ein Benutzerprofil.

```
PS C:\> if (!(Test-Path -Path $PROFILE ))  
{ New-Item -Type File -Path $PROFILE -Force }
```

Weitere Informationen zu PowerShell Profilen finden Sie unter [So verwenden Sie Profile in Windows PowerShell ISE](#) auf der Microsoft Docs-Website.

2. Um die Befehlsvervollständigung zu aktivieren, fügen Sie dem Profil den folgenden Codeblock hinzu, speichern und schließen Sie die Datei.

```
Register-ArgumentCompleter -Native -CommandName aws -ScriptBlock {  
    param($commandName, $wordToComplete, $cursorPosition)  
    $env:COMP_LINE=$wordToComplete  
    if ($env:COMP_LINE.Length -lt $cursorPosition){  
        $env:COMP_LINE=$env:COMP_LINE + " "  
    }  
    $env:COMP_POINT=$cursorPosition  
    aws_completer.exe | ForEach-Object {  
        [System.Management.Automation.CompletionResult]::new($_, $_,  
'ParameterValue', $_)  
    }  
    Remove-Item Env:\COMP_LINE  
    Remove-Item Env:\COMP_POINT
```



```
}
```

- Nachdem Sie die Befehlsvervollständigung aktiviert haben, laden Sie Ihre Shell neu, geben Sie einen Teilbefehl ein und drücken Sie die Tab-Taste, um die verfügbaren Befehle zu durchlaufen.

```
$ aws sTab
```

```
$ aws s3
```

Um alle verfügbaren Befehle zu sehen, die zur Vervollständigung verfügbar sind, geben Sie einen Teil eines Befehls ein und drücken Sie Strg + Leerzeichen.

```
$ aws sCtrl + Space
s3          ses          sqs          sts          swf
s3api       sns          storagegateway support
```

AWS CLI Wiederholungen in der AWS CLI

In diesem Thema wird beschrieben, wie AWS CLI Dienstanrufe aufgrund AWS unerwarteter Probleme fehlschlagen können. Diese Probleme können serverseitig auftreten oder auf eine Ratenbegrenzung des AWS -Services, den Sie aufrufen möchten, zurückzuführen sein. Solche Ausfälle erfordern in der Regel keine besondere Behandlung und der Aufruf wird, oft nach einer kurzen Wartezeit, automatisch erneut getätigt. Das AWS CLI bietet viele Funktionen, mit denen Sie versuchen können, Kundenanrufe bei AWS Diensten zu wiederholen, wenn solche Fehler oder Ausnahmen auftreten.

Themen

- [Verfügbare Wiederholungsmodi](#)
- [Konfigurieren eines Wiederholungsversuchsmodus](#)
- [Anzeigen von Protokollen von Wiederholungsversuchen](#)

Verfügbare Wiederholungsmodi

Je nach Version stehen mehrere Modi zur Auswahl: AWS CLI

- [Legacy-Wiederholungsmodus](#)

- [Standardmodus für die Wiederholung](#)
- [Adaptiver Wiederholungsmodus](#)

Legacy-Wiederholungsmodus

Der Legacy-Modus ist der Standardmodus, der von AWS CLI Version 1 verwendet wird. Der Legacy-Modus verwendet einen älteren Wiederholungs-Handler mit eingeschränkter Funktionalität, der Folgendes umfasst:

- Einen Standardwert von 4 für maximale Wiederholungsversuche, was insgesamt 5 Aufrufversuche ergibt. Dieser Wert kann durch den Konfigurationsparameter `max_attempts` überschrieben werden.
- Dynamo DB hat einen Standardwert von 9 für maximale Wiederholungsversuche, was insgesamt 10 Aufrufversuche ergibt. Dieser Wert kann durch den Konfigurationsparameter `max_attempts` überschrieben werden.
- Wiederholungsversuche für die folgende begrenzte Anzahl von Fehlern/Ausnahmen:
 - Allgemeine Socket-/Verbindungsfehler:
 - `ConnectionError`
 - `ConnectionClosedError`
 - `ReadTimeoutError`
 - `EndpointConnectionError`
 - Serviceseitige Fehler und Ausnahmen durch Drosselung/Begrenzung:
 - `Throttling`
 - `ThrottlingException`
 - `ThrottledException`
 - `RequestThrottledException`
 - `ProvisionedThroughputExceededException`
- Versuchen Sie es erneut mit mehreren HTTP Statuscodes, darunter 429, 500, 502, 503, 504 und 509.
- Jeder Wiederholungsversuch enthält ein exponentielles Backoff um den Basisfaktor 2.

Standardmodus für die Wiederholung

Der Standardmodus ist ein Standardsatz von Wiederholungsregeln für alle, die mehr Funktionen bieten als die AWS SDKs Vorgängerversion. Der Standardmodus wurde für AWS CLI Version 2 erstellt und auf AWS CLI Version 1 zurückportiert. Die Funktionalität des Standardmodus umfasst:

- Ein Standardwert von 2 für maximale Wiederholungsversuche, was insgesamt 3 Anrufversuche ergibt. Dieser Wert kann durch den Konfigurationsparameter `max_attempts` überschrieben werden.
- Wiederholungsversuche für die folgende erweiterte Liste von Fehlern/Ausnahmen:
 - Transiente Fehler/Ausnahmen
 - `RequestTimeout`
 - `RequestTimeoutException`
 - `PriorRequestNotComplete`
 - `ConnectionError`
 - `HTTPClientError`
 - Serviceseitige Fehler und Ausnahmen durch Drosselung/Begrenzung:
 - `Throttling`
 - `ThrottlingException`
 - `ThrottledException`
 - `RequestThrottledException`
 - `TooManyRequestsException`
 - `ProvisionedThroughputExceededException`
 - `TransactionInProgressException`
 - `RequestLimitExceeded`
 - `BandwidthLimitExceeded`
 - `LimitExceededException`
 - `RequestThrottled`
 - `SlowDown`
 - `EC2ThrottledException`

- Wiederholungsversuche für nicht beschreibende, transiente Fehlercodes. Insbesondere diese HTTP Statuscodes: 500, 502, 503, 504.
- Jeder Wiederholungsversuch beinhaltet ein exponentielles Backoff um einen Basisfaktor von 2 für eine maximale Backoff-Zeit von 20 Sekunden.

Adaptiver Wiederholungsmodus

Warning

Der adaptive Modus ist ein experimenteller Modus und kann sich sowohl in den Funktionen als auch im Verhalten ändern.

Der adaptive Wiederholungsmodus ist ein experimenteller Wiederholungsmodus, der alle Funktionen des Standardmodus enthält. Zusätzlich zu den Standardmodusfunktionen führt der adaptive Modus auch die clientseitige Ratenbegrenzung durch die Verwendung eines Token-Buckets und Ratenbegrenzungsvariablen ein, die bei jedem Wiederholungsversuch dynamisch aktualisiert werden. Dieser Modus bietet Flexibilität bei clientseitigen Wiederholungsversuchen und passt sich an die Fehler-/Ausnahmestatusantwort eines Dienstes an. AWS

Bei jedem erneuten Versuch ändert der adaptive Modus die Variablen für die Ratenbegrenzung auf der Grundlage des in der Antwort des Dienstes angegebenen Fehler-, Ausnahme- oder HTTP Statuscode. AWS Diese Ratenbegrenzungsvariablen werden dann verwendet, um eine neue Aufruftrate für den Client zu berechnen. Jede Ausnahmen/Fehler- oder HTTP Nichterfolgsantwort (in der obigen Liste aufgeführt) eines AWS Dienstes aktualisiert die Variablen für die Ratenbegrenzung bei Wiederholungsversuchen, bis der Erfolg erreicht ist, der Token-Bucket erschöpft ist oder der konfigurierte Wert für maximale Versuche erreicht ist.

Konfigurieren eines Wiederholungsversuchsmodus

AWS CLI Dazu gehören eine Vielzahl von Wiederholungskonfigurationen und Konfigurationsmethoden, die Sie bei der Erstellung Ihres Client-Objekts berücksichtigen sollten.

Verfügbare Konfigurationsmethoden

In der AWS CLI können Benutzer Wiederholungsversuche auf folgende Weise konfigurieren:

- Umgebungsvariablen

- AWS CLI Konfigurationsdatei

Die folgenden Optionen für Wiederholungsversuche können angepasst werden:

- Wiederholungsmodus — Gibt an, welchen Wiederholungsmodus der AWS CLI Benutzer verwendet. Wie zuvor beschrieben, stehen drei Wiederholungsmodi zur Verfügung: Legacy, Standard und Adaptive. Der Standardwert für AWS CLI Version 1 ist AWS CLI Legacy-Version Standard.
- Max. Versuche — Gibt den Wert der maximalen Anzahl von Wiederholungsversuchen an, die der AWS CLI Wiederholungshandler verwendet, wobei der erste Aufruf auf den von Ihnen angegebenen Wert angerechnet wird. Der Standardwert ist 5.

Definieren einer Wiederholungskonfiguration in Ihren Umgebungsvariablen

Um Ihre Wiederholungskonfiguration für zu definieren AWS CLI, aktualisieren Sie die Umgebungsvariablen Ihres Betriebssystems.

Die Umgebungsvariablen für den Wiederholungsversuch sind:

- AWS_RETRY_MODE
- AWS_MAX_ATTEMPTS

Weitere Informationen zu Umgebungsvariablen finden Sie unter [Konfiguration von Umgebungsvariablen für AWS CLI](#).

Anzeigen von Protokollen von Wiederholungsversuchen

Das AWS CLI verwendet die Wiederholungsmethode und Protokollierung von Boto3. Sie können die `--debug`-Option für einen beliebigen Befehl benutzen, um Debug-Protokolle zu empfangen. Weitere Informationen zur Verwendung der Option `--debug` finden Sie unter [Befehlszeilenoptionen in der AWS CLI](#).

Wenn Sie in den Debug-Protokollen nach „retry“ suchen, finden Sie die erforderlichen Wiederholungsinformationen. Die Clientprotokolleinträge für Wiederholungsversuche hängen davon ab, welchen Wiederholungsmodus Sie aktiviert haben.

Legacy-Modus:

Wiederholungsmeldungen werden von `botocore.retryhandler` generiert. Es wird eine von drei Meldungen angezeigt:

- No retry needed
- Retry needed, action of: `<action_name>`
- Reached the maximum number of retry attempts: `<attempt_number>`

Standard- oder adaptiver Modus:

Wiederholungsmeldungen werden von `botocore.retries.standard` generiert. Es wird eine von drei Meldungen angezeigt:

- No retrying request
- Retry needed, retrying request after delay of: `<delay_value>`
- Retry needed but retry quota reached, not retrying request

[Die vollständige Definitionsdatei der Botocore-Wiederholungen finden Sie unter `_retry.json` im Botocore-Repository. GitHub](#)

Verwendung eines HTTP Proxys für AWS CLI

Für den Zugriff AWS über Proxyserver können Sie die `HTTPS_PROXY` Umgebungsvariablen `HTTP_PROXY` und entweder mit den DNS Domännennamen oder den IP-Adressen und Portnummern konfigurieren, die Ihre Proxyserver verwenden.

Themen

- [Verwenden der -Beispiele](#)
- [Authentifizieren bei einem Proxy](#)
- [Verwendung eines Proxys auf EC2 Amazon-Instances](#)
- [Fehlerbehebung](#)

Verwenden der -Beispiele

Note

Die folgenden Beispiele zeigen den Namen der Umgebungsvariablen in Großbuchstaben. Wenn Sie jedoch zweimal eine Variable in Groß- und in Kleinbuchstaben angeben, haben die Kleinbuchstaben Vorrang. Wir empfehlen, jede Variable nur einmal zu definieren, um Verwirrung und unerwartetes Verhalten zu vermeiden.

Die folgenden Beispiele zeigen, wie Sie entweder die explizite IP-Adresse Ihres Proxys oder einen DNS Namen verwenden können, der in die IP-Adresse Ihres Proxys aufgelöst wird. In beiden Fällen können ein Doppelpunkt und die Portnummer folgen, an die Abfragen gesendet werden sollen.

Linux or macOS

```
$ export HTTP_PROXY=http://10.15.20.25:1234
$ export HTTP_PROXY=http://proxy.example.com:1234
$ export HTTPS_PROXY=http://10.15.20.25:5678
$ export HTTPS_PROXY=http://proxy.example.com:5678
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx HTTP_PROXY http://10.15.20.25:1234
C:\> setx HTTP_PROXY http://proxy.example.com:1234
C:\> setx HTTPS_PROXY http://10.15.20.25:5678
C:\> setx HTTPS_PROXY http://proxy.example.com:5678
```

Bei Verwendung von [setx](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert in der aktuellen Eingabeaufforderungssitzung und allen nach Ausführung des Befehls erstellten Eingabeaufforderungssitzungen geändert. Andere Befehls-Shells, die zum Zeitpunkt der Befehlsausführung bereits ausgeführt werden, sind hiervon nicht betroffen.

Einrichten nur für die aktuelle Sitzung

Bei Verwendung von [set](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert bis zum Ende der aktuellen Eingabeaufforderungssitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert.

```
C:\> set HTTP_PROXY=http://10.15.20.25:1234
C:\> set HTTP_PROXY=http://proxy.example.com:1234
C:\> set HTTPS_PROXY=http://10.15.20.25:5678
C:\> set HTTPS_PROXY=http://proxy.example.com:5678
```

Authentifizieren bei einem Proxy

Note

Der unterstützt AWS CLI keine NTLM Proxys. [Wenn Sie einen Proxy NTLM oder das Kerberos-Protokoll verwenden, können Sie möglicherweise eine Verbindung über einen Authentifizierungsproxy wie Cntlm herstellen.](#)

Der AWS CLI unterstützt die Standardauthentifizierung. HTTP Geben Sie den Benutzernamen und das Passwort im Proxy URL wie folgt an.

Linux or macOS

```
$ export HTTP_PROXY=http://username:password@proxy.example.com:1234
$ export HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx HTTP_PROXY http://username:password@proxy.example.com:1234
C:\> setx HTTPS_PROXY http://username:password@proxy.example.com:5678
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set HTTP_PROXY=http://username:password@proxy.example.com:1234
C:\> set HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Verwendung eines Proxys auf EC2 Amazon-Instances

Wenn Sie einen Proxy auf einer EC2 Amazon-Instance konfigurieren, die mit einer angehängten IAM Rolle gestartet wurde, stellen Sie sicher, dass Sie die Adresse ausnehmen, die für den Zugriff auf die

[Instance-Metadaten](#) verwendet wird. Legen Sie dazu die Umgebungsvariable `NO_PROXY` auf die IP-Adresse des Instance-Metadaten-Services, `169.254.169.254`, fest. Diese Adresse variiert nicht.

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx NO_PROXY 169.254.169.254
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set NO_PROXY=169.254.169.254
```

Fehlerbehebung

Wenn Sie Probleme mit dem haben AWS CLI, finden Sie unter Schritte [Beheben von Fehlern](#) zur Fehlerbehebung weitere Informationen. Die wichtigsten Maßnahmen zur Fehlerbehebung finden Sie unter [the section called “SSLZertifikatsfehler”](#).

Verwenden von Endpunkten in der AWS CLI

Um programmgesteuert eine Verbindung zu einem herzustellen AWS-Service, verwenden Sie einen Endpunkt. Ein Endpunkt ist URL der Einstiegspunkt für einen AWS Webdienst. Die AWS Command Line Interface (AWS CLI) verwendet automatisch den Standardendpunkt für jeden Dienst in einem AWS-Region, aber Sie können einen alternativen Endpunkt für Ihre API Anfragen angeben.

Themen zu Endpunkten

- [Festlegen eines Endpunkts für einen einzelnen Befehl](#)
- [Legen Sie den globalen Endpunkt für alle fest AWS-Services](#)
- [Stellen Sie ein, dass FIPs Endpunkte für alle verwendet werden AWS-Services](#)
- [So einstellen, dass Dual-Stack-Endpunkte für alle AWS-Services verwendet werden](#)
- [Festlegen servicespezifischer Endpunkte](#)
 - [Servicespezifische Endpunkte: Umgebungsvariablen](#)

- [Servicespezifische Endpunkte: freigegebene config-Datei](#)
- [Servicespezifische Endpunkte: Liste servicespezifischer Kennungen](#)
- [Priorität der Endpunktconfiguration und der Einstellungen](#)

Festlegen eines Endpunkts für einen einzelnen Befehl

Verwenden Sie die Befehlszeilenoption `--endpoint-url`, um Endpunkteinstellungen oder Umgebungsvariablen für einen einzelnen Befehl zu überschreiben. Das folgende Befehlsbeispiel verwendet einen benutzerdefinierten Amazon S3 S3-EndpointURL.

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

Legen Sie den globalen Endpunkt für alle fest AWS-Services

Verwenden Sie eine der folgenden EinstellungenURL, um Anfragen für alle Dienste an einen benutzerdefinierten Endpunkt weiterzuleiten:

- Umgebungsvariablen:
 - [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#)- Ignoriere den konfigurierten EndpointURLs.

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS_ENDPOINT_URL](#)- Legen Sie den globalen Endpunkt festURL.

Linux or macOS

```
$ export AWS_ENDPOINT_URL=http://localhost:4567
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_ENDPOINT_URL http://localhost:4567
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set AWS_ENDPOINT_URL=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL="http://localhost:4567"
```

- Die Datei config:
 - [ignore_configure_endpoint_urls](#)- Ignoriere den konfigurierten EndpunktURLs.

```
ignore_configure_endpoint_urls = true
```

- [endpoint_url](#)- Legen Sie den globalen Endpunkt festURL.

```
endpoint_url = http://localhost:4567
```

Servicespezifische Endpunkte und die Befehlszeilenoption `--endpoint-url` haben Vorrang vor allen globalen Endpunkten.

Stellen Sie ein, dass FIPs Endpunkte für alle verwendet werden AWS-Services

Verwenden Sie eine der folgenden Methoden, um Anfragen für alle Dienste zur Verwendung von FIPs Endpunkten weiterzuleiten:

- [AWS_USE_FIPS_ENDPOINT](#)-Umgebungsvariable.

Linux or macOS

```
$ export AWS_USE_FIPS_ENDPOINT=true
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_USE_FIPS_ENDPOINT true
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set AWS_USE_FIPS_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_FIPS_ENDPOINT="true"
```

- [use_fips_endpoint](#)-Dateieinstellung.

```
use_fips_endpoint = true
```

Einige AWS Dienste bieten Endgeräte, die in einigen Fällen den [Federal Information Processing Standard \(FIPS\) 140-2](#) unterstützen. AWS-Regionen Wenn der AWS Dienst dies unterstützt FIPS, gibt diese Einstellung an, welcher FIPS Endpunkt verwendet werden soll. AWS CLI Im Gegensatz zu AWS Standardendpunkten verwenden FIPS Endpunkte eine TLS Softwarebibliothek, die 140-2 entspricht FIPS. Diese Endpunkte können von Unternehmen erfordert werden, die mit der US-Regierung interagieren.

Wenn diese Einstellung aktiviert ist, aber kein FIPS Endpunkt für den Dienst in Ihrem vorhanden ist AWS-Region, schlägt der Befehl möglicherweise fehl. AWS Geben Sie in diesem Fall mithilfe der [--endpoint-url](#)-Option manuell den Endpunkt an, der im Befehl verwendet werden soll, oder verwenden Sie [servicespezifische Endpunkte](#).

Weitere Informationen zur Angabe von FIPS Endpunkten nach finden Sie AWS-Region unter [FIPSEndpunkte nach Dienst](#).

So einstellen, dass Dual-Stack-Endpunkte für alle AWS-Services verwendet werden

Verwenden Sie eine der folgenden Einstellungen, um Anforderungen für alle Services so weiterzuleiten, dass Dual-Stack-Endpunkte verwendet werden:

- [AWS_USE_DUALSTACK_ENDPOINT](#)-Umgebungsvariable.

Linux or macOS

```
$ export AWS_USE_DUALSTACK_ENDPOINT=true
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set AWS_USE_DUALSTACK_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_DUALSTACK_ENDPOINT="true"
```

- [use_dualstack_endpoint](#)-Dateieinstellung.

```
use_dualstack_endpoint = true
```

Ermöglicht die Verwendung von Dual-Stack-Endpunkten zum Senden von Anfragen. AWS Weitere Informationen zu Dual-Stack-Endpunkten, die IPv4 sowohl IPv6 Datenverkehr als auch unterstützen, finden Sie unter [Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten](#) im Amazon Simple Storage Service-Benutzerhandbuch. Dual-Stack-Endpunkte sind für einige Services in einigen Regionen verfügbar. Wenn kein Dual-Stack-Endpunkt für den Service oder AWS-Region existiert, schlägt die Anfrage fehl. Diese ist standardmäßig deaktiviert.

Festlegen servicespezifischer Endpunkte

Die dienstspezifische Endpunktkonfiguration bietet die Option, einen persistenten Endpunkt Ihrer Wahl für AWS CLI Anfragen zu verwenden. Diese Einstellungen bieten Flexibilität bei der Unterstützung lokaler Endpunkte, VPC Endpunkte und lokaler AWS Entwicklungsumgebungen von Drittanbietern. Verschiedene Endpunkte können für Test- und Produktionsumgebungen verwendet werden. Sie können einen Endpunkt URL für eine einzelne Person angeben. AWS-Services

Servicespezifische Endpunkte können wie folgt festgelegt werden:

- Mit der Befehlszeilenoption [--endpoint-url](#) für einen einzelnen Befehl.
- Umgebungsvariablen:
 - [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#)- Ignoriert alle konfigurierten EndpunkteURLs, sofern sie nicht in der Befehlszeile angegeben wurden.
 - [AWS_ENDPOINT_URL_<SERVICE>](#) – Gibt einen benutzerdefinierten Endpunkt an, der für einen bestimmten Service verwendet wird, wobei <SERVICE> durch die AWS-Service -Kennung ersetzt wird. Informationen zu allen servicespezifischen Variablen finden Sie unter [the section called "Liste der servicespezifischen Kennungen"](#).
- config-Datei:
 - [ignore_configure_endpoint_urls](#)- Ignoriert alle konfigurierten EndpunkteURLs, sofern sie nicht mithilfe von Umgebungsvariablen oder in der Befehlszeile angegeben wurden.
 - Der Abschnitt [services](#) der config-Datei in Kombination mit der Dateieinstellung [endpoint_url](#).

Themen zu servicespezifischen Endpunkten:

- [Servicespezifische Endpunkte: Umgebungsvariablen](#)
- [Servicespezifische Endpunkte: freigegebene config-Datei](#)
- [Servicespezifische Endpunkte: Liste servicespezifischer Kennungen](#)

Servicespezifische Endpunkte: Umgebungsvariablen

Umgebungsvariablen überschreiben die Einstellungen in Ihrer Konfigurationsdatei, aber nicht die in der Befehlszeile angegebenen Optionen. Verwenden Sie Umgebungsvariablen, wenn Sie möchten, dass alle Profile dieselben Endpunkte auf Ihrem Gerät verwenden.

Die folgenden Umgebungsvariablen sind servicespezifisch:

- [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#)- Ignoriert alle konfigurierten EndpunkteURLs, sofern sie nicht in der Befehlszeile angegeben wurden.

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS_ENDPOINT_URL_<SERVICE>](#)- Gibt einen benutzerdefinierten Endpunkt an, der für einen bestimmten Dienst verwendet <SERVICE> wird, wobei dieser durch den AWS-Service Bezeichner ersetzt wird. Informationen zu allen servicespezifischen Variablen finden Sie unter [the section called "Liste der servicespezifischen Kennungen"](#).

In den folgenden Beispielen für Umgebungsvariablen wird ein Endpunkt für AWS Elastic Beanstalk festgelegt:

Linux or macOS

```
$ export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

Windows Command Prompt

Einrichten für alle Sitzungen

```
C:\> setx AWS_ENDPOINT_URL_ELASTIC_BEANSTALK http://localhost:4567
```

Einrichten nur für die aktuelle Sitzung

```
C:\> set AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL_ELASTIC_BEANSTALK="http://localhost:4567"
```

Weitere Informationen zum Festlegen von Umgebungsvariablen finden Sie unter [the section called “Umgebungsvariablen”](#).

Servicespezifische Endpunkte: freigegebene **config**-Datei

In der freigegebenen config-Datei wird `endpoint_url` in mehreren Abschnitten verwendet. Wenn Sie einen servicespezifischen Endpunkt festlegen möchten, verwenden Sie die Einstellung `endpoint_url`, die unter einem Service-ID-Schlüssel innerhalb eines `services`-Abschnitts verschachtelt ist. Details zur Definition eines `services`-Abschnitts in Ihrer freigegebenen config-Datei finden Sie unter [the section called “services”](#).

Das folgende Beispiel verwendet einen `services` Abschnitt zur Konfiguration eines dienstspezifischen Endpunkts URL für Amazon S3 und eines benutzerdefinierten globalen Endpunkts, der für alle anderen Dienste verwendet wird:

```
[profile dev1]  
endpoint_url = http://localhost:1234  
services = s3-specific  
  
[services testing-s3]  
s3 =  
  endpoint_url = http://localhost:4567
```

Mit einem einzigen Profil können Endpunkte für mehrere Services konfiguriert werden. Im folgenden Beispiel wird der dienstspezifische Endpunkt URLs für Amazon S3 und AWS Elastic Beanstalk im selben Profil festgelegt.

Eine Liste aller Service-ID-Schlüssel, die im `services`-Abschnitt verwendet werden können, finden Sie unter [Liste der servicespezifischen Kennungen](#).

```
[profile dev1]  
services = testing-s3-and-eb
```



```
[services testing-s3-and-eb]  
s3 =  
  endpoint_url = http://localhost:4567  
elastic_beanstalk =  
  endpoint_url = http://localhost:8000
```

Der Abschnitt zur Servicekonfiguration kann in mehreren Profilen verwendet werden. Im folgenden Beispiel verwenden zwei Profile dieselbe `services`-Definition:

```
[profile dev1]  
output = json  
services = testing-s3  
  
[profile dev2]  
output = text  
services = testing-s3  
  
[services testing-s3]  
s3 =  
  endpoint_url = https://localhost:4567
```

Servicespezifische Endpunkte: Liste servicespezifischer Kennungen

Der AWS-Service Bezeichner basiert auf dem API Modell, indem alle Leerzeichen `serviceId` durch Unterstriche ersetzt und alle Buchstaben klein geschrieben werden.

Das folgende Beispiel für eine Service-ID verwendet. AWS Elastic Beanstalk AWS Elastic Beanstalk hat einen Wert `serviceId` von [Elastic Beanstalk](#), daher lautet der Service-Identifizierungsschlüssel `elastic_beanstalk`.

In der folgenden Tabelle sind alle servicespezifischen Kennungen, `config`-Dateischlüssel und Umgebungsvariablen aufgeführt.

Priorität der Endpunktkonfiguration und der Einstellungen

Die Einstellungen für die Endpunktkonfiguration befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien oder werden explizit in der Befehlszeile als Parameter deklariert. Die AWS CLI - Endpunktkonfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. Die Befehlszeilenoption [--endpoint-url](#)
2. Bei aktivierter Option die globale Endpunkt-Umgebungsvariable [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) oder die Profileinstellung [ignore_configure_endpoint_urls](#) zum Ignorieren von benutzerdefinierten Endpunkten
3. Der Wert, der von einer servicespezifischen Umgebungsvariablen [AWS_ENDPOINT_URL_<SERVICE>](#) bereitgestellt wird, z. B. [AWS_ENDPOINT_URL_DYNAMODB](#)
4. Die von den [AWS_USE_DUALSTACK_ENDPOINT-](#), [AWS_USE_FIPS_ENDPOINT-](#) und [AWS_ENDPOINT_URL-](#)Umgebungsvariablen bereitgestellten Werte.
5. Der servicespezifische Endpunktwert, der durch die Einstellung [endpoint_url](#) in einem services-Abschnitt der freigegebenen config-Datei bereitgestellt wird
6. Der Wert, der durch die Einstellung [endpoint_url](#) in einem profile der freigegebenen config-Datei bereitgestellt wird
7. [use_dualstack_endpoint-](#), [use_fips_endpoint-](#) und [endpoint_url](#)-Einstellungen.
8. Jeder Standard-Endpunkt URL für den jeweiligen Endpunkt AWS-Service wird zuletzt verwendet. Eine Liste der Standard-Service-Endpunkte, die in den einzelnen Regionen verfügbar sind, finden Sie unter [AWS -Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

Authentifizierungs- und Zugangsdaten für AWS CLI

Sie müssen bei der Entwicklung mit AWS Diensten festlegen AWS, wie sich das AWS CLI authentifiziert. Wählen Sie eine der folgenden Optionen AWS CLI, um Anmeldeinformationen für den programmatischen Zugriff auf zu konfigurieren. Die Optionen sind in der Reihenfolge aufgeführt, in der sie empfohlen werden.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Zweck	Anweisungen
IAM	Verwenden Sie kurzfristige Anmeldeinformationen.	the section called “Kurzfristige Anmeldeinformationen”
IAM	Verwenden Sie Rollen als Anmeldeinformationen.	the section called “IAMRollen”
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen.	the section called “IAMBenutzer”

Vorrang der Konfiguration und der Anmeldeinformationen

Anmeldeinformationen und Konfigurationseinstellungen befinden sich an mehreren Stellen, z. B. in den System- oder Benutzerumgebungsvariablen, in lokalen AWS Konfigurationsdateien, oder werden explizit in der Befehlszeile als Parameter deklariert. Bestimmte Speicherorte haben Vorrang vor anderen. Die AWS CLI -Anmeldeinformationen und Konfigurationseinstellungen haben Vorrang in der folgenden Reihenfolge:

1. [Befehlszeilenoptionen](#) – überschreiben Einstellungen an jedem anderen Speicherort, z. B. die Parameter `--region`, `--output` und `--profile`.
2. [Umgebungsvariablen](#) – Sie können Werte in den Umgebungsvariablen Ihres Systems speichern.
3. [Rolle übernehmen](#) — Übernehmen Sie die Berechtigungen einer IAM Rolle über die Konfiguration oder den `aws sts assume-role` Befehl.
4. [Rolle mit Webidentität übernehmen](#) — Übernehmen Sie die Rechte einer IAM Rolle, die die Webidentität über die Konfiguration oder den `aws sts assume-role` Befehl verwendet.

5. [Anmeldeinformationsdatei](#) – die Dateien `credentials` und `config` werden aktualisiert, wenn Sie den Befehl `aws configure` ausführen. Die Datei `credentials` befindet sich in `~/.aws/credentials` unter Linux und in macOS oder in `C:\Users\USERNAME\.aws\credentials` unter Windows.
6. [Benutzerdefinierter Prozess](#) – rufen Sie Ihre Anmeldeinformationen von einer externen Quelle ab.
7. [Konfigurationsdatei](#) – die Dateien `credentials` und `config` werden aktualisiert, wenn Sie den Befehl `aws configure` ausführen. Die Datei `config` befindet sich in `~/.aws/config` unter Linux und in macOS oder in `C:\Users\USERNAME\.aws\config` unter Windows.
8. [Container-Anmeldeinformationen](#) — Sie können jeder Ihrer Amazon Elastic Container Service (AmazonECS) -Aufgabendefinitionen eine IAM Rolle zuordnen. Temporäre Anmeldeinformationen für diese Rolle stehen dann für die Container dieser Aufgabe zur Verfügung. Weitere Informationen finden Sie unter [IAMRollen für Aufgaben](#) im Amazon Elastic Container Service Developer Guide.
9. [Anmeldeinformationen für das EC2 Amazon-Instance-Profil](#) — Sie können jeder Ihrer Amazon Elastic Compute Cloud (AmazonEC2) -Instances eine IAM Rolle zuordnen. Temporäre Anmeldeinformationen für diese Rolle stehen dann für den Code zur Verfügung, der in dieser Instance ausgeführt wird. Die Anmeldeinformationen werden über den EC2 Amazon-Metadatenservice bereitgestellt. Weitere Informationen finden Sie unter [IAMRollen für Amazon EC2](#) im EC2Amazon-Benutzerhandbuch und [Verwenden von Instanzprofilen](#) im IAMBenutzerhandbuch.

Weitere Themen in diesem Abschnitt

- [the section called “Kurzfristige Anmeldeinformationen”](#)
- [the section called “IAMRollen”](#)
- [the section called “IAMBenutzer”](#)
- [the section called “Verwendung von EC2 Amazon-Instance-Metadaten als Anmeldeinformationen in der AWS CLI”](#)
- [the section called “Externe Anmeldeinformationen”](#)

Authentifizierung mit kurzfristigen Zugangsdaten für den AWS CLI

Wir empfehlen, Ihr SDK Tool so zu konfigurieren, dass die [IAMIdentity Center-Authentifizierung](#) mit Optionen für die erweiterte Sitzungsdauer verwendet wird. Sie können jedoch temporäre Anmeldeinformationen, die im AWS Zugangsportaal verfügbar sind, kopieren und verwenden. Wenn

Wenn der einen Service-Client SDK erstellt, greift er auf diese temporären Anmeldeinformationen zu und verwendet sie für jede Anfrage. Die Einstellungen für die in Schritt 2a gewählte IAM Rolle bestimmen, [wie lange die temporären Anmeldeinformationen gültig sind](#). Die maximale Dauer beträgt zwölf Stunden.

Wiederholen Sie diese Schritte jedes Mal, wenn Ihre Anmeldeinformationen ablaufen.

Verwenden einer IAM Rolle in der AWS CLI

Eine [Rolle AWS Identity and Access Management \(IAM\)](#) ist ein Autorisierungstool, mit dem ein Benutzer zusätzliche (oder andere) Berechtigungen oder Berechtigungen zum Ausführen von Aktionen in einem anderen AWS Konto erhält.

Themen

- [Voraussetzungen](#)
- [Überblick über die Verwendung von IAM Rollen](#)
- [Konfigurieren und Verwenden einer Rolle](#)
- [Verwenden von Multi-Factor Authentication \(MFA\)](#)
- [Kontenübergreifende Rollen und externe ID](#)
- [Angaben eines Rollensitzungsnamens für eine einfachere Prüfung](#)
- [Übernehmen einer Rolle mit Web-Identität](#)
- [Anmeldeinformationen aus dem Cache löschen](#)

Voraussetzungen

Wenn Sie diese `iam`-Befehle verwenden möchten, müssen Sie die AWS CLI installieren und konfigurieren. Weitere Informationen finden Sie unter [Installieren Sie das AWS CLI](#).

Überblick über die Verwendung von IAM Rollen

Sie können AWS Command Line Interface (AWS CLI) für die Verwendung einer IAM Rolle konfigurieren, indem Sie ein Profil für die Rolle in der `~/ .aws/config` Datei definieren.

Im folgenden Beispiel sehen Sie ein Rollenprofil namens `marketingadmin`.

Wenn Sie Befehle mit `ausführen --profile marketingadmin` (oder sie mit der [AWS_PROFILE Umgebungsvariablen](#) angeben), AWS CLI verwendet der die in einem separaten Profil definierten Anmeldeinformationen, `user1` um die Rolle mit dem Amazon-Ressourcennamen

(ARN) anzunehmen: `arn:aws:iam::123456789012:role/marketingadminrole`. Sie können alle Operationen ausführen, die gemäß den der Rolle zugewiesenen Berechtigungen zulässig sind.

```
[profile marketingadmin]  
role_arn = arn:aws:iam::123456789012:role/marketingadminrole  
source_profile = user1
```

Sie können dann ein `source_profile` angeben, das auf ein separates benanntes Profil zeigt, welches Benutzer-Anmeldeinformationen mit der Berechtigung zur Verwendung der Rolle enthält. Im vorherigen Beispiel verwendet das Profil `marketingadmin` die Anmeldeinformationen im Profil `user1`. Wenn Sie angeben, dass ein AWS CLI Befehl das Profil verwenden soll `marketingadmin`, sucht das AWS CLI automatisch nach den Anmeldeinformationen für das verknüpfte `user1` Profil und verwendet sie, um temporäre Anmeldeinformationen für die angegebene IAM Rolle anzufordern. Der CLI verwendet dazu die AssumeRole Operation [sts](#): im Hintergrund. Diese temporären Anmeldeinformationen werden dann verwendet, um den angeforderten AWS CLI -Befehl auszuführen. An die angegebene Rolle müssen IAM Berechtigungsrichtlinien angehängt sein, die die Ausführung des angeforderten AWS CLI Befehls ermöglichen.

Um einen AWS CLI Befehl innerhalb einer Amazon Elastic Compute Cloud (AmazonEC2) -Instance oder eines Amazon Elastic Container Service (AmazonECS) -Containers auszuführen, können Sie eine IAM Rolle verwenden, die dem Instance-Profil oder dem Container zugewiesen ist. Wenn Sie kein Profil angeben oder keine Umgebungsvariablen festlegen, wird diese Rolle direkt verwendet. So müssen Sie keine Langzeit-Zugriffsschlüssel in Ihren Instances speichern. Sie können diese Instance- oder Container-Rollen auch verwenden, um Anmeldeinformationen für eine andere Rolle abzurufen. Zu diesem Zweck verwenden Sie `credential_source` (anstelle von `source_profile`), um anzugeben, wie die Anmeldeinformationen zu finden sind. Das Attribut `credential_source` unterstützt die folgenden Werte:

- `Environment` – um die Quell-Anmeldeinformationen aus Umgebungsvariablen abzurufen.
- `Ec2InstanceMetadata`— Verwendet die IAM Rolle, die dem EC2 Amazon-Instance-Profil zugewiesen ist.
- `EcsContainer`— Verwendet die IAM Rolle, die dem ECS Amazon-Container zugewiesen ist.

Das folgende Beispiel zeigt dieselbe `marketingadminrole` Rolle, die verwendet wird, indem auf ein EC2 Amazon-Instance-Profil verwiesen wird.

```
[profile marketingadmin]
```

```
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
```

Wenn Sie eine Rolle aufrufen, stehen Ihnen zusätzliche Optionen zur Verfügung, die Sie anfordern können, z. B. die Verwendung einer Multifaktor-Authentifizierung und einer externen ID (die von Drittanbietern für den Zugriff auf die Ressourcen ihrer Kunden verwendet wird). Sie können auch eindeutige Namen für Rollensitzungen angeben, die einfacher in AWS CloudTrail Protokollen überprüft werden können.

Konfigurieren und Verwenden einer Rolle

Wenn Sie Befehle mit einem Profil ausführen, das eine IAM Rolle spezifiziert, AWS CLI verwendet die Anmeldeinformationen des Quellprofils, um AWS Security Token Service (AWS STS) aufzurufen und temporäre Anmeldeinformationen für die angegebene Rolle anzufordern. Der Benutzer im Quellprofil muss über die Berechtigung zum Aufrufen von `sts:assume-role` für die Rolle im angegebenen Profil verfügen. Die Rolle muss über eine Vertrauensstellung verfügen, die es dem Benutzer im Quellprofil ermöglicht, die Rolle zu übernehmen. Der Prozess des Abrufens mit anschließendem Verwenden von temporären Anmeldeinformationen für eine Rolle wird oft als Annehmen der Rolle bezeichnet.

Sie können eine Rolle IAM mit den Berechtigungen erstellen, die Benutzer annehmen sollen, indem Sie das Verfahren unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM Benutzer](#) im AWS Identity and Access Management Benutzerhandbuch befolgen. Wenn die Rolle und der -Benutzer des Quellprofils im selben Konto vorhanden sind, können Sie beim Konfigurieren der Vertrauensstellung der Rolle Ihre eigene Konto-ID eingeben.

Nachdem Sie die Rolle erstellt haben, ändern Sie die Vertrauensstellung, um die Übernahme durch den -Benutzer zuzulassen.

Im folgenden Beispiel sehen Sie eine Vertrauensrichtlinie, die Sie einer Rolle anfügen könnten. Mit dieser Richtlinie kann die Rolle von jedem Benutzer im Konto 123456789012 angenommen werden, falls der Administrator des Kontos dem Benutzer explizit die Berechtigung `sts:AssumeRole` erteilt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      }
    }
  ]
}
```



```
    },
    "Action": "sts:AssumeRole"
  }
]
}
```

Die Vertrauensrichtlinie selbst gewährt keine Berechtigungen. Der Administrator des Kontos muss die Berechtigung zur Übernahme der Rolle an einzelne Benutzer delegieren, indem eine Richtlinie mit den entsprechenden Berechtigungen angefügt wird. Das folgende Beispiel zeigt eine Richtlinie, die Sie einem Benutzer anfügen können und mit der der Benutzer nur die Rolle `marketingadminrole` annehmen kann. Weitere Informationen darüber, wie Sie einem Benutzer Zugriff auf die Übernahme einer Rolle [gewähren können, finden Sie im Benutzerhandbuch unter Erteilen von Benutzerberechtigungen zum Rollenwechsel](#). IAM

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

Der Benutzer benötigt keine zusätzlichen Berechtigungen, um die AWS CLI Befehle mithilfe des Rollenprofils auszuführen. Die Berechtigungen zum Ausführen des Befehls entstammen aus den der Rolle angefügten Berechtigungen. Sie fügen der Rolle Berechtigungsrichtlinien hinzu, um anzugeben, welche Aktionen für welche AWS Ressourcen ausgeführt werden können. Weitere Informationen zum Zuweisen von Berechtigungen an eine Rolle (was genauso funktioniert wie bei einem Benutzer) finden Sie unter [Ändern von Berechtigungen für einen IAM Benutzer](#) im IAMBenutzerhandbuch.

Nachdem Sie nun das Rollenprofil, die Rollenberechtigungen, die Rollen-Vertrauensstellung und die Benutzerberechtigungen konfiguriert haben, können Sie die Rolle an der Befehlszeile verwenden, indem Sie die Option `--profile` aufrufen. Beispiel: Der folgende Befehl ruft den Amazon-S3-Befehl `ls` mit den Berechtigungen auf, die an die `marketingadmin`-Rolle angefügt sind, wie im Beispiel zu Beginn dieses Themas angegeben.

```
$ aws s3 ls --profile marketingadmin
```

Um die Rolle für mehrere Aufrufe zu verwenden, können Sie die Umgebungsvariable `AWS_PROFILE` für die aktuelle Sitzung über die Befehlszeile festlegen. Da die Umgebungsvariable definiert ist, müssen Sie die Option `--profile` nicht bei jedem Befehl angeben.

Linux oder macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

Weitere Informationen zur Konfiguration von Benutzern und Rollen finden Sie im Benutzerhandbuch unter [IAMIdentitäten \(Benutzer, Benutzergruppen und Rollen\)](#) und [IAMRollen](#). IAM

Verwenden von Multi-Factor Authentication (MFA)

Für zusätzliche Sicherheit können Sie verlangen, dass Benutzer einen einmaligen Schlüssel angeben, der von einem Gerät mit Multifaktor-Authentifizierung (MFA), einem U2F-Gerät oder einer mobilen App generiert wurde, wenn sie versuchen, über das Rollenprofil einen Anruf zu tätigen.

Zunächst können Sie die Vertrauensstellung für die IAM Rolle so ändern, dass sie erforderlich ist. MFA Auf diese Weise wird verhindert, dass jemand die Rolle verwenden kann, ohne sich zuvor mithilfe MFA von zu authentifizieren. Ein Beispiel finden Sie im folgenden Code in der Zeile `Condition`. Diese Richtlinie ermöglicht es dem genannten Benutzer, die Rolle `anika` anzunehmen, mit der die Richtlinie verknüpft ist, aber nur, wenn er sich mithilfe von `authenticate.MFA`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/anika" },
      "Action": "sts:AssumeRole",
      "Condition": { "Bool": { "aws:multifactorAuthPresent": true } }
    }
  ]
}
```

Als Nächstes fügen Sie dem Rollenprofil eine Zeile hinzu, die das MFA Gerät ARN des Benutzers angibt. In den folgenden Einträgen der Beispieldatei `config` verwenden zwei Rollenprofile die Zugriffsschlüssel für den Benutzer `anika`, um temporäre Anmeldeinformationen für die Rolle `cli-role` anzufordern. Der Benutzer `anika` hat die Berechtigung, die Rolle zu übernehmen. Sie wird ihm von der Vertrauensrichtlinie der Rolle erteilt.

```
[profile role-without-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile=cli-user

[profile role-with-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile = cli-user
mfa_serial = arn:aws:iam::128716708097:mfa/cli-user

[profile cli-user]
region = us-west-2
output = json
```

Die `mfa_serial` Einstellung kann ARN, wie abgebildet, eine oder die Seriennummer eines MFA Hardwaretokens annehmen.

Das erste Profil `role-without-mfa`,, erfordert nicht MFA. Da das vorherige Beispiel für die der Rolle zugeordnete Vertrauensrichtlinie dies erfordert MFA, schlägt jedoch jeder Versuch fehl, einen Befehl mit diesem Profil auszuführen.

```
$ aws iam list-users --profile role-without-mfa
```

```
An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

Der zweite Profileintrag, `role-with-mfa`, identifiziert ein zu MFA verwendendes Gerät. Wenn der Benutzer versucht, einen AWS CLI Befehl mit diesem Profil auszuführen, AWS CLI fordert er den Benutzer auf, das vom MFA Gerät bereitgestellte Einmalkennwort (OTP) einzugeben. Wenn die MFA Authentifizierung erfolgreich ist, führt der Befehl den angeforderten Vorgang aus. Das OTP wird nicht auf dem Bildschirm angezeigt.

```
$ aws iam list-users --profile role-with-mfa
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
```

```
{
  "Users": [
    {
      ...
    }
  ]
}
```

Kontenübergreifende Rollen und externe ID

Sie können -Benutzern ermöglichen, Rollen zu verwenden, die zu verschiedenen Konten gehören, indem Sie kontenübergreifende Rollen konfigurieren. Stellen Sie bei der Rollenerstellung den Rollentyp auf [Anderes AWS Konto](#) ein, wie unter [Eine Rolle zum Delegieren von Berechtigungen an einen IAM Benutzer erstellen](#) beschrieben. Wählen Sie optional Erforderlich aus. MFA MFAResolve konfiguriert die entsprechende Bedingung in der Vertrauensbeziehung, wie unter [Verwenden von Multi-Factor Authentication \(MFA\)](#) beschrieben.

Wenn Sie eine [externe ID](#) verwenden, um besser zu kontrollieren, wer eine kontenübergreifende Rolle verwenden kann, müssen Sie auch den Parameter `external_id` zum Rollenprofil hinzufügen. Normalerweise verwenden Sie dieses nur, wenn das andere Konto von einer Person außerhalb Ihres Unternehmens kontrolliert wird.

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

Angeben eines Rollensitzungsnamens für eine einfachere Prüfung

Wenn viele Personen eine Rolle gemeinsam nutzen, wird die Prüfung zu einer größeren Herausforderung. Sie möchten jede aufgerufene Operation mit der Person verknüpfen, die diese Aktion aufgerufen hat. Wenn die Person jedoch eine Rolle verwendet, ist die Annahme der Rolle durch die Person eine vom Aufrufen einer Operation getrennte Aktion und Sie müssen die beiden manuell korrelieren.

Sie können dies vereinfachen, indem Sie eindeutige Rollensitzungsnamen angeben, wenn Benutzer eine Rolle annehmen. Dazu fügen Sie jedem benannten Profil in der Datei `config`, die eine Rolle angibt, einen `role_session_name`-Parameter hinzu. Der `role_session_name` Wert wird an den `AssumeRole` Vorgang übergeben und wird Teil der ARN For-the-Rollensitzung. Er ist auch in den AWS CloudTrail Protokollen für alle protokollierten Vorgänge enthalten.

Sie können beispielsweise folgendermaßen ein rollenbasiertes Profil erstellen.

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
role_session_name = Session_Maria_Garcia
```

Dies führt dazu, dass die Rollensitzung wie folgt aussiehtARN.

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

Außerdem enthalten alle AWS CloudTrail Protokolle den Namen der Rollensitzung in den für jeden Vorgang erfassten Informationen.

Übernehmen einer Rolle mit Web-Identität

Sie können ein Profil so konfigurieren, dass es angibt, dass sie mithilfe von [Web Identity Federation und Open ID Connect \(OIDC\)](#) eine Rolle übernehmen AWS CLI sollen. Wenn Sie dies in einem Profil angeben, führt der AWS CLI automatisch den entsprechenden AWS STS AssumeRoleWithWebIdentity Anruf für Sie durch.

Note

Wenn Sie ein Profil angeben, das eine IAM Rolle verwendet, werden die AWS CLI entsprechenden Aufrufe ausgeführt, um temporäre Anmeldeinformationen abzurufen. Diese Anmeldeinformationen werden in `~/.aws/cli/cache` gespeichert. Nachfolgende AWS CLI Befehle, die dasselbe Profil angeben, verwenden die zwischengespeicherten temporären Anmeldeinformationen, bis sie ablaufen. Zu diesem Zeitpunkt werden die AWS CLI Anmeldeinformationen automatisch aktualisiert.

Um mithilfe des Web-Identitätsverbands temporäre Anmeldeinformationen abzurufen und zu verwenden, können Sie die folgenden Konfigurationswerte in einem freigegebenen Profil angeben.

role_arn

Gibt die Rolle ARN an, die übernommen werden soll.

web_identity_token_file

Gibt den Pfad zu einer Datei an, die ein OAuth 2.0-Zugriffstoken oder ein OpenID Connect-ID-Token enthält, das vom Identitätsanbieter bereitgestellt wird. Die AWS CLI lädt diese

Datei und übergibt den Inhalt als `WebIdentityToken`-Argument an die Operation `AssumeRoleWithWebIdentity`.

role_session_name

Gibt einen optionalen Namen an, der auf diese Rollenübernahme-Sitzung angewendet wird.

Unten finden Sie eine Beispielkonfiguration für den mindestens erforderlichen Umfang einer Konfiguration für eine angenommene Rolle mit Web-Identitätsprofil.

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoleNameToAssume
web_identity_token_file=/path/to/a/token
```

Sie können diese Konfiguration auch mithilfe von [Umgebungsvariablen](#) bereitstellen.

AWS_ROLE_ARN

Die ARN Rolle, die übernommen werden soll.

AWS_WEB_IDENTITY_TOKEN_FILE

Der Pfad zur Datei mit dem Web-Identitäts-Token.

AWS_ROLE_SESSION_NAME

Der Name für diese Sitzung der Rollenübernahme.

Note

Diese Umgebungsvariablen gelten derzeit nur für die Anbieter für Rollenübernahme mit Web-Identität. Sie gelten nicht für die allgemeine Konfiguration für einen Anbieter für Rollenübernahme.

Anmeldeinformationen aus dem Cache löschen

Wenn Sie eine Rolle verwenden, werden die temporären Anmeldeinformationen lokal AWS CLI zwischengespeichert, bis sie ablaufen. Wenn Sie das nächste Mal versuchen, sie zu verwenden, wird AWS CLI versucht, sie in Ihrem Namen zu erneuern.

Wenn die temporären Anmeldeinformationen Ihrer Rolle [widerrufen](#) werden, werden sie nicht automatisch erneuert und Versuche, sie zu verwenden, schlagen fehl. Sie können den Cache jedoch löschen, um das Abrufen neuer Anmeldeinformationen AWS CLI zu erzwingen.

Linux oder macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\aws\cli\cache
```

Authentifizierung mit IAM Benutzeranmeldedaten für den AWS CLI

Warning

Verwenden Sie zur Vermeidung von Sicherheitsrisiken keine IAM Benutzer zur Authentifizierung, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie [AWS IAM Identity Center](#).

In diesem Abschnitt wird erklärt, wie Sie die Grundeinstellungen mit einem IAM Benutzer konfigurieren. Dazu gehören Ihre Sicherheitsanmeldeinformationen unter Verwendung der Dateien `config` und `credentials`.

Themen

- [Schritt 1: Erstellen Sie Ihren IAM Benutzer](#)
- [Schritt 2: Abrufen Ihrer Zugriffsschlüssel](#)
- [Konfigurieren Sie AWS CLI](#)
 - [Verwenden von aws configure](#)

Schritt 1: Erstellen Sie Ihren IAM Benutzer

Erstellen Sie Ihren IAM Benutzer, indem Sie den Anweisungen [zum Erstellen von IAM Benutzern \(Konsole\)](#) im IAMBenutzerhandbuch folgen.

- Wählen Sie unter Berechtigung-Optionen mit der Option Direktes Anfügen von Richtlinien aus, wie Sie diesem Benutzer Berechtigungen zuweisen möchten.
- Die meisten „Erste Schritte“ SDK -Tutorials verwenden den Amazon S3 S3-Service als Beispiel. Wenn Sie Ihrer Anwendung Vollzugriff auf Amazon S3 gewähren möchten, wählen Sie die AmazonS3FullAccess-Richtlinie zum Anfügen an diesen Benutzer aus.

Schritt 2: Abrufen Ihrer Zugriffsschlüssel

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Konsole Benutzer und dann den **User name** Benutzer aus, den Sie zuvor erstellt haben.
3. Wählen Sie auf der Seite des Benutzers die Seite Sicherheitsanmeldeinformationen aus. Wählen Sie dann unter Zugriffsschlüssel die Option Zugriffsschlüssel erstellen aus.
4. Wählen Sie für Schritt 1 „Zugriffsschlüssel erstellen“ die Option Befehlszeilenschnittstelle (CLI) aus.
5. Geben Sie für Zugriffsschlüssel erstellen – Schritt 2 ein optionales Tag ein und wählen Sie Weiter aus.
6. Wählen Sie für Schritt 3 „Zugriffsschlüssel erstellen“ die Option „.csv-Datei herunterladen“ aus, um eine .csv Datei mit dem Zugriffsschlüssel und dem geheimen Zugriffsschlüssel Ihres IAM Benutzers zu speichern. Sie benötigen diese Informationen später wieder.
7. Wählen Sie Done (Fertig).

Konfigurieren Sie AWS CLI

Für den allgemeinen Gebrauch AWS CLI benötigt er die folgenden Informationen:

- Zugriffsschlüssel-ID
- Geheimer Zugriffsschlüssel
- AWS Region
- Ausgabeformat

Die AWS CLI speichert diese Informationen in einem Profil (einer Sammlung von Einstellungen), das default in der `credentials` Datei benannt ist. Standardmäßig werden die Informationen

in diesem Profil verwendet, wenn Sie einen AWS CLI Befehl ausführen, der nicht explizit ein zu verwendendes Profil angibt. Weitere Informationen zur `credentials`-Datei finden Sie unter [Einstellungen für die Konfiguration und die Anmeldeinformationsdatei in der AWS CLI](#).

Verwenden Sie eines der folgenden Verfahren AWS CLI, um das zu konfigurieren:

Themen

- [Verwenden von `aws configure`](#)

Verwenden von `aws configure`

Für den allgemeinen Gebrauch ist der `aws configure` Befehl der schnellste Weg, Ihre AWS CLI Installation einzurichten. Dieser Konfigurationsassistent fordert Sie auf, alle Informationen einzugeben, die Sie für die ersten Schritte benötigen. Sofern mit der `--profile` Option nicht anders angegeben, AWS CLI speichert diese Informationen im `default` Profil.

Im folgenden Beispiel wird ein `default`-Profil anhand von Beispielwerten konfiguriert. Ersetzen Sie sie durch eigene Werte, wie in den folgenden Abschnitten beschrieben.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Im folgenden Beispiel wird ein Profil mit dem Namen `userprod` anhand von Beispielwerten konfiguriert. Ersetzen Sie sie durch eigene Werte, wie in den folgenden Abschnitten beschrieben.

```
$ aws configure --profile userprod
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Verwendung von EC2 Amazon-Instance-Metadaten als Anmeldeinformationen in der AWS CLI

Wenn Sie das AWS CLI von einer Amazon Elastic Compute Cloud (AmazonEC2) -Instance aus ausführen, können Sie die Bereitstellung von Anmeldeinformationen für Ihre Befehle vereinfachen.

Jede EC2 Amazon-Instance enthält Metadaten, die direkt nach temporären Anmeldeinformationen abgefragt werden AWS CLI können. Wenn der Instance eine IAM Rolle zugewiesen wird, ruft sie die Anmeldeinformationen AWS CLI automatisch und sicher aus den Instance-Metadaten ab.

Verwenden Sie die DISABLED Umgebungsvariable [AWS_EC2_METADATA_](#), um diesen Dienst zu deaktivieren.

Themen

- [Voraussetzungen](#)
- [Konfiguration eines Profils für EC2 Amazon-Metadaten](#)

Voraussetzungen

Um EC2 Amazon-Anmeldeinformationen mit dem zu verwenden AWS CLI, müssen Sie die folgenden Schritte ausführen:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Sie verstehen Konfigurationsdateien und benannte Profile. Weitere Informationen finden Sie unter [Einstellungen für die Konfiguration und die Anmeldeinformationsdatei in der AWS CLI](#).
- Sie haben eine Rolle AWS Identity and Access Management (IAM) erstellt, die Zugriff auf die benötigten Ressourcen hat, und diese Rolle der EC2 Amazon-Instance zugewiesen, wenn Sie sie starten. Weitere Informationen finden Sie unter [IAMRichtlinien für Amazon EC2 im EC2 Amazon-Benutzerhandbuch](#) und [Gewähren von Zugriff auf AWS Ressourcen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden](#), im IAMBenutzerhandbuch.

Konfiguration eines Profils für EC2 Amazon-Metadaten

Um anzugeben, dass Sie die im EC2 Hosting-Amtazon-Instance-Profil verfügbaren Anmeldeinformationen verwenden möchten, verwenden Sie die folgende Syntax im benannten Profil in Ihrer Konfigurationsdatei. Weitere Anweisungen finden Sie in den folgenden Schritten.

```
[profile profilename]  
role_arn = arn:aws:iam::123456789012:role/rolename  
credential_source = Ec2InstanceMetadata  
region = region
```

1. Erstellen Sie ein Profil in Ihrer Konfigurationsdatei.

```
[profile profilename]
```

2. Fügen Sie Ihre IAM ARN-Rolle hinzu, die Zugriff auf die benötigten Ressourcen hat.

```
role_arn = arn:aws:iam::123456789012:role/rolename
```

3. Geben Sie Ec2InstanceMetadata als Quelle für die Anmeldeinformationen an.

```
credential_source = Ec2InstanceMetadata
```

4. Legen Sie Ihre Region fest.

```
region = region
```

Beispiel

Das folgende Beispiel setzt voraus, dass *Rolle eines Marketingadministrators* Rolle und verwendet die *us-west-2* Region in einem EC2 Amazon-Instance-Profil mit dem Namen *marketingadmin*.

```
[profile marketingadmin]  
role_arn = arn:aws:iam::123456789012:role/marketingadminrole  
credential_source = Ec2InstanceMetadata  
region = us-west-2
```

Beschaffung von Referenzen mit einem externen Prozess in der AWS CLI

Warning

In diesem Thema wird die Beschaffung von Anmeldeinformationen von einem externen Prozess erläutert. Dies könnte ein Sicherheitsrisiko darstellen, wenn der Befehl zum Generieren der Anmeldeinformationen nicht zulässigen Prozessen oder Benutzern zugänglich wird. Wir empfehlen Ihnen, die unterstützten, sicheren Alternativen zu verwenden, die von AWS CLI und bereitgestellt werden, um das Risiko AWS zu verringern, dass Ihre

Anmeldeinformationen kompromittiert werden. Sorgen Sie dafür, dass Sie die `config`-Datei und alle zugehörigen Dateien und Tools vor einer Offenlegung sichern.

Stellen Sie sicher, dass Ihr benutzerdefiniertes Anmeldeinformationstool keine geheimen Informationen speichert, `StdErr` da es solche Informationen erfassen und protokollieren AWS CLI kann, wodurch sie möglicherweise unbefugten Benutzern zugänglich gemacht werden. SDKs

Wenn Sie über eine Methode zum Generieren oder Nachschlagen von Anmeldeinformationen verfügen, die nicht direkt von der unterstützt wird AWS CLI, können Sie die AWS CLI so konfigurieren, dass sie verwendet wird, indem Sie die `credential_process` Einstellung in der `config` Datei konfigurieren.

Beispielsweise könnten Sie einen etwa wie folgt aussehenden Eintrag in die `config`-Datei aufnehmen.

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

Syntax

Um diese Zeichenfolge auf eine Weise zu erstellen, die mit jedem Betriebssystem kompatibel ist, gehen Sie nach den folgenden Regeln vor:

- Wenn der Pfad oder Dateiname ein Leerzeichen enthält, umgeben Sie den vollständigen Pfad und Dateinamen mit doppelten Anführungszeichen („“). Pfad und Dateiname dürfen nur aus folgenden Zeichen bestehen: A-Z a-z 0-9 – _ . Leerzeichen
- Wenn ein Parametername oder ein Parameterwert ein Leerzeichen enthält, umgeben Sie dieses Element mit doppelten Anführungszeichen („“). Umgeben Sie dabei nur den Namen oder den Wert, nicht beides.
- Fügen Sie keine Umgebungsvariablen in die Zeichenfolgen ein. Sie können beispielsweise nicht `$HOME` oder `%USERPROFILE%` einschließen.
- Geben Sie den Basisordner nicht als `~` an. Sie müssen den vollständigen Pfad angeben.

Beispiel für Windows

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter with spaces"
```

Beispiel für Linux oder macOS

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces "parameter with spaces"
```

Erwartete Ausgabe des Anmeldeinformationsprogramms


Der AWS CLI führt den Befehl wie im Profil angegeben aus und liest dann Daten von `STDOUT`. Der von Ihnen angegebene Befehl muss eine JSON Ausgabe generieren `STDOUT`, die der folgenden Syntax entspricht.

```
{
  "Version": 1,
  "AccessKeyId": "an AWS access key",
  "SecretAccessKey": "your AWS secret access key",
  "SessionToken": "the AWS session token for temporary credentials",
  "Expiration": "ISO8601 timestamp when the credentials expire"
}
```

Note

Derzeit muss der `Version`-Schlüssel auf 1 gesetzt sein. Im Laufe der Zeit kann ein höherer Wert erforderlich sein, wenn sich die Struktur weiterentwickelt.

Der `Expiration` Schlüssel ist ein Zeitstempel im Format [ISO8601](#). Wenn der `Expiration` Schlüssel nicht in der Ausgabe des Tools enthalten ist, wird CLI davon ausgegangen, dass es sich bei den Anmeldeinformationen um langfristige Anmeldeinformationen handelt, die nicht aktualisiert werden. Andernfalls werden die Anmeldeinformationen als temporäre Anmeldeinformationen angesehen und automatisch aktualisiert, indem der Befehl `credential_process` vor ihrem Ablauf erneut ausgeführt wird.

 Note

Der speichert AWS CLI keine externen Prozessanmeldedaten im Cache, so wie er es bei der Übernahme von Rollenmeldedaten tut. Wenn Caching erforderlich ist, müssen Sie dies im externen Prozess implementieren.

Der externe Prozess kann einen Rückgabecode ungleich Null zurückgeben, um anzuzeigen, dass beim Abrufen der Anmeldeinformationen ein Fehler aufgetreten ist.

Mit dem AWS CLI

Dieser Abschnitt bietet einen umfassenden Überblick über die allgemeine Verwendung, die allgemeinen Funktionen und Optionen, die in AWS Command Line Interface (AWS CLI) verfügbar sind, und geht dabei über die im [the section called “Endpunkte”](#) Abschnitt Konfiguration behandelten Details hinaus.

Dieses Handbuch befasst sich mit den grundlegenden Aspekten des Schreibens von AWS CLI Befehlen, einschließlich ihrer grundlegenden Struktur, Formatierung und Filterfunktionen. Wenn Sie diese Kernelemente verstehen, können Sie Befehle erstellen, die genau auf die Ressourcen und Aktionen abzielen, die Sie benötigen, ohne sich in komplexen webbasierten Konsolen zurechtfinden zu müssen.

Darüber hinaus werden die Hilfeinhalte und die Dokumentation hervorgehoben, die für die verfügbar sind AWS CLI. Von der integrierten Befehlszeilenhilfe bis hin zum umfassenden [AWS CLI Referenzhandbuch AWS CLI](#) haben Sie Zugriff auf Informationen, die Sie bei der Erkundung der Funktionen und Möglichkeiten von unterstützen AWS CLI.

AWS-Service Konkrete Beispiele und Anwendungsfälle finden Sie im [AWS CLI Referenzhandbuch AWS CLI](#). Diese enthalten befehlspezifische Informationen und zeigen Beispiele, wie Sie das AWS CLI für verschiedene Zwecke nutzen können AWS-Services.

Note

Standardmäßig AWS CLI sendet der Anfragen an, AWS-Services indem HTTPS er TCP Port 443 verwendet. Um eine erfolgreiche Verwendung von sicherzustellen AWS CLI, müssen Sie in der Lage sein, ausgehende Verbindungen über diesen Port herzustellen.

Themen in diesem Leitfaden

- [Zugriff auf Hilfe und Ressourcen für die AWS CLI](#)
- [Befehlsstruktur in der AWS CLI](#)
- [Angaben von Parameterwerten in der AWS CLI](#)
- [Steuerung der Befehlsausgabe in der AWS CLI](#)
- [Befehlszeilen-Rückgabecodes in der AWS CLI](#)

- [Aliase erstellen und verwenden in AWS CLI](#)

Zugriff auf Hilfe und Ressourcen für die AWS CLI

In diesem Thema wird beschrieben, wie Sie auf Hilfeinhalte für AWS Command Line Interface (AWS CLI) zugreifen.

Themen

- [Der integrierte AWS CLI -help-Befehl](#)
- [AWS CLI Referenzhandbuch](#)
- [APIDokumentation](#)
- [Behebung von Fehlern](#)
- [Weitere Hilfe](#)

Der integrierte AWS CLI -help-Befehl

Sie können mit jedem Befehl Hilfe erhalten, wenn Sie die AWS Command Line Interface (AWS CLI) verwenden. Geben Sie dazu hinter einem Befehlsnamen einfach `help` ein.

Mit dem folgenden Befehl wird beispielsweise Hilfe zu den allgemeinen AWS CLI Optionen und den verfügbaren Befehlen der obersten Ebene angezeigt.

```
$ aws help
```

Der folgende Befehl zeigt die verfügbaren Amazon Elastic Compute Cloud (AmazonEC2) - spezifischen Befehle an.

```
$ aws ec2 help
```

Das folgende Beispiel zeigt detaillierte Hilfe für den EC2 `DescribeInstances` Amazon-Vorgang. Die Hilfe enthält Informationen zu den Eingabeparametern, verfügbaren Filter und dem Inhalt der Ausgabe. Sie enthält auch Beispiele dazu, wie Sie Variationen des Befehls eingeben können.

```
$ aws ec2 describe-instances help
```

Die Hilfe ist für jeden Befehl in sechs Abschnitte unterteilt:

Name

Der Name des Befehls.

```
NAME
    describe-instances -
```

Beschreibung

Eine Beschreibung des API Vorgangs, den der Befehl aufruft.

```
DESCRIPTION
    Describes one or more of your instances.

    If you specify one or more instance IDs, Amazon EC2 returns information
    for those instances. If you do not specify instance IDs, Amazon EC2
    returns information for all relevant instances. If you specify an
    instance ID that is not valid, an error is returned. If you specify an
    instance that you do not own, it is not included in the returned
    results.

    ...
```

Syntax

Die grundlegende Syntax für die Nutzung des Befehls und dessen Optionen. Wenn eine Option in eckigen Klammern dargestellt wird, ist sie optional oder hat einen Standardwert oder es gibt eine alternative Option, die verwendet werden kann.

```
SYNOPSIS
    describe-instances
    [--dry-run | --no-dry-run]
    [--instance-ids <value>]
    [--filters <value>]
    [--cli-input-json <value>]
    [--starting-token <value>]
    [--page-size <value>]
    [--max-items <value>]
    [--generate-cli-skeleton]
```

`describe-instances` hat beispielsweise ein Standardverhalten, das alle Instanzen im aktuellen Konto und in der AWS Region beschreibt. Sie können optional eine Liste von `instance-ids` angeben, um eine oder mehrere Instanzen zu beschreiben. `dry-run` ist ein optionales

boolesches Flag, das keinen Wert akzeptiert. Zur Verwendung eines booleschen Flags geben Sie einen der dargestellten Werte an, in diesem Fall `--dry-run` oder `--no-dry-run`. Ebenso akzeptiert `--generate-cli-skeleton` keinen Wert. Enthält eine Option Bedingungen für ihre Verwendung, sind diese im Abschnitt `OPTIONS` oder in den aufgeführten Beispielen angegeben.

Optionen

Eine Beschreibung der einzelnen Optionen aus der Zusammenfassung.

OPTIONS

`--dry-run | --no-dry-run` (boolean)

Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

`--instance-ids` (list)

One or more instance IDs.

Default: Describes all your instances.

...

Beispiele

Beispiele für die Nutzung des Befehls und seiner Optionen. Wenn kein Beispiel für einen Befehl oder Anwendungsfall verfügbar ist, den Sie benötigen, fordern Sie eines über den Feedback-Link auf dieser Seite oder in der AWS CLI Befehlsreferenz auf der Hilfeseite für den Befehl an.

EXAMPLES

To describe an Amazon EC2 instance

Command:

```
aws ec2 describe-instances --instance-ids i-5203422c
```

To describe all instances with the instance type `m1.small`

Command:

```
aws ec2 describe-instances --filters "Name=instance-type,Values=m1.small"
```

To describe all instances with an Owner tag

Command:

```
aws ec2 describe-instances --filters "Name=tag-key,Values=Owner"
```

...

Output

Beschreibungen der einzelnen Felder und Datentypen, die in der Antwort von AWS zurückgegeben werden.

Für `describe-instances` ist die Ausgabe eine Liste von Reservierungsobjekten, die jeweils mehrere Felder und Objekte enthalten, die Informationen über die damit verknüpften Instances bereitstellen. Diese Informationen stammen aus der [API-Dokumentation für den von Amazon verwendeten Reservierungsdatentyp EC2](#).

OUTPUT

Reservations -> (list)

One or more reservations.

(structure)

Describes a reservation.

ReservationId -> (string)

The ID of the reservation.

OwnerId -> (string)

The ID of the AWS account that owns the reservation.

RequesterId -> (string)

The ID of the requester that launched the instances on your behalf (for example, AWS Management Console or Auto Scaling).

Groups -> (list)

One or more security groups.

(structure)

Describes a security group.

GroupName -> (string)

The name of the security group.

GroupId -> (string)

The ID of the security group.

```

Instances -> (list)
    One or more instances.

(structure)
    Describes an instance.

    InstanceId -> (string)
        The ID of the instance.

    ImageId -> (string)
        The ID of the AMI used to launch the instance.

    State -> (structure)
        The current state of the instance.

    Code -> (integer)
        The low byte represents the state. The high byte
        is an opaque internal value and should be ignored.

...

```

Wenn die Ausgabe in AWS CLI gerendert wirdJSON, wird sie zu einem Array von Reservierungsobjekten, ähnlich dem folgenden Beispiel.

```

{
  "Reservations": [
    {
      "OwnerId": "012345678901",
      "ReservationId": "r-4c58f8a0",
      "Groups": [],
      "RequesterId": "012345678901",
      "Instances": [
        {
          "Monitoring": {
            "State": "disabled"
          },
          "PublicDnsName": "ec2-52-74-16-12.us-
west-2.compute.amazonaws.com",
          "State": {
            "Code": 16,
            "Name": "running"
          }
        }
      ]
    }
  ]
}
...

```

Jedes Reservierungsobjekt enthält Felder mit einer Beschreibung der Reservierung und einem Array von Instance-Objekten, jedes mit eigenen Feldern (z. B. `PublicDnsName`) und Objekten (z. B. `State`), die es beschreiben.

Windows-Nutzer

Sie können die Ausgabe des Hilfebefehls weiterleiten (`|`) an den `more`-Befehl, um die Hilfedatei seitenweise anzusehen. Drücken Sie die Leertaste oder `PgDn`, um einen größeren Teil des Dokuments anzuzeigen und den Vorgang `q` zu beenden.

```
C:\> aws ec2 describe-instances help | more
```

AWS CLI Referenzhandbuch

Die Hilfedateien enthalten Links, die nicht von der Befehlszeile aus angezeigt oder aufgerufen werden können. Sie können sich diese Links ansehen und mit ihnen interagieren, indem Sie das [Online-Referenzhandbuch AWS CLI 1](#) verwenden. Die Referenz enthält auch den Hilfeinhalt für alle AWS CLI Befehle. Die Beschreibungen sind so strukturiert, dass eine einfache Navigation und Anzeige auf Mobilgeräten, Tablets und Desktop-Bildschirmen möglich ist.

API Dokumentation

Alle Befehle in der AWS CLI entsprechen Anfragen, die an die Öffentlichkeit eines AWS Dienstes gestellt wurden. Jeder Dienst, der öffentlich API ist, hat eine API Referenz, die auf der Homepage des Dienstes auf der [AWS Dokumentationswebsite](#) zu finden ist. Der Inhalt einer API Referenz hängt davon ab, wie sie aufgebaut ist und welches Protokoll verwendet wird. In der Regel enthält eine API Referenz detaillierte Informationen zu den vom Dienst unterstützten Vorgängen, zu den an und vom Dienst gesendeten Daten sowie zu allen Fehlerzuständen, die der Dienst melden kann.

API Abschnitte zur Dokumentation

- **Aktionen** – Detaillierte Informationen zu den einzelnen Operationen und ihren Parametern (einschließlich Einschränkungen bei der Länge oder den Inhalten und Standardwerte). Listet die Fehler auf, die bei dieser Operation auftreten können. Jede Operation entspricht einem Unterbefehl in der AWS CLI.
- **Datentypen** – Detaillierte Informationen zu Strukturen, die ein Befehl möglicherweise als Parameter erfordert oder als Reaktion auf eine Anfrage zurückgibt.

- Häufige Parameter – Detaillierte Informationen zu Parametern, die von allen Aktionen für den Service geteilt werden.
- Häufige Fehler – Ausführliche Informationen zu Fehlern, die von den Aktionen eines Services ausgegeben werden können.

Der Name und die Verfügbarkeit jedes Abschnitts kann abhängig vom Service variieren.

Servicespezifisch CLIs

Einige Dienste haben eine separate Version CLI, die aus der Zeit stammt, bevor ein einziger Dienst erstellt AWS CLI wurde, sodass er mit allen Diensten funktioniert. Für diese dienstspezifischen Dienste CLIs gibt es eine separate Dokumentation, die auf der Dokumentationsseite des Dienstes verlinkt ist. Die Dokumentation für dienstspezifische Zwecke gilt CLIs nicht für die. AWS CLI

Behebung von Fehlern

Hilfe zur Diagnose und Behebung von AWS CLI Fehlern finden Sie unter. [Beheben von Fehlern](#)

Weitere Hilfe

Weitere Hilfe bei Ihren AWS CLI Problemen finden Sie in der [AWS CLI Community](#) unter GitHub.

Befehlsstruktur in der AWS CLI

In diesem Thema wird beschrieben, wie der Befehl AWS Command Line Interface (AWS CLI) strukturiert ist und wie die Befehle wait verwendet werden.

Themen

- [Befehlsstruktur](#)
- [Wait-Befehle](#)

Befehlsstruktur

Der AWS CLI verwendet eine mehrteilige Struktur in der Befehlszeile, die in dieser Reihenfolge angegeben werden muss:

1. Basisaufruf des aws-Programms.
2. Der Befehl der obersten Ebene, der in der Regel einem AWS Dienst entspricht, der von der unterstützt wird. AWS CLI
3. Der Unterbefehl, der den auszuführenden Vorgang angibt.
4. Allgemeine AWS CLI Optionen oder Parameter, die für den Vorgang erforderlich sind. Sie können diese in beliebiger Reihenfolge angeben, da diese Informationen nach den ersten drei Teilen aufgeführt werden. Bei mehrfacher Angabe eines exklusiven Parameters wird nur der letzte Wert angewendet.

```
$ aws <command> <subcommand> [options and parameters]
```

Parameter können verschiedene Arten von Eingabewerten annehmen, z. B. Zahlen, Zeichenketten, Listen, Zuordnungen und JSON Strukturen. Was unterstützt wird, hängt von dem angegebenen Befehl und Unterbefehl ab.

Beispiele

Amazon S3

Das folgende Beispiel listet alle Ihre Amazon-S3-Buckets auf.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

Weitere Informationen zu den Amazon-S3-Befehlen finden Sie unter [aws s3](#) in der AWS CLI - Befehlsreferenz.

AWS CloudFormation

Im folgenden [create-change-set](#) Befehlsbeispiel wird der Name des Cloudformation-Stacks in geändert *my-change-set*.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

Weitere Informationen zu den AWS CloudFormation Befehlen finden Sie [aws cloudformation](#) in der AWS CLI Befehlsreferenz.

Wait-Befehle

Für einige AWS Dienste sind `wait` Befehle verfügbar. Jeder Befehl, der `aws wait` verwendet, wartet normalerweise, bis ein Befehl abgeschlossen ist, bevor er zum nächsten Schritt übergeht. Dies ist besonders nützlich für mehrteilige Befehle oder Skripterstellung, da Sie einen Wait-Befehl verwenden können, um zu verhindern, dass zu nachfolgenden Schritten übergegangen wird, wenn der Wait-Befehl fehlschlägt.

Der AWS CLI verwendet eine mehrteilige Struktur in der Befehlszeile für den `wait` Befehl, der in dieser Reihenfolge angegeben werden muss:

1. Basisaufruf des `aws`-Programms.
2. Der Befehl der obersten Ebene, der in der Regel einem AWS Dienst entspricht, der von der unterstützt wird. AWS CLI
3. Der `wait`-Befehl.
4. Der Unterbefehl, der den auszuführenden Vorgang angibt.
5. Allgemeine CLI Optionen oder Parameter, die für den Vorgang erforderlich sind. Sie können diese in beliebiger Reihenfolge angeben, da diese Informationen nach den ersten drei Teilen aufgeführt werden. Bei mehrfacher Angabe eines exklusiven Parameters wird nur der letzte Wert angewendet.

```
$ aws <command> wait <subcommand> [options and parameters]
```

Parameter können verschiedene Arten von Eingabewerten annehmen, z. B. Zahlen, Zeichenketten, Listen, Zuordnungen und JSON Strukturen. Was unterstützt wird, hängt von dem angegebenen Befehl und Unterbefehl ab.

Note

Nicht jeder AWS Dienst unterstützt `wait` Befehle. Sehen Sie im [AWS CLI Referenzhandbuch AWS CLI](#) nach, ob Ihr Dienst `wait` Befehle unterstützt.

Beispiele

AWS CloudFormation

Die folgenden [wait change-set-create-complete](#) Befehlsbeispiele werden erst unterbrochen und wieder aufgenommen, wenn bestätigt werden kann, dass *my-change-set* Änderungssatz im *my-stack* Der Stack ist bereit zur Ausführung.

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

Weitere Informationen zu den AWS CloudFormation wait-Befehlen finden Sie unter [wait](#) in der AWS CLI -Befehlsreferenz.

AWS CodeDeploy

Die folgenden [wait deployment-successful](#) Befehlsbeispiele pausieren, bis *d-A1B2C3111* Die Bereitstellung wurde erfolgreich abgeschlossen.

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

Weitere Informationen zu den AWS CodeDeploy wait-Befehlen finden Sie unter [wait](#) in der AWS CLI -Befehlsreferenz.

Angeben von Parameterwerten in der AWS CLI

Viele der in AWS Command Line Interface (AWS CLI) verwendeten Parameter sind einfache Zeichenfolgen- oder numerische Werte, wie z. B. der Name des Schlüsselpaars *my-key-pair* im folgenden `aws ec2 create-key-pair` Befehlsbeispiel.

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

Die Formatierung des Befehls kann je nach Terminal variieren. Beispielsweise unterscheiden die meisten Terminals zwischen Groß- und Kleinschreibung, aber Powershell unterscheidet nicht zwischen Groß- und Kleinschreibung. Dies bedeutet, dass die beiden folgenden Befehlsbeispiele für Terminals, bei denen zwischen Groß- und Kleinschreibung unterschieden wird, unterschiedliche Ergebnisse liefern würden, da sie *MyFile*.txt* und *myfile*.txt* als unterschiedliche Parameter betrachten.

PowerShell Würde diese Anfragen jedoch genauso verarbeiten, wie es sieht, *MyFile*.txt* und *myfile*.txt* mit denselben Parametern. Das folgende Befehlsbeispiel demonstriert diese Parameter mithilfe des `aws s3 cp` Befehls:

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "MyFile*.txt"  
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "myfile*.txt"
```

Weitere Informationen zur Unterscheidung von Groß- und PowerShell Kleinschreibung finden Sie in der Dokumentation unter [about_Case-Sensitivity](#). PowerShell

Manchmal müssen Sie Anführungszeichen oder Literale für Zeichenketten verwenden, die Sonder- oder Leerzeichen enthalten. Die Regeln für diese Formatierung können zwischen den Terminals auch variieren. Weitere Hinweise zur Verwendung von Anführungszeichen um komplexe Parameter finden Sie unter [Verwenden von Anführungszeichen und Literalen mit Zeichenketten in AWS CLI](#).

In diesen Themen werden die gängigsten Regeln zur Terminalformatierung behandelt. Wenn Sie Probleme damit haben, dass Ihr Terminal Ihre Parameterwerte erkennt, lesen Sie unbedingt die Themen in diesem Abschnitt und lesen Sie auch in der Dokumentation Ihres Terminals nach den spezifischen Syntaxregeln.

Parameter-Themen

- [Allgemeine Parametertypen in der AWS CLI](#)
- [Verwenden von Anführungszeichen und Literalen mit Zeichenketten in AWS CLI](#)
- [Laden von Parametern aus einer Datei in AWS CLI](#)
- [AWS CLI Skelette und Eingabedateien in der AWS CLI](#)
- [Verwendung der Kurzsyntax in der AWS CLI](#)

Allgemeine Parametertypen in der AWS CLI

In diesem Abschnitt werden einige häufige Parametertypen sowie das in der Regel erforderliche Format beschrieben.

Wenn Sie Probleme mit der Formatierung eines Parameters für einen bestimmten Befehl haben, sehen Sie in der Hilfe nach, indem Sie nach dem Befehlsnamen **help** eingeben. Die Hilfe für jeden Unterbefehl enthält den Namen und die Beschreibung einer Option. Der Parametertyp der Option ist in Klammern angegeben. Weitere Informationen zur Anzeige der Hilfe finden Sie unter [the section called "Hilfe"](#).

Parametertypen sind:

- [String](#)

- [Zeitstempel](#)
- [Auflisten](#)
- [Boolesch](#)
- [Ganzzahl](#)
- [Binary/Blob \(Binary Large Object\) und Streaming-Blob](#)
- [Zuordnung](#)
- [Dokument](#)

String

Zeichenkettenparameter können alphanumerische Zeichen, Symbole und Leerzeichen aus dem [ASCII](#) Zeichensatz enthalten. Zeichenfolgen, die Leerzeichen enthalten, müssen in Anführungszeichen gesetzt werden. Wir empfehlen, keine anderen Symbole oder Leerzeichen als das Standardleerzeichen zu verwenden und die [Anführungszeichenregeln](#) Ihres Terminals zu beachten, um unerwartete Ergebnisse zu vermeiden.

Einige Zeichenfolgenparameter können Binärdaten aus einer Datei akzeptieren. Ein Beispiel finden Sie unter [Binärdateien](#).

Zeitstempel

[Zeitstempel werden gemäß dem 8601-Standard formatiert. ISO](#) Diese werden oft als „DateTime“- oder „Date“-Parameter bezeichnet.

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

Zu den akzeptierte Formaten zählen folgende:

- *YYYY-MM-DDThh:mm:ss.sssTZD (UTC)*, zum Beispiel 2014-10-01T 20:30:00.000 Z
- *YYYY-MM-DDThh:mm:ss.sssTZD (with offset)*, zum Beispiel 2014-10-01T 12:30:00.000-08:00
- *YYYY-MM-DD*, zum Beispiel 2014-10-01
- Unix-Zeit in Sekunden, beispielsweise 1 412 195 400. Dies wird manchmal auch als [Unix-Epochenzeit](#) bezeichnet und stellt die Anzahl der Sekunden seit Mitternacht, dem 1. Januar 1970, dar. UTC

Sie können das Zeitstempelformat über die [cli_timestamp_format](#)-Dateieinstellung festlegen.

Auflisten

Eine oder mehrere durch Leerzeichen voneinander getrennte Zeichenfolgen. Wenn eines der Zeichenfolgenelemente ein Leerzeichen enthält, müssen Sie es in Anführungszeichen setzen. Beachten Sie die [Anführungszeichenregeln](#) Ihres Terminals, um unerwartete Ergebnisse zu vermeiden

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

Boolesch

Binäres Flag, das eine Option aktiviert oder deaktiviert. Zum Beispiel verfügt `ec2 describe-spot-price-history` über einen booleschen Parameter `--dry-run`. Wenn dieser angegeben wird, verifiziert er die Abfrage mit dem Service, ohne tatsächlich die Abfrage auszuführen.

```
$ aws ec2 describe-spot-price-history --dry-run
```

Die Ausgabe gibt an, ob der Befehl ordnungsgemäß formatiert wurde oder nicht. Dieser Befehl enthält auch eine Version des Parameters `--no-dry-run`, mit dem Sie explizit angeben können, dass der Befehl normal ausgeführt werden soll. Dies ist allerdings nicht erforderlich, da dies das Standardverhalten ist.

Ganzzahl

Eine nicht signierte ganze Zahl.

```
$ aws ec2 describe-spot-price-history --max-items 5
```

Binary/Blob (Binary Large Object) und Streaming-Blob

In der AWS CLI können Sie einen Binärwert als Zeichenfolge direkt in der Befehlszeile übergeben. Es gibt zwei Arten von Blobs:

- [Blob](#)
- [Streaming-Blob](#)

Blob

Um einen Wert an einen Parameter vom Typ blob zu übergeben, müssen Sie unter Verwendung des Präfix `fileb://` einen Pfad zu einer lokalen Datei angeben, die die Binärdaten enthält. Dateien, auf die mit dem Präfix `fileb://` verwiesen wird, werden immer als unkodierte Binärdateien behandelt. Der angegebene Pfad wird als relativ zum aktuellen Arbeitsverzeichnis interpretiert.

Beispiel: Der Parameter `--plaintext` für `aws kms encrypt` ist ein Blob.

```
$ aws kms encrypt \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --plaintext fileb://ExamplePlaintextFile \
  --output text \
  --query CiphertextBlob | base64 \
  --decode > ExampleEncryptedFile
```

Streaming-Blob

Streaming-Blobs wie z. B. `aws cloudsearchdomain upload-documents` verwenden keine Präfixe. Stattdessen werden Streaming-Blob-Parameter unter Verwendung des direkten Dateipfads formatiert. Im folgenden Beispiel wird der direkte Dateipfad `document-batch.json` für den Befehl `aws cloudsearchdomain upload-documents` verwendet:

```
$ aws cloudsearchdomain upload-documents \
  --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \
  --content-type application/json \
  --documents document-batch.json
```

Zuordnung

Ein Satz von Schlüssel-Wert-Paaren, die in JSON oder unter Verwendung der CLI [Kurzsyntax](#) von angegeben werden. Das folgende JSON Beispiel liest ein Element aus einer Amazon DynamoDB-Tabelle namens `my-table` mit einem Map-Parameter, `--key`. Der Parameter spezifiziert den Primärschlüssel mit dem Namen `id` mit einem Zahlenwert von 1 in einer verschachtelten Struktur.

JSON

Für eine fortgeschrittenere JSON Verwendung in einer Befehlszeile sollten Sie erwägen, einen JSON Befehlszeilenprozessor zu verwenden `jq`, z. B. um JSON Zeichenketten zu erstellen. Weitere Informationen dazu `jq` finden Sie im [jq-Repository](#) unter GitHub.

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N": "1"}}'
```

```
{
  "Item": {
    "name": {
      "S": "John"
    },
    "id": {
      "N": "1"
    }
  }
}
```

Dokument

Note

[Kurzsyntax](#) ist mit Dokumenttypen nicht kompatibel.

Dokumenttypen werden verwendet, um Daten zu senden, ohne dass sie in Zeichenketten JSON eingebettet werden müssen. Der Dokumenttyp ermöglicht Services, willkürliche Schemata bereitzustellen, damit Sie flexiblere Datentypen verwenden können.

Dies ermöglicht das Senden von JSON Daten, ohne dass Werte maskiert werden müssen. Anstatt beispielsweise die folgende JSON Escape-Eingabe zu verwenden:

```
{"document": "{\"key\": true}"}
```

Können Sie den folgenden Dokumenttyp verwenden:

```
{"document": {"key": true}}
```

Gültige Werte für Dokumenttypen

Aufgrund der flexiblen Natur von Dokumenttypen gibt es mehrere gültige Werttypen. Gültige Werte sind unter anderem:

String

```
--option "value"
```

Zahl

```
--option 123
--option 123.456
```

Boolesch

```
--option true
```

Null

```
--option null
```

Array

```
--option ["value1", "value2", "value3"]
--option ["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]
```

Object

```
--option {"key": "value"}
--option {"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":
["value3", "value4"], "key6": {"value5": "value6"}}
```

Verwenden von Anführungszeichen und Literalen mit Zeichenketten in AWS CLI

Es gibt generell zwei Möglichkeiten, wie einfache und doppelte Anführungszeichen in der AWS CLI verwendet werden.

- [Verwenden von Anführungszeichen um Zeichenfolgen, die Leerzeichen enthalten](#)
- [Verwenden von Anführungszeichen in Zeichenfolgen](#)

Verwenden von Anführungszeichen um Zeichenfolgen, die Leerzeichen enthalten

Parameternamen und ihre Werte werden in der Befehlszeile durch Leerzeichen getrennt. Wenn ein Zeichenkettenwert ein eingebettetes Leerzeichen enthält, müssen Sie die gesamte Zeichenfolge in

Anführungszeichen setzen, um zu verhindern, dass das AWS CLI Leerzeichen als Trennzeichen zwischen dem Wert und dem nächsten Parameternamen falsch interpretiert wird. Welche Art von Anführungszeichen Sie verwenden, hängt vom Betriebssystem ab, auf dem Sie das Programm ausführen. AWS CLI

Linux and macOS

Verwenden Sie einfache Anführungszeichen ' '.

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

Weitere Informationen zur Verwendung von Anführungszeichen finden Sie in der Benutzerdokumentation für Ihre bevorzugte Shell.

PowerShell

Einfache Anführungszeichen (empfohlen)

Einfache Anführungszeichen ' ' werden als `verbatim`-Zeichenfolge bezeichnet. Die Zeichenfolge wird genau so an den Befehl übergeben, wie Sie sie eingeben, was bedeutet, dass PowerShell Variablen nicht weitergegeben werden.

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

Doppelte Anführungszeichen

Doppelte Anführungszeichen " " werden als `expandable`-Zeichenfolge bezeichnet. Variablen können in erweiterbaren Zeichenfolgen übergeben werden.

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Weitere Informationen zur Verwendung von Anführungszeichen finden Sie unter [About Rules](#) in den Microsoft PowerShell Docs.

Windows command prompt

Verwenden Sie doppelte Anführungszeichen " ".

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```


Sie können optional den Parameternamen vom Wert durch ein Gleichheitszeichen = statt eines Leerzeichens trennen. Dies ist in der Regel nur erforderlich, wenn der Wert des Parameters mit einem Bindestrich beginnt.

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

Verwenden von Anführungszeichen in Zeichenfolgen

Zeichenfolgen können Anführungszeichen enthalten und Ihre Shell erfordert möglicherweise Escape-Anführungszeichen, damit sie ordnungsgemäß funktionieren. Einer der häufigsten Typen von Parameterwerten ist eine JSON Zeichenfolge. Dies ist komplex, da jeder Elementname und jeder Wert in " " der JSON Struktur Leerzeichen und doppelte Anführungszeichen enthält. Die Art und Weise, wie Sie Parameter JSON im -Format in der Befehlszeile eingeben, ist je nach Betriebssystem unterschiedlich.

Für eine fortgeschrittenere JSON Verwendung in der Befehlszeile sollten Sie die Verwendung eines JSON Befehlszeilenprozessors in Betracht ziehen `jq`, z. B. um Zeichenketten zu erstellen JSON. Weitere Informationen dazu `jq` finden Sie im [jq-Repository](#) unter GitHub.

Linux and macOS

Damit Linux und macOS Zeichenketten wörtlich interpretieren können, verwenden Sie einfache Anführungszeichen, ' ' um die JSON Datenstruktur einzuschließen, wie im folgenden Beispiel. In der JSON Zeichenfolge eingebettete doppelte Anführungszeichen müssen Sie nicht maskieren, da sie wortwörtlich behandelt werden. Da das in einfache Anführungszeichen eingeschlossen JSON ist, müssen alle einfachen Anführungszeichen in der Zeichenfolge maskiert werden. In der Regel wird dazu ein umgekehrter Schrägstrich vor dem einfachen Anführungszeichen verwendet. \'

```
$ aws ec2 run-instances \  
  --image-id ami-12345678 \  
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":  
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]'
```

Weitere Informationen zur Verwendung von Anführungszeichen finden Sie in der Benutzerdokumentation für Ihre bevorzugte Shell.

PowerShell

Verwenden Sie einfache Anführungszeichen ' ' oder doppelte Anführungszeichen " ".

Einfache Anführungszeichen (empfohlen)

Einfache Anführungszeichen ' ' werden als `verbatim`-Zeichenfolge bezeichnet. Die Zeichenfolge wird genau so an den Befehl übergeben, wie Sie sie eingeben, was bedeutet, dass PowerShell Variablen nicht weitergeleitet werden.

Da JSON Datenstrukturen doppelte Anführungszeichen enthalten, empfehlen wir einfache Anführungszeichen ' ', um sie einzuschließen. Wenn Sie einfache Anführungszeichen verwenden, müssen Sie in der JSON Zeichenfolge eingebettete doppelte Anführungszeichen nicht maskieren. Sie müssen jedoch jedes einzelne Anführungszeichen ` innerhalb der JSON Struktur mit einem Backtick maskieren.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]`
```

Doppelte Anführungszeichen

Doppelte Anführungszeichen " " werden als `expandable`-Zeichenfolgen bezeichnet. Variablen können in erweiterbaren Zeichenfolgen übergeben werden.

Wenn Sie doppelte Anführungszeichen verwenden, müssen Sie die in der JSON Zeichenfolge eingebetteten einfachen Anführungszeichen nicht maskieren. Sie müssen jedoch jedes doppelte Anführungszeichen mit einem Backtick ` innerhalb der JSON Struktur maskieren, wie im folgenden Beispiel.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings "[{"DeviceName`":"/dev/sdb`",`"Ebs`":
{"VolumeSize`":20,`"DeleteOnTermination`":false,`"VolumeType`":`"standard`"}]`"
```

Weitere Informationen zur Verwendung von Anführungszeichen finden Sie unter [About Rules](#) in den Microsoft PowerShell Docs.

Warning

Before PowerShell sendet einen Befehl an die AWS CLI und bestimmt, ob Ihr Befehl anhand typischer Regeln PowerShell oder mit `CommandLineToArgvW` Anführungszeichen interpretiert wird. Wenn PowerShell Prozesse

verwenden `CommandLineToArgvW`, müssen Sie Zeichen mit einem `\` umgekehrten Schrägstrich maskieren.

Weitere Informationen dazu finden Sie unter [Was ist mit der seltsamen Behandlung von Anführungszeichen und Backslashes durch CommandLineToArgv W](#) in Microsoft DevBlogs, [Jeder zitiert Befehlszeilenargumente falsch im](#) Microsoft Docs-Blog und [CommandLineToArgvW-Funktion](#) in den Microsoft Docs. `CommandLineToArgvW` PowerShell

Einfache Anführungszeichen

Einfache Anführungszeichen `' '` werden als `verbatim`-Zeichenfolge bezeichnet.

Die Zeichenfolge wird genau so an den Befehl übergeben, wie Sie sie eingeben, was bedeutet, dass PowerShell Variablen nicht weitergeleitet werden. Verwenden Sie einen Backslash `\` als Escape-Zeichen.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings '[{"DeviceName\":"\\/dev/sdb\\",\"Ebs\":"
  {\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":"standard\"}]']`
```

Doppelte Anführungszeichen

Doppelte Anführungszeichen `" "` werden als `expandable`-Zeichenfolgen bezeichnet.

Variablen können in `expandable`-Zeichenfolgen übergeben werden. Bei Zeichenketten in doppelten Anführungszeichen müssen Sie zweimal die Escape-Methode verwenden `\\` für jedes Anführungszeichen, anstatt nur einen Backtick zu verwenden. Der Backtick maskiert den Backslash, und der Backslash wird als Escape-Zeichen für den `CommandLineToArgvW`-Prozess verwendet.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings "[{\"DeviceName \\\":\"\\/dev/sdb \\", \"Ebs \\\":
  {\"VolumeSize \\\":20, \"DeleteOnTermination \\\":false, \"VolumeType \\\": `
  \\\"standard \\\"}]"]`
```

Blobs (empfohlen)

Um die PowerShell Anführungsregeln für die JSON Dateneingabe zu umgehen, verwenden Sie Blobs, um Ihre JSON Daten direkt an die zu übergeben. AWS CLI Weitere Informationen zu Blobs finden Sie unter [Blob](#).

Windows command prompt

Die Windows-Eingabeaufforderung erfordert doppelte Anführungszeichen " ", um die Datenstruktur einzuschließen. JSON Um zu verhindern, dass der Befehlsprozessor die in der eingebetteten doppelten Anführungszeichen falsch interpretiertJSON, müssen Sie außerdem jedes doppelte Anführungszeichen " innerhalb der JSON Datenstruktur selbst maskieren (mit einem umgekehrten \ Schrägstrich voranstellen), wie im folgenden Beispiel.

```
C:\> aws ec2 run-instances ^
  --image-id ami-12345678 ^
  --block-device-mappings "[{\ "DeviceName\":"\ /dev/sdb\ ",\ "Ebs\ ":
{\ "VolumeSize\ ":20,\ "DeleteOnTermination\ ":false,\ "VolumeType\ ":"standard\ "}]]"
```

Nur die äußersten doppelten Anführungszeichen benötigen kein Escape-Zeichen.

Laden von Parametern aus einer Datei in AWS CLI

Einige Parameter erwarten Dateinamen als Argumente, aus denen die Daten AWS CLI geladen werden. Mit anderen Parametern können Sie den Parameterwert als Text angeben, der in die Befehlszeile eingegeben wird oder aus einer Datei gelesen wird. Unabhängig davon, ob eine Datei erforderlich oder optional ist, müssen Sie die Datei korrekt codieren, damit sie verstanden AWS CLI werden kann. Die Kodierung der Datei muss mit dem Standardgebietsschema des Lesesystems übereinstimmen. Sie können dies mithilfe der Python-Methode `locale.getpreferredencoding()` bestimmen.

Note

Standardmäßig PowerShell gibt Windows Text als UTF -16 aus, was zu einem Konflikt mit der UTF -8-Kodierung führt, die von JSON Dateien und vielen Linux-Systemen verwendet wird. Wir empfehlen Ihnen, dies `-Encoding ascii` zusammen mit Ihren PowerShell Out-File Befehlen zu verwenden, um sicherzustellen, dass die resultierende AWS CLI Datei gelesen werden kann.

Themen

- [Laden von Parametern aus einer Datei](#)
- [Binärdateien](#)

- [Remote-Dateien](#)

Laden von Parametern aus einer Datei

Manchmal ist es praktisch, einen Parameterwert aus einer Datei zu laden, anstatt zu versuchen, alles als Befehlszeilenparameterwert einzugeben, z. B. wenn es sich bei dem Parameter um eine komplexe JSON Zeichenfolge handelt. Um eine Datei anzugeben, die den Wert enthält, geben Sie eine Datei URL im folgenden Format an.

```
file://complete/path/to/file
```

- Die ersten beiden Schrägstriche "/" sind Teil der Spezifikation. Wenn der erforderliche Pfad mit einem "/" beginnt, besteht das Ergebnis aus drei Schrägstrichen: `file:///folder/file`.
- Das URL gibt den Pfad zu der Datei an, die den eigentlichen Parameterinhalt enthält.
- Wenn Sie Dateien mit Leerzeichen oder Sonderzeichen verwenden, befolgen Sie die [Anführungszeichen- und Escape-Regeln](#) Ihres Terminals.

Note

Dieses Verhalten wird automatisch für Parameter deaktiviert, die bereits einen erwartenURL, z. B. für einen Parameter, der eine AWS CloudFormation Vorlage identifiziertURL. Sie können dieses Verhalten auch deaktivieren, indem Sie die [cli_follow_urlparam](#)Einstellung in Ihrer AWS CLI Konfigurationsdatei deaktivieren.

Die Dateipfade in den folgenden Beispielen werden als relativ zum aktuellen Arbeitsverzeichnis interpretiert.

Linux or macOS

```
// Read from a file in the current directory
$ aws ec2 describe-instances --filters file://filter.json

// Read from a file in /tmp
$ aws ec2 describe-instances --filters file:///tmp/filter.json

// Read from a file with a filename with whitespaces
```

```
$ aws ec2 describe-instances --filters 'file://filter content.json'
```

Windows command prompt

```
// Read from a file in C:\temp  
C:\> aws ec2 describe-instances --filters file://C:\temp\filter.json  
  
// Read from a file with a filename with whitespaces  
C:\> aws ec2 describe-instances --filters "file://C:\temp\filter content.json"
```

Die Präfixoption `file://` unterstützt Erweiterungen im Unix-Stil, einschließlich `~/`, `./` und `../`. Unter Windows erfolgt mit dem Ausdruck `~/` die Erweiterung zu Ihrem Benutzerverzeichnis, das in der Umgebungsvariablen `%USERPROFILE%` gespeichert ist. Beispielsweise befindet sich das Benutzerverzeichnis in Windows 10 in der Regel unter `C:\Users\UserName\`.

JSONDokumente, die als Wert eines anderen JSON Dokuments eingebettet sind, müssen trotzdem maskiert werden.

```
$ aws sqs create-queue --queue-name my-queue --attributes file://attributes.json
```

attributes.json

```
{  
  "RedrivePolicy": "{\\"deadLetterTargetArn\\":\\"arn:aws:sqs:us-west-2:0123456789012:deadletter\\", \\"maxReceiveCount\\":\\"5\\"}"  
}
```

Binärdateien

Für Befehle, die Binärdaten als Parameter annehmen, geben Sie mit dem Präfix `fileb://` an, dass es sich bei den Daten um binäre Inhalte handelt. Zu den Befehlen, die Binärdaten akzeptieren, zählen:

- **aws ec2 run-instances:** `--user-data-Parameter`.
- **aws s3api put-object:** `--sse-customer-key-Parameter`.
- **aws kms decrypt:** `--ciphertext-blob-Parameter`.

Das folgende Beispiel generiert mithilfe eines Linux-Befehlszeilentools einen binären AES 256-Bit-Schlüssel und stellt ihn dann Amazon S3 zur serverseitigen Verschlüsselung einer hochgeladenen Datei zur Verfügung.

```
$ dd if=/dev/urandom bs=1 count=32 > sse.key
32+0 records in
32+0 records out
32 bytes (32 B) copied, 0.000164441 s, 195 kB/s
$ aws s3api put-object \
  --bucket amzn-s3-demo-bucket \
  --key test.txt \
  --body test.txt \
  --sse-customer-key fileb://sse.key \
  --sse-customer-algorithm AES256
{
  "SSECustomerKeyMD5": "iVg8oWa8sy714+FjtesrJg==",
  "SSECustomerAlgorithm": "AES256",
  "ETag": "\"a6118e84b76cf98bf04bbe14b6045c6c\""
}
```

Remote-Dateien

Das unterstützt AWS CLI auch das Laden von Parametern aus einer im Internet gehosteten Datei mit einem oder. `http://` `https://` URL Im folgenden Beispiel wird eine Datei, die in einem Amazon-S3-Bucket gespeichert ist, referenziert. Auf diese Weise können Sie von jedem Computer aus auf Parameterdateien zugreifen. Dabei muss der Container öffentlich zugänglich sein.

```
$ aws ec2 run-instances \
  --image-id ami-12345678 \
  --block-device-mappings http://amzn-s3-demo-bucket.s3.amazonaws.com/filename.json
```

Im vorherigen Beispiel wird davon ausgegangen, dass die Datei die folgenden JSON Daten `filename.json` enthält.

```
[
  {
    "DeviceName": "/dev/sdb",
    "Ebs": {
      "VolumeSize": 20,
      "DeleteOnTermination": false,
      "VolumeType": "standard"
    }
  }
]
```

```
}  
}  
]
```

Ein weiteres Beispiel, das auf eine Datei verweist, die Parameter im JSON -Format enthält, finden Sie unter. [Einem Benutzer eine IAM verwaltete Richtlinie zuordnen](#)

AWS CLI Skelette und Eingabedateien in der AWS CLI

Die meisten AWS CLI Befehle akzeptieren alle Parametereingaben aus einer Datei. Diese Vorlagen können mit der `generate-cli-skeleton`-Option generiert werden.

Themen

- [Über AWS CLI Skelette und Eingabedateien](#)
- [Generieren eines Befehls-Skeletons](#)

Über AWS CLI Skelette und Eingabedateien

Die meisten Befehle AWS Command Line Interface (AWS CLI) unterstützen die Fähigkeit, alle Parametereingaben aus einer Datei mit den `--cli-input-yaml` Parametern `--cli-input-json` zu akzeptieren.

Dieselben Befehle stellen hilfreich den `--generate-cli-skeleton` Parameter bereit, mit dem Sie eine Datei mit allen Parametern generieren können, die Sie bearbeiten und ausfüllen können. Anschließend können Sie den Befehl mit dem relevanten Parameter `--cli-input-json` ausführen und auf die gefüllte Datei zeigen.

Important

Einige AWS CLI Befehle lassen sich nicht direkt einzelnen AWS API Vorgängen zuordnen, wie z. B. die [aws s3](#). Diese Befehle unterstützen nicht die in diesem Thema beschriebenen Parameter `--generate-cli-skeleton` oder `--cli-input-json`. Wenn Sie nicht wissen, ob ein bestimmter Befehl diese Parameter unterstützt, führen Sie den folgenden Befehl aus und ersetzen Sie den *service* and *command* Namen mit denen, die Sie interessieren.

```
$ aws service command help
```


Die Ausgabe enthält einen Abschnitt namens `Synopsis`, der die Parameter zeigt, die von dem angegebenen Befehl unterstützt werden.

```
$ aws iam list-users help
...
SYNOPSIS
    list-users
    ...
    [--cli-input-json]
    ...
    [--generate-cli-skeleton <value>]
...
```

Der Parameter `--generate-cli-skeleton` bewirkt, dass der Befehl nicht ausgeführt wird. Stattdessen wird eine Parametervorlage generiert und angezeigt, die Sie anpassen und bei einem späteren Befehl als Eingabe verwenden können. Die generierte Vorlage enthält alle Parameter, die der Befehl unterstützt.

Der `--generate-cli-skeleton`-Parameter akzeptiert einen der folgenden Werte:

- `input`— Die generierte Vorlage enthält alle Eingabeparameter, die als JSON formatiert sind. Dies ist der Standardwert.
- `output`— Die generierte Vorlage enthält alle Ausgabeparameter, die als formatiert sind. JSON

Da AWS CLI es sich im Grunde um einen „Wrapper“ um den Dienst handelt API, erwartet die Skelettdatei, dass Sie auf alle Parameter anhand ihrer zugrunde liegenden API Parameternamen verweisen. Dies unterscheidet sich wahrscheinlich vom AWS CLI Parameternamen. Beispielsweise `user-name` könnte ein benannter AWS CLI Parameter dem genannten API Parameter des AWS Dienstes zugeordnet sein `UserName` (beachten Sie die geänderte Groß-/Kleinschreibung und den fehlenden Bindestrich). Wir empfehlen, die Option `--generate-cli-skeleton` zu verwenden, um die Vorlage mit den „richtigen“ Parameternamen zu generieren und so Fehler zu vermeiden. Die erwarteten Parameternamen finden Sie auch im API Referenzhandbuch für den Service. Sie können alle Parameter aus der Vorlage löschen, die nicht erforderlich sind und für die Sie keinen Wert angeben möchten.

Wenn Sie beispielsweise den folgenden Befehl ausführen, generiert er die Parametervorlage für den Befehl Amazon Elastic Compute Cloud (AmazonEC2)`run-instances`.

JSON

Das folgende Beispiel zeigt, wie Sie eine Vorlage generieren, die JSON mithilfe des Standardwerts (`input`) für den `--generate-cli-skeleton` Parameter formatiert ist.

```
$ aws ec2 run-instances --generate-cli-skeleton
```

```
{
  "DryRun": true,
  "ImageId": "",
  "MinCount": 0,
  "MaxCount": 0,
  "KeyName": "",
  "SecurityGroups": [
    ""
  ],
  "SecurityGroupIds": [
    ""
  ],
  "UserData": "",
  "InstanceType": "",
  "Placement": {
    "AvailabilityZone": "",
    "GroupName": "",
    "Tenancy": ""
  },
  "KernelId": "",
  "RamdiskId": "",
  "BlockDeviceMappings": [
    {
      "VirtualName": "",
      "DeviceName": "",
      "Ebs": {
        "SnapshotId": "",
        "VolumeSize": 0,
        "DeleteOnTermination": true,
        "VolumeType": "",
        "Iops": 0,
        "Encrypted": true
      },
      "NoDevice": ""
    }
  ],
}
```

```
"Monitoring": {
  "Enabled": true
},
"SubnetId": "",
"DisableApiTermination": true,
"InstanceInitiatedShutdownBehavior": "",
"PrivateIpAddress": "",
"ClientToken": "",
"AdditionalInfo": "",
"NetworkInterfaces": [
  {
    "NetworkInterfaceId": "",
    "DeviceIndex": 0,
    "SubnetId": "",
    "Description": "",
    "PrivateIpAddress": "",
    "Groups": [
      ""
    ],
    "DeleteOnTermination": true,
    "PrivateIpAddresses": [
      {
        "PrivateIpAddress": "",
        "Primary": true
      }
    ],
    "SecondaryPrivateIpAddressCount": 0,
    "AssociatePublicIpAddress": true
  }
],
"IamInstanceProfile": {
  "Arn": "",
  "Name": ""
},
"EbsOptimized": true
}
```

Generieren eines Befehls-Skeletons

So generieren und verwenden Sie eine Parameter-Skeleton-Datei

1. Führen Sie den Befehl mit dem `--generate-cli-skeleton` Parameter aus, um zu erzeugen, YAML und leiten Sie die Ausgabe in eine Datei um, um sie zu speichern.

JSON

```
$ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
```

2. Öffnen Sie die Parameter-Skeleton-Datei in Ihrem Texteditor und entfernen Sie alle Parameter, die Sie nicht benötigen. Beispielsweise können Sie die Vorlage auf Folgendes beschränken. Vergewissern Sie sich, dass die Datei noch gültig ist JSON ob Sie die Elemente, die Sie nicht benötigen, entfernt haben.

JSON

```
{
  "DryRun": true,
  "ImageId": "",
  "KeyName": "",
  "SecurityGroups": [
    ""
  ],
  "InstanceType": "",
  "Monitoring": {
    "Enabled": true
  }
}
```

In diesem Beispiel belassen wir den `DryRun` Parametersatz auf, `true` um die Amazon EC2 Dry-Run-Funktion zu verwenden. Mit dieser Funktion können Sie den Befehl sicher testen, ohne tatsächlich Ressourcen zu erstellen oder zu ändern.

3. Füllen Sie den Rest mit Werten auf, die sich für Ihr Szenario eignen. In diesem Beispiel geben wir den Instance-Typ, den Schlüsselnamen, die Sicherheitsgruppe und die ID des zu verwendenden Amazon Machine Image (AMI) an. In diesem Beispiel wird von der AWS Standardregion ausgegangen. Das AMI `ami-dfc39aef` ist ein 64-Bit-Image von Amazon Linux,

das in der `us-west-2` Region gehostet wird. Wenn Sie eine andere Region verwenden, müssen Sie [die richtige AMI ID finden](#).

JSON

```
{
  "DryRun": true,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

4. Führen Sie den Befehl mit den abgeschlossenen Parametern aus, indem Sie die fertige Vorlagendatei mit dem `--cli-input-json`-Präfix an den Parameter `cli-input-yaml` übergeben. Der AWS CLI interpretiert den Pfad als relativ zu Ihrem aktuellen Arbeitsverzeichnis. Im folgenden Beispiel, in dem nur der Dateiname ohne Pfad angezeigt wird, wird direkt im aktuellen Arbeitsverzeichnis nach der Datei gesucht.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json
```

```
A client error (DryRunOperation) occurred when calling the RunInstances
operation: Request would have succeeded, but DryRun flag is set.
```

Der Probelauffehler weist darauf hin, dass das JSON korrekt formatiert YAML ist und dass die Parameterwerte gültig sind. Wenn andere Probleme in der Ausgabe gemeldet werden, beheben Sie sie und wiederholen Sie den vorherigen Schritt, bis die Meldung „Request would have succeeded“ angezeigt wird.

5. Jetzt können Sie den Parameter `DryRun` auf `false` setzen, um den Probelauf zu deaktivieren.

JSON

```
{
  "DryRun": false,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

6. Führen Sie den Befehl aus und starten Sie `run-instances` tatsächlich eine EC2 Amazon-Instance und zeigt die Details an, die durch den erfolgreichen Start generiert wurden. Das Format der Ausgabe wird vom `--output`-Parameter unabhängig vom Format Ihrer Eingabeparametervorlage gesteuert.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
```

```
{
  "OwnerId": "123456789012",
  "ReservationId": "r-d94a2b1",
  "Groups": [],
  "Instances": [
    ...
  ]
}
```

Verwendung der Kurzsyntax in der AWS CLI

Die AWS Command Line Interface (AWS CLI) kann viele ihrer Optionsparameter im Format akzeptieren. JSON Es kann jedoch mühsam sein, große JSON Listen oder Strukturen in der Befehlszeile einzugeben. Um dies zu vereinfachen, unterstützt der AWS CLI auch eine Kurzsyntax, die eine einfachere Darstellung Ihrer Optionsparameter ermöglicht als die Verwendung des vollständigen Formats. JSON

Themen

- [Strukturparameter](#)
- [Verwenden Sie die Kurzsyntax mit dem AWS Command Line Interface](#)

Strukturparameter

Die Kurzsyntax in der AWS CLI erleichtert es Benutzern, Parameter einzugeben, die flach sind (nicht verschachtelte Strukturen). Das Format ist eine durch Kommata getrennte Liste von Schlüssel-Wert-Paaren. Verwenden Sie die geeigneten [Anführungszeichen-](#) und Escape-Zeichen-Regeln für Ihr Terminal, da es sich bei der Kurznotation um Zeichenfolgen handelt.

Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

PowerShell

```
--option "key1=value1,key2=value2,key3=value3"
```

Beide entsprechen dem folgenden Beispiel, formatiert in. JSON

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}
```

Zwischen den einzelnen kommagetrennten Schlüssel-Wert-Paaren sind keine Leerzeichen erlaubt. Hier sehen Sie ein Beispiel für den `update-table`-Amazon-DynamoDB-Befehl mit der Option `--provisioned-throughput` in der Kurznotation.

```
$ aws dynamodb update-table \  
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \  
  --table-name MyDDBTable
```

Dies entspricht dem folgenden Beispiel, formatiert in. JSON

```
$ aws dynamodb update-table \  
  --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \  
  --table-name MyDDBTable
```

Verwenden Sie die Kurzsyntax mit dem AWS Command Line Interface

Sie können Eingabeparameter in Listenform auf zwei Arten angeben: oder in Kurzform. JSON Die Syntax-Kurznotation in der AWS CLI erleichtert die Eingabe von Listen mit Zahlen, Zeichenfolgen oder nicht geschachtelten Strukturen.

Das grundlegende Format wird hier dargestellt, wobei die Werte in der Liste durch ein einzelnes Leerzeichen voneinander getrennt sind.

```
--option value1 value2 value3
```

Dies entspricht dem folgenden Beispiel, formatiert in JSON

```
--option '[value1,value2,value3]'
```

Wie bereits erwähnt, können Sie eine Liste von Zahlen, eine Liste mit Zeichenfolgen oder eine Liste nicht geschachtelter Strukturen in Kurznotation angeben. Im Folgenden finden Sie ein Beispiel für den `stop-instances` Befehl für Amazon Elastic Compute Cloud (AmazonEC2), bei dem der Eingabeparameter (Liste von Zeichenketten) für die `--instance-ids` Option in Kurzform angegeben wird.

```
$ aws ec2 stop-instances \  
  --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

Dies entspricht dem folgenden Beispiel, das in formatiert ist. JSON

```
$ aws ec2 stop-instances \  
  --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

Das folgende Beispiel zeigt den EC2 `create-tags` Amazon-Befehl, der eine Liste von nicht verschachtelten Strukturen für die `--tags` Option verwendet. Die Option `--resources` gibt die ID der Instance an, die markiert werden soll.

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2  
Key=My3rdTag,Value=Value3
```


Dies entspricht dem folgenden Beispiel, formatiert in JSON. Der JSON Parameter wird aus Gründen der Lesbarkeit über mehrere Zeilen geschrieben.

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags '[  
    {"Key": "My1stTag", "Value": "Value1"},  
    {"Key": "My2ndTag", "Value": "Value2"},  
    {"Key": "My3rdTag", "Value": "Value3"}  
  ]'
```

Steuerung der Befehlsausgabe in der AWS CLI

In diesem Abschnitt werden die verschiedenen Möglichkeiten beschrieben, die Ausgabe von AWS Command Line Interface (AWS CLI) zu steuern. Durch das Anpassen der AWS CLI Ausgabe in Ihrem Terminal können Sie die Lesbarkeit verbessern, die Skriptautomatisierung optimieren und die Navigation durch größere Datensätze vereinfachen.

Einige Dienste verfügen über eine serverseitige [Paginierung](#) für ihre Daten zusätzliche Paginierungsoptionen.

Schließlich AWS CLI bietet der sowohl [serverseitige als auch clientseitige Filterung](#), die Sie einzeln oder zusammen verwenden können, um Ihre Ausgabe zu filtern. AWS CLI

Themen

- [Sensible Ausgabe](#)
- [Serverseitige und clientseitige Ausgabeoptionen](#)
- [Einstellung des Ausgabeformats in der AWS CLI](#)
- [Verwenden der Paginierungsoptionen in der AWS CLI](#)
- [Ausgabe filtern in AWS CLI](#)

Sensible Ausgabe

Einige Operationen von geben AWS CLI möglicherweise Informationen zurück, die als sensibel angesehen werden könnten, einschließlich Informationen aus Umgebungsvariablen. Die Offenlegung dieser Informationen kann in bestimmten Szenarien ein Sicherheitsrisiko darstellen. Beispielsweise könnten die Informationen in CI/CD-Protokollen (Continuous Integration and Continuous Deployment)

enthalten sein. Es ist daher wichtig, dass Sie überprüfen, wann Sie solche Ausgaben in Ihre Protokolle aufnehmen, und die Ausgabe unterdrücken, wenn sie nicht benötigt werden.

Weitere Informationen zum Schutz sensibler Daten finden Sie unter [the section called “Datenschutz”](#).

Beachten Sie die folgenden bewährten Methoden:

- Erwägen Sie, Ihre Secrets programmatisch aus einem Secrets-Speicher abzurufen, wie z. AWS Secrets Manager
- Überprüfen Sie den Inhalt Ihrer Build-Logs, um sicherzustellen, dass sie keine vertraulichen Informationen enthalten. Erwägen Sie Methoden wie die Weiterleitung an die Ausgabe `/dev/null` oder deren Erfassung als Bash oder PowerShell Variable, um Befehlsausgaben zu unterdrücken.

Im Folgenden finden Sie ein Bash-Beispiel für die Umleitung von Ausgaben, aber nicht von Fehlern, an: `/dev/null`

```
$ aws s3 ls > /dev/null
```

Einzelheiten zur Unterdrückung der Ausgabe für Ihr Terminal finden Sie in der Benutzerdokumentation des von Ihnen verwendeten Terminals.

- Berücksichtigen Sie den Zugriff auf Ihre Protokolle und legen Sie den Umfang des Zugriffs entsprechend Ihrem Anwendungsfall fest.

Serverseitige und clientseitige Ausgabeoptionen

Der AWS CLI bietet sowohl [serverseitige als auch clientseitige Filterung](#), die Sie einzeln oder zusammen verwenden können, um Ihre Ausgabe zu filtern. AWS CLI Die serverseitige Filterung wird zuerst verarbeitet und gibt Ihre Ausgabe für die clientseitige Filterung zurück. Die serverseitige Filterung wird vom Dienst unterstützt. API Die clientseitige Filterung wird vom AWS CLI Client mithilfe des Parameters unterstützt. `--query`

Serverseitige Ausgabeoptionen sind Funktionen, die direkt von der unterstützt werden. AWS-Service API Alle Daten, die gefiltert oder ausgelagert werden, werden nicht an den Client gesendet. Dadurch können die HTTP Reaktionszeiten verkürzt und die Bandbreite für größere Datensätze verbessert werden.

Clientseitige Ausgabeoptionen sind Funktionen, die von der erstellt wurden. AWS CLI Alle Daten werden an den Client gesendet, dann die AWS CLI Filter oder Seiten, die den angezeigten Inhalt

anzeigen. Bei clientseitigen Vorgängen werden bei größeren Datensätzen weder Geschwindigkeit noch Bandbreite eingespart.

Wenn serverseitige und clientseitige Optionen zusammen verwendet werden, werden serverseitige Operationen zuerst abgeschlossen und dann für clientseitige Operationen an den Client gesendet. Dabei werden die potenziellen Geschwindigkeits- und Bandbreiteneinsparungen serverseitiger Optionen genutzt und gleichzeitig zusätzliche AWS CLI Funktionen verwendet, um die gewünschte Ausgabe zu erzielen.

Einstellung des Ausgabeformats in der AWS CLI

In diesem Thema werden die verschiedenen Ausgabeformate für AWS Command Line Interface (AWS CLI) beschrieben. AWS CLI unterstützt die folgenden Ausgabeformate:

- **json**— Die Ausgabe ist als [JSON](#) Zeichenfolge formatiert.
- **text** – Die Ausgabe wird als mehrere Zeilen mit tabulatorgetrennten Zeichenfolgenwerten formatiert. Dies kann nützlich sein, um die Ausgabe an einen Textprozessor wie `grep`, `sed` oder `awk` zu übergeben.
- **table** – Die Ausgabe erfolgt in Form einer Tabelle mit den Zeichen `+|-`, um die Zellenrahmen zu bilden. Normalerweise wird die Information in einem benutzerfreundlichen Format wiedergegeben, das viel einfacher zu lesen ist als die anderen, jedoch programmatisch nicht so nützlich ist.

Auswählen des Ausgabeformats

Wie im Thema [Konfiguration](#) erläutert, gibt es drei verschiedene Methoden, das Ausgabeformat anzugeben:

- Verwenden der Option **output** in einem benannten Profil in der **config**-Datei – Im folgenden Beispiel wird das Standardausgabeformat auf `text` festgelegt.

```
[default]
output=text
```

- Verwenden der Umgebungsvariablen **AWS_DEFAULT_OUTPUT** – Die folgende Ausgabe legt das Format für die Befehle in dieser Befehlszeilensitzung auf `table` fest, bis die Variable geändert wird oder die Sitzung endet. Diese Umgebungsvariable überschreibt alle Werte, die in der Datei `config` festgelegt sind.

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- Verwenden der Option **--output** in der Befehlszeile – Im folgenden Beispiel wird nur die Ausgabe dieses einen Befehls auf `json` gesetzt. Mit dieser Option für den Befehl werden alle aktuell festgelegten Umgebungsvariablen überschrieben oder der Wert in der Datei `config`.

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

Important

Der von Ihnen angegebene Ausgabebetyp ändert die Funktionsweise der Option `--query`:

- Wenn Sie angeben `--output text`, wird die Ausgabe paginiert, bevor der `--query` Filter angewendet wird, und die Abfrage wird auf jeder Seite der Ausgabe einmal AWS CLI ausgeführt. Aus diesem Grund enthält die Abfrage das erste passende Element auf jeder Seite, was zu unerwarteten zusätzlichen Ausgaben führen kann. Um die Ausgabe zusätzlich zu filtern, können Sie andere Befehlszeilentools wie `head` oder `tail` verwenden.
- Wenn Sie `--output json` angeben, wird die Ausgabe vollständig als einzelne, native Struktur verarbeitet, bevor der `--query`-Filter angewendet wird. Der AWS CLI führt die Abfrage nur einmal für die gesamte Struktur aus, wodurch ein gefiltertes Ergebnis erzeugt wird, das dann ausgegeben wird.

JSON-Ausgabeformat

[JSON](#) ist das Standardausgabeformat von AWS CLI. Die meisten Programmiersprachen können JSON Zeichenketten mithilfe integrierter Funktionen oder mit öffentlich zugänglichen Bibliotheken problemlos dekodieren. Sie können die JSON Ausgabe auf leistungsstarke Weise mit der [Option --query](#) kombinieren, um die mit AWS CLI JSON -formatierte Ausgabe zu filtern und zu formatieren.

Für komplexere Filterung, mit der Sie möglicherweise nicht arbeiten können `--query`, können Sie einen Befehlszeilenprozessor in Betracht ziehen `jq`. JSON Sie können dies unter <http://stedolan.github.io/jq/> herunterladen und finden hier auch die offizielle praktische Anleitung.

Das Folgende ist ein Beispiel für eine JSON Ausgabe.

```
$ aws iam list-users --output json
```

```
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDA111111111111EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/Admin",
      "CreateDate": "2014-10-16T16:03:09+00:00",
      "PasswordLastUsed": "2016-06-03T18:37:29+00:00"
    },
    {
      "Path": "/backup/",
      "UserName": "backup-user",
      "UserId": "AIDA222222222222EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/backup/backup-user",
      "CreateDate": "2019-09-17T19:30:40+00:00"
    },
    {
      "Path": "/",
      "UserName": "cli-user",
      "UserId": "AIDA333333333333EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/cli-user",
      "CreateDate": "2019-09-17T19:11:39+00:00"
    }
  ]
}
```

Textausgabeformat

Das text Format organisiert die AWS CLI Ausgabe in tabulatorgetrennten Zeilen. Es funktioniert gut mit herkömmlichen Unix-Textwerkzeugen wiegrep, sedawk, und und der Textverarbeitung von PowerShell

Das text-Ausgabeformat entspricht der unten gezeigten grundlegenden Struktur. Die Spalten sind alphabetisch nach den entsprechenden Schlüsselnamen des zugrunde liegenden JSON Objekts sortiert.

```
IDENTIFIER sorted-column1 sorted-column2
IDENTIFIER2 sorted-column1 sorted-column2
```

Es folgt ein Beispiel für die `text`-Ausgabe. Jedes Feld ist durch einen Tabulator getrennt von den anderen, mit einem zusätzlichen Tabulator, wenn ein leeres Feld vorhanden ist.

```
$ aws iam list-users --output text
```

```
USERS   arn:aws:iam::123456789012:user/Admin                2014-10-16T16:03:09+00:00
2016-06-03T18:37:29+00:00 / AIDA111111111111EXAMPLE Admin
USERS   arn:aws:iam::123456789012:user/backup/backup-user  2019-09-17T19:30:40+00:00
/backup/ AIDA222222222222EXAMPLE backup-user
USERS   arn:aws:iam::123456789012:user/cli-user           2019-09-17T19:11:39+00:00
/ AIDA333333333333EXAMPLE cli-user
```

Die vierte Spalte ist das `PasswordLastUsed`-Feld und ist für die letzten beiden Einträge leer, da sich diese Benutzer nie bei der AWS Management Console anmelden.

Important

Wir empfehlen dringend, dass Sie, wenn Sie eine `text`-Ausgabe angeben, immer die Option `--query` verwenden, um ein konsistentes Verhalten zu gewährleisten.

Das liegt daran, dass das Textformat die Ausgabespalten alphabetisch nach dem Schlüsselnamen des zugrunde liegenden JSON Objekts sortiert, das vom AWS Dienst zurückgegeben wird, und ähnliche Ressourcen möglicherweise nicht dieselben Schlüsselnamen haben. Beispielsweise kann die JSON Darstellung einer Linux-basierten EC2 Amazon-Instance Elemente enthalten, die in der JSON Darstellung einer Windows-basierten Instance nicht vorhanden sind, oder umgekehrt. In zukünftigen Aktualisierungen können auch Schlüssel-Wert-Elemente zu Ressourcen hinzugefügt oder aus diesen entfernt werden, wodurch sich die Spaltensortierung ändert. In diesem Fall erweitert `--query` die Funktionalität der `text`-Ausgabe, um eine vollständige Kontrolle über das Ausgabeformat zu ermöglichen.

Im folgenden Beispiel gibt der Befehl an, welche Elemente angezeigt werden sollen. definiert die Reihenfolge der Spalten über die Listennotation `[key1, key2, ...]`. Auf diese Weise können Sie sich vollständig darauf verlassen, dass immer die richtigen Schlüsselwerte in der erwarteten Spalte zu sehen sind. Beachten Sie schließlich, dass die AWS CLI Ausgaben `None` als Wert für Schlüssel ausgegeben werden, die nicht existieren.

```
$ aws iam list-users --output text --query 'Users[*].
[UserName,Arn,CreateDate,PasswordLastUsed,UserId]'
```

```
Admin          arn:aws:iam::123456789012:user/Admin
2014-10-16T16:03:09+00:00  2016-06-03T18:37:29+00:00  AIDA111111111111EXAMPLE
backup-user    arn:aws:iam::123456789012:user/backup-user
2019-09-17T19:30:40+00:00  None                          AIDA222222222222EXAMPLE
cli-user       arn:aws:iam::123456789012:user/cli-backup
2019-09-17T19:11:39+00:00  None                          AIDA333333333333EXAMPLE
```

Das folgende Beispiel zeigt, wie Sie `grep` und `awk` mit der `text`-Ausgabe des Befehls `aws ec2 describe-instances` verwenden können. Durch den ersten Befehl erhalten Sie die Availability Zone, den aktuellen Status und die Instance-ID der einzelnen Instances als `text`-Ausgabe angezeigt. Der zweite Befehl verarbeitet diese Ausgabe so, dass nur die Instanz IDs aller laufenden Instances in der `us-west-2a` Availability Zone angezeigt wird.

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text
```

```
us-west-2a    running i-4b41a37c
us-west-2a    stopped i-a071c394
us-west-2b    stopped i-97a217a0
us-west-2a    running i-3045b007
us-west-2a    running i-6fc67758
```

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text | grep us-west-2a |
grep running | awk '{print $3}'
```

```
i-4b41a37c
i-3045b007
i-6fc67758
```

Das folgende Beispiel geht noch einen Schritt weiter und zeigt nicht nur, wie Sie die Ausgabe filtern, sondern auch, wie Sie die Ausgabe verwenden, um sich ändernde Instance-Typen für angehaltene Instances zu automatisieren.

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].[State.Name,
InstanceId]' --output text |
> grep stopped |
```

```
> awk '{print $2}' |
> while read line;
> do aws ec2 modify-instance-attribute --instance-id $line --instance-type '{"Value":
"m1.medium"}';
> done
```

Die text Ausgabe kann auch nützlich sein in PowerShell. Da die Spalten in der text Ausgabe durch Tabulatoren getrennt sind, können Sie die Ausgabe mithilfe des PowerShell Trennzeichens `t` problemlos in ein Array aufteilen. Mit dem folgenden Befehl bekommen Sie den Wert der dritten Spalte (InstanceId) angezeigt, wenn die erste Spalte (AvailabilityZone) mit der Zeichenfolge us-west-2a übereinstimmt.

```
PS C:\>aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text |
%{if ($_.split("`t")[0] -match "us-west-2a") { $_.split("`t")[2]; } }
```

```
-4b41a37c
i-a071c394
i-3045b007
i-6fc67758
```

Beachten Sie, dass das vorherige Beispiel zwar zeigt, wie der `--query` Parameter verwendet wird, um die zugrunde liegenden JSON Objekte zu analysieren und die gewünschte Spalte herauszuziehen, aber es PowerShell hat seine eigene Fähigkeit, damit umzugehen JSON, wenn die plattformübergreifende Kompatibilität kein Problem darstellt. Anstatt die Ausgabe als Text zu behandeln, wie es die meisten Befehls-Shells erfordern, PowerShell können Sie das `ConvertFrom-Json` Cmdlet verwenden, um ein hierarchisch strukturiertes Objekt zu erzeugen. Sie können dann direkt von diesem Objekt aus auf das gewünschte Element zugreifen.

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

Tip

Wenn Sie Text ausgeben und die Ausgabe mithilfe des Parameters `--query` zu einem einzelnen Feld filtern, besteht die Ausgabe aus einer einzelnen Zeile mit Tabulator-getrennten Werten. Um jeden Wert auf eine separate Zeile zu bekommen, können Sie das Ausgabefeld in Klammern setzen, wie in den folgenden Beispielen gezeigt.

Tabulatorgetrennte, einzeilige Ausgabe:


```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].GroupName"
```

```
HRDepartment    Developers    SpreadsheetUsers    LocalAdmins
```

Jeder Wert auf einer eigenen Zeile durch Setzen von [GroupName] in Klammern:

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].[GroupName]"
```

```
HRDepartment
Developers
SpreadsheetUsers
LocalAdmins
```

Tabellenausgabeformat

Das table-Format produziert lesbare Darstellungen der komplexen AWS CLI -Ausgabe in Tabellenform.

```
$ aws iam list-users --output table
```

```
-----
|
| ListUsers |
+-----+
|
||
| Users |
|+-----+-----+-----+-----+
+-----+-----+-----+-----+
||          Arn          |          CreateDate          |
| PasswordLastUsed | Path | UserId | Username ||
|+-----+-----+-----+-----+
+-----+-----+-----+-----+
|| arn:aws:iam::123456789012:user/Admin | 2014-10-16T16:03:09+00:00 |
| 2016-06-03T18:37:29+00:00 | / | AIDA111111111111EXAMPLE | Admin ||
-----
```

```

|| arn:aws:iam::123456789012:user/backup/backup-user | 2019-09-17T19:30:40+00:00 | |
| /backup/ | AIDA222222222222EXAMPLE | backup-user ||
|| arn:aws:iam::123456789012:user/cli-user | 2019-09-17T19:11:39+00:00 |
| / | AIDA333333333333EXAMPLE | cli-user ||
+-----+
+

```

Sie können die `--query`-Option mit dem `table`-Format kombinieren, um eine Gruppe von Elementen zu erhalten, die vorab aus der unformatierten Ausgabe ausgewählt wurden. Beachten Sie die Ausgabeunterschiede in Wörterbuch- und Listennotation: Im ersten Beispiel werden die Spaltennamen alphabetisch sortiert und im zweiten Beispiel werden die unbenannten Spalten nach der Definition des Benutzers geordnet. Weitere Informationen zur Option `--query` finden Sie unter [Ausgabe filtern in AWS CLI](#).

```

$ aws ec2 describe-volumes --query 'Volumes[*].
{ID:VolumeId,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}' --
output table

```

```

-----
|                               DescribeVolumes                               |
+-----+-----+-----+-----+
|  AZ      |  ID      | InstanceId | Size |
+-----+-----+-----+-----+
| us-west-2a | vol-e11a5288 | i-a071c394 | 30 |
| us-west-2a | vol-2e410a47 | i-4b41a37c | 8 |
+-----+-----+-----+-----+

```

```

$ aws ec2 describe-volumes --query 'Volumes[*].
[VolumeId,Attachments[0].InstanceId,AvailabilityZone,Size]' --output table

```

```

-----
|                               DescribeVolumes                               |
+-----+-----+-----+-----+
| vol-e11a5288 | i-a071c394 | us-west-2a | 30 |
| vol-2e410a47 | i-4b41a37c | us-west-2a | 8 |
+-----+-----+-----+-----+

```

Verwenden der Paginierungsoptionen in der AWS CLI

In diesem Thema werden die verschiedenen Möglichkeiten beschrieben, die Ausgabe der AWS CLI zu paginieren.

Serverseitige Paginierung

Bei Befehlen, die eine große Liste von Elementen zurückgeben können, bietet AWS Command Line Interface (AWS CLI) mehrere Optionen, um die Anzahl der Elemente zu steuern, die in der Ausgabe enthalten sind, wenn der API Dienst die Liste AWS CLI auffüllt.

Es gibt die folgenden Optionen:

- [So verwenden Sie den `--no-paginate`-Parameter](#)
- [So verwenden Sie den `--page-size`-Parameter](#)
- [So verwenden Sie den `--max-items`-Parameter](#)
- [So verwenden Sie den `--starting-token`-Parameter](#)

Standardmäßig AWS CLI verwendet der eine Seitengröße, die vom jeweiligen Dienst bestimmt wird, und ruft alle verfügbaren Elemente ab. Amazon S3 hat beispielsweise eine Standardseitengröße von 1 000. Wenn Sie `aws s3api list-objects` auf einem Amazon-S3-Bucket ausführen, der 3 500 Objekte enthält, ruft die AWS CLI Amazon S3 automatisch vier Mal auf, verarbeitet die servicespezifische Paginierungslogik für Sie im Hintergrund und gibt alle 3 500 Objekte in der endgültigen Ausgabe zurück.

So verwenden Sie den `--no-paginate`-Parameter

Die `--no-paginate`-Option deaktiviert folgende Paginierungs-Token auf der Client-Seite. Wenn Sie einen Befehl verwenden, führt der standardmäßig AWS CLI automatisch mehrere Aufrufe durch, um alle möglichen Ergebnisse zurückzugeben und eine Seitennummerierung zu erstellen. Ein Aufruf für jede Seite. Beim Deaktivieren der Paginierung wird die erste Seite mit Befehlsergebnissen AWS CLI nur einmal aufgerufen.

Wenn Sie beispielsweise einen `aws s3api list-objects` Amazon S3-Bucket verwenden, der 3.500 Objekte enthält, ruft der Amazon S3 AWS CLI nur zum ersten Mal auf und gibt in der endgültigen Ausgabe nur die ersten 1.000 Objekte zurück.

```
$ aws s3api list-objects \
```

```
--bucket amzn-s3-demo-bucket \  
--no-paginate  
{  
  "Contents": [  
  ...
```

So verwenden Sie den `--page-size`-Parameter

Wenn beim Ausführen von Listenbefehlen für eine große Anzahl von Ressourcen Probleme auftreten, ist die Standardseitengröße möglicherweise zu hoch. Dies kann dazu führen, dass Aufrufe von AWS Diensten die maximal zulässige Zeit überschreiten und zu einem „Timeout“-Fehler führen. Sie können die `--page-size` Option verwenden, um anzugeben, dass bei jedem Aufruf des AWS Dienstes eine geringere Anzahl von Elementen AWS CLI angefordert wird. Der ruft AWS CLI immer noch die vollständige Liste ab, führt jedoch eine größere Anzahl von API Serviceaufrufen im Hintergrund durch und ruft bei jedem Aufruf eine geringere Anzahl von Elementen ab. Dadurch ist es wahrscheinlicher, dass Aufrufe nicht zu einem Zeitüberschreitungsfehler führen. Eine Änderung der Seitengröße wirkt sich nicht auf die Ausgabe aus, sondern nur auf die Anzahl der API Aufrufe, die zur Generierung der Ausgabe erforderlich sind.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --page-size 100  
{  
  "Contents": [  
  ...
```

So verwenden Sie den `--max-items`-Parameter

Verwenden Sie die `--max-items` Option, um weniger Elemente gleichzeitig in die AWS CLI Ausgabe aufzunehmen. Der verarbeitet die Seitennummerierung mit dem Service AWS CLI weiterhin wie zuvor beschrieben, druckt aber nur die Anzahl der Elemente gleichzeitig aus, die Sie angeben.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --max-items 100  
{  
  "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxZDQ=",  
  "Contents": [  
  ...
```

So verwenden Sie den `--starting-token`-Parameter

Wenn die Anzahl der Output (`--max-items`)-Elemente geringer ist als die Gesamtzahl der von den zugrunde liegenden API Aufrufen zurückgegebenen Elemente, enthält die Ausgabe eine, `NextToken` die Sie an einen nachfolgenden Befehl übergeben können, um die nächste Gruppe von Elementen abzurufen. Im folgenden Beispiel wird gezeigt, wie Sie den Wert `NextToken` aus dem vorherigen Beispiel verwenden und die zweiten hundert Elemente abrufen.

Note

Der Parameter `--starting-token` kann nicht null oder leer sein. Wenn der vorherige Befehl keinen `NextToken`-Wert zurückgibt, können keine weiteren Elemente zurückgegeben werden und Sie müssen den Befehl nicht erneut aufrufen.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --max-items 100 \  
  --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== \  
{  
  "Contents": [  
  ...
```

Der angegebene AWS Dienst gibt möglicherweise nicht bei jedem Aufruf Elemente in derselben Reihenfolge zurück. Wenn Sie verschiedene Werte für `--page-size` und `--max-items` angeben, können Sie unerwartete Ergebnisse mit fehlenden oder doppelten Elementen bekommen. Um dies zu verhindern, verwenden Sie dieselbe Zahl für `--page-size` und `--max-items`, um die Paginierung der AWS CLI mit der Paginierung der zugrunde liegenden Services zu synchronisieren. Sie können auch die gesamte Liste abrufen und erforderliche Paginierungsvorgänge lokal durchführen.

Ausgabe filtern in AWS CLI

The AWS Command Line Interface (AWS CLI) verfügt sowohl über serverseitige als auch clientseitige Filterung, die Sie einzeln oder zusammen verwenden können, um Ihre Ausgabe zu filtern. AWS CLI Die serverseitige Filterung wird zuerst verarbeitet und gibt Ihre Ausgabe für die clientseitige Filterung zurück.

- Die serverseitige Filterung wird von der unterstützten API, und Sie implementieren sie normalerweise mit einem Parameter. `--filter` Der Dienst gibt nur übereinstimmende Ergebnisse zurück, wodurch die HTTP Antwortzeiten für große Datenmengen verkürzt werden können.
- Die clientseitige Filterung wird vom AWS CLI Client mithilfe des Parameters `--query` unterstützt. Dieser Parameter verfügt über Funktionen, die die serverseitige Filterung möglicherweise nicht aufweist.

Themen

- [Serverseitige Filterung](#)
- [Clientseitige Filterung](#)
- [Serverseitige und clientseitige Filterung kombinieren](#)
- [Weitere Ressourcen](#)

Serverseitige Filterung

Die serverseitige Filterung in AWS CLI wird vom Dienst bereitgestellt. AWS API Der AWS Dienst gibt nur die Datensätze in der HTTP Antwort zurück, die Ihrem Filter entsprechen, wodurch die HTTP Antwortzeiten für große Datensätze verkürzt werden können. Da die serverseitige Filterung durch den Dienst definiert wird, variieren die Parameternamen und Funktionen von Dienst zu Dienst. Einige allgemeine Parameternamen, die zum Filtern verwendet werden, sind:

- `--filter` wie beispielsweise [ses](#) und [ce](#).
- `--filters` wie beispielsweise [ec2](#), [autoscaling](#) und [rds](#).
- Namen, die mit dem Wort `filter` beginnen, zum Beispiel `--filter-expression` für den Befehl [aws dynamodb scan](#).

Informationen darüber, ob für einen bestimmten Befehl serverseitige Filterung und die Filterregeln gelten, finden Sie im [AWS CLI Referenzhandbuch AWS CLI](#)

Clientseitige Filterung

Der AWS CLI bietet integrierte JSON clientseitige Filterfunktionen mit dem Parameter `--query`. Der Parameter `--query` ist ein leistungsstarkes Werkzeug, mit dem Sie den Inhalt und den Stil Ihrer Ausgabe anpassen können. Der `--query` Parameter verwendet die HTTP Antwort, die vom Server zurückgesendet wird, und filtert die Ergebnisse, bevor sie angezeigt werden. Da die gesamte

HTTP Antwort vor dem Filtern an den Client gesendet wird, kann die clientseitige Filterung bei großen Datensätzen langsamer sein als die serverseitige Filterung.

Beim Abfragen wird [JMESPathSyntax](#) verwendet, um Ausdrücke zum Filtern Ihrer Ausgabe zu erstellen. Informationen zum Erlernen der JMESPath Syntax finden Sie im [Tutorial](#) auf der JMESPathWebsite.

Important

Der von Ihnen angegebene Ausgabebetyp ändert die Funktionsweise der Option `--query`:

- Wenn Sie angeben `--output text`, wird die Ausgabe paginiert, bevor der `--query` Filter angewendet wird, und die Abfrage wird auf jeder Seite der Ausgabe einmal AWS CLI ausgeführt. Aus diesem Grund enthält die Abfrage das erste passende Element auf jeder Seite, was zu unerwarteten zusätzlichen Ausgaben führen kann. Um die Ausgabe zusätzlich zu filtern, können Sie andere Befehlszeilentools wie `head` oder `tail` verwenden.
- Wenn Sie `--output json` angeben, wird die Ausgabe vollständig als einzelne, native Struktur verarbeitet, bevor der `--query`-Filter angewendet wird. Der AWS CLI führt die Abfrage nur einmal für die gesamte Struktur aus, wodurch ein gefiltertes Ergebnis erzeugt wird, das dann ausgegeben wird.

Clientseitiges Filtern von Themen

- [Bevor Sie beginnen](#)
- [IDs](#)
- [Auswählen aus einer Liste](#)
- [Filtern verschachtelter Daten](#)
- [Abflachen der Ergebnisse](#)
- [Filtern nach bestimmten Werten](#)
- [Weiterleitungsausdruck](#)
- [Filtern nach mehreren ID-Werten](#)
- [Hinzufügen von Beschriftungen zu ID-Werten](#)
- [Funktionen](#)
- [Fortschrittliche --query-Beispiele](#)

Bevor Sie beginnen

Wenn Sie Filterausdrücke verwenden, die in diesen Beispielen verwendet werden, müssen Sie die richtigen Anführungsregeln für Ihre Terminalshell verwenden. Weitere Informationen finden Sie unter [the section called “Anführungszeichen mit Zeichenfolgen”](#).

Die folgende JSON Ausgabe zeigt ein Beispiel dafür, was der `--query` Parameter bewirken kann. Die Ausgabe beschreibt drei EBS Amazon-Volumes, die an separate EC2 Amazon-Instances angehängt sind.

Beispielausgabe

```
$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-17T00:55:03.000Z",
          "InstanceId": "i-a071c394",
          "VolumeId": "vol-e11a5288",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-e11a5288",
      "State": "in-use",
      "SnapshotId": "snap-f23ec1c8",
      "CreateTime": "2013-09-17T00:55:03.000Z",
      "Size": 30
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-18T20:26:16.000Z",
          "InstanceId": "i-4b41a37c",
          "VolumeId": "vol-2e410a47",
          "State": "attached",
          "DeleteOnTermination": true,
```



```

    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-2e410a47",
"State": "in-use",
"SnapshotId": "snap-708e8348",
"CreateTime": "2013-09-18T20:26:15.000Z",
"Size": 8
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-a1b3c7nd",
  "State": "in-use",
  "SnapshotId": "snap-234087fb",
  "CreateTime": "2020-11-20T19:54:05.000Z",
  "Size": 15
}
]
}

```

IDs

IDs sind die Beschriftungen für Ausgabewerte. Beim Erstellen von Filtern verwenden Sie IDs, um die Abfrageergebnisse einzugrenzen. Im folgenden Ausgabebeispiel werden alle IDs wie `Volume`, `AvailabilityZone` und `AttachTime` hervorgehoben.

```

$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",

```

```
"Attachments": [
  {
    "AttachTime": "2013-09-17T00:55:03.000Z",
    "InstanceId": "i-a071c394",
    "VolumeId": "vol-e11a5288",
    "State": "attached",
    "DeleteOnTermination": true,
    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-2e410a47",
  "State": "in-use",
  "SnapshotId": "snap-708e8348",
  "CreateTime": "2013-09-18T20:26:15.000Z",
  "Size": 8
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
```

```

        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
    }
],
"VolumeType": "standard",
"VolumeId": "vol-a1b3c7nd",
"State": "in-use",
"SnapshotId": "snap-234087fb",
"CreateTime": "2020-11-20T19:54:05.000Z",
"Size": 15
}
]
}

```

Weitere Informationen finden Sie auf der JMESPathWebsite unter [Identifikatoren](#).

Auswählen aus einer Liste

Eine Liste oder ein Array ist ein Bezeichner, auf den eine eckige Klammer „[“ folgt, wie Volumes und Attachments in [the section called “Bevor Sie beginnen”](#).

Syntax

```
<listName>[ ]
```

Um durch die gesamte Ausgabe eines Arrays zu filtern, können Sie die Platzhalternotation verwenden. [Platzhalterausdrücke](#) sind Ausdrücke, die verwendet werden, um Elemente unter Verwendung der *-Notation zurückzugeben.

Im folgenden Beispiel werden alle Volumes-Inhalte abgefragt.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,

```

```

    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-a1b3c7nd",
  "State": "in-use",
  "SnapshotId": "snap-234087fb",
  "CreateTime": "2020-11-20T19:54:05.000Z",
  "Size": 15
}
]

```

Um ein bestimmtes Volume im Array nach Index anzuzeigen, rufen Sie den Array-Index auf. Das erste Element im Array `Volumes` hat beispielsweise einen Index von 0, was zur Abfrage `Volumes[0]` führt. Weitere Informationen zu Array-Indizes finden Sie auf der Website unter [Indexausdrücke](#). JMESPath

```

$ aws ec2 describe-volumes \
  --query 'Volumes[0]'
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",

```

```
    "InstanceId": "i-a071c394",
    "VolumeId": "vol-e11a5288",
    "State": "attached",
    "DeleteOnTermination": true,
    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
}
```

Um einen bestimmten Bereich von Volumes nach Index anzuzeigen, verwenden Sie `slice` mit der folgenden Syntax, wobei Start der Start-Array-Index ist, Stopp der Index ist, bei dem der Filter die Verarbeitung stoppt und Schritt das Sprungintervall ist.

Syntax

```
<arrayName>[<start>:<stop>:<step>]
```

Wenn einer dieser Werte aus dem Slice-Ausdruck weggelassen wird, verwenden sie die folgenden Standardwerte:

- Start – Der erste Index in der Liste, 0.
- Stopp – Der letzte Index in der Liste.
- Schritt – Kein Überspringen von Schritten, wobei der Wert 1 ist.

Um nur die ersten beiden Volumes zurückzugeben, verwenden Sie einen Startwert von 0, einen Stoppwert von 2 und einen Schrittwert von 1, wie im folgenden Beispiel gezeigt.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[0:2:1]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
```

```

    "InstanceId": "i-a071c394",
    "VolumeId": "vol-e11a5288",
    "State": "attached",
    "DeleteOnTermination": true,
    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-2e410a47",
  "State": "in-use",
  "SnapshotId": "snap-708e8348",
  "CreateTime": "2013-09-18T20:26:15.000Z",
  "Size": 8
}
]

```

Da dieses Beispiel Standardwerte enthält, können Sie das Slice von `Volumes[0:2:1]` auf `Volumes[:2]` kürzen.

Im folgenden Beispiel werden Standardwerte weggelassen und alle zwei Volumes im gesamten Array zurückgegeben.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[:2]'

```

```
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  }
]
```

Schritte können negative Zahlen auch verwenden, um in umgekehrter Reihenfolge eines Arrays zu filtern, wie im folgenden Beispiel gezeigt.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[::-2]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  }
]
```


Weitere Informationen finden Sie unter [Slices](#) auf der JMESPathWebsite.

Filtern verschachtelter Daten

Um die Filterung des `Volume[*]` für verschachtelte Werte einzuschränken, verwenden Sie Unterausdrücke, indem Sie einen Punkt und Ihre Filterkriterien anhängen.

Syntax

```
<expression>.<expression>
```

Im folgenden Beispiel werden alle Attachments-Informationen für alle Volumes angezeigt.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments'
[
  [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
```

```
    "Device": "/dev/sda1"
  }
]
]
```

Um weiter in die verschachtelten Werte zu filtern, hängen Sie den Ausdruck für jede verschachtelte ID an. Das folgende Beispiel listet State für alle Volumes auf.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

Abflachen der Ergebnisse

Weitere Informationen finden Sie [SubExpressions](#) auf der JMESPathWebsite.

Sie können die Ergebnisse für `Volumes[*].Attachments[*].State` reduzieren, indem Sie die Platzhalternotation entfernen, die zur Abfrage `Volumes[*].Attachments[].State` führt. Abflachung ist oft nützlich, um die Lesbarkeit der Ergebnisse zu verbessern.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]
```

Weitere Informationen finden Sie unter [Flatten](#) auf der JMESPathWebsite.

Filtern nach bestimmten Werten

Um nach bestimmten Werten in einer Liste zu filtern, verwenden Sie einen Filterausdruck, wie in der folgenden Syntax dargestellt.

Syntax

```
? <expression> <comparator> <expression>]
```

Ausdruckscomparatoren umfassen ==, !=, <, <=, > und >=. Das folgende Beispiel filtert für die VolumeIds für alle Volumes in einem AttachedState.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'  
[  
  [  
    "vol-e11a5288"  
  ],  
  [  
    "vol-2e410a47"  
  ],  
  [  
    "vol-a1b3c7nd"  
  ]  
]
```

Dies kann dann abgeflacht werden, was zu folgendem Beispiel führt.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'  
[  
  "vol-e11a5288",  
  "vol-2e410a47",  
  "vol-a1b3c7nd"  
]
```

Das folgende Beispiel filtert für die VolumeIds aller Volumes, die eine Größe kleiner als 20 haben.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[?Size < `20`].VolumeId'  
[  
  "vol-2e410a47",
```

```
"vol-a1b3c7nd"  
]
```

Weitere Informationen finden Sie auf der JMESPathWebsite unter [Filterausdrücke](#).

Weiterleitungsausdruck

Sie können Ergebnisse eines Filters über die Pipeline an eine neue Liste übergeben und das Ergebnis anschließend mithilfe der folgenden Syntax mit einem anderen Ausdruck filtern:

Syntax

```
<expression> | <expression>]
```

Das folgende Beispiel nimmt die Filterergebnisse des Ausdrucks `Volumes[*].Attachments[].InstanceId` und gibt das erste Ergebnis im Array aus.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId | [0]'  
"i-a071c394"
```

In diesem Beispiel wird zuerst das Array aus dem folgenden Ausdruck erstellt.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId'  
"i-a071c394",  
"i-4b41a37c",  
"i-1jd73kv8"
```

Und dann gibt es das erste Element in diesem Array zurück.

```
"i-a071c394"
```

Weitere Informationen finden Sie unter [Pipe Expressions](#) auf der JMESPathWebsite.

Filtern nach mehreren ID-Werten

Um nach mehreren IDs zu filtern, verwenden Sie eine Multi-Auswahl-Liste mithilfe der folgenden Syntax:

Syntax

```
<listName>[].[<expression>, <expression>]
```

Im folgenden Beispiel werden `VolumeId` und `VolumeType` in der Liste `Volumes` gefiltert, was zu folgendem Ausdruck führt.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[][VolumeId, VolumeType]'  
[  
  [  
    "vol-e11a5288",  
    "standard"  
  ],  
  [  
    "vol-2e410a47",  
    "standard"  
  ],  
  [  
    "vol-a1b3c7nd",  
    "standard"  
  ]  
]
```

Um verschachtelte Daten zur Liste hinzuzufügen, fügen Sie eine weitere Multi-Auswahl-Liste hinzu. Das folgende Beispiel erweitert das vorherige Beispiel, indem auch nach `InstanceId` und `State` in der verschachtelten Liste `Attachments` gefiltert wird. Daraus ergibt sich der folgende Ausdruck.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[][VolumeId, VolumeType, Attachments[][InstanceId, State]]'  
[  
  [  
    "vol-e11a5288",  
    "standard",  
    [  
      [  
        "i-a071c394",  
        "attached"  
      ]  
    ]  
  ],  
  [  
    "vol-2e410a47",  
    "standard",  
    []  
  ]  
]
```

```
[
  [
    "i-4b41a37c",
    "attached"
  ]
],
[
  "vol-a1b3c7nd",
  "standard",
  [
    [
      "i-1jd73kv8",
      "attached"
    ]
  ]
]
]
```

Um besser lesbar zu sein, flachen Sie den Ausdruck wie im folgenden Beispiel gezeigt ab.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[][VolumeId, VolumeType, Attachments[][InstanceId, State]][[]]'
```

```
[
  "vol-e11a5288",
  "standard",
  [
    "i-a071c394",
    "attached"
  ],
  "vol-2e410a47",
  "standard",
  [
    "i-4b41a37c",
    "attached"
  ],
  "vol-a1b3c7nd",
  "standard",
  [
    "i-1jd73kv8",
    "attached"
  ]
]
```

Weitere Informationen finden Sie auf der JMESPathWebsite unter [Multiselect-Liste](#).

Hinzufügen von Beschriftungen zu ID-Werten

Um diese Ausgabe leichter lesbar zu machen, verwenden Sie einen Multi-Auswahl-Hash mit der folgenden Syntax.

Syntax

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

Ihre ID-Bezeichnung muss nicht mit dem Namen der ID übereinstimmen. Im folgenden Beispiel wird die Beschriftung `VolumeType` für `VolumeType`-Werte genutzt.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[].{VolumeType: VolumeType}'  
[  
  {  
    "VolumeType": "standard",  
  },  
  {  
    "VolumeType": "standard",  
  },  
  {  
    "VolumeType": "standard",  
  }  
]
```

Der Einfachheit halber behält das folgende Beispiel die Bezeichnernamen für jede Markierung bei und zeigt die `VolumeId`, `VolumeType`, `InstanceId` und `State` für alle `Volumes` an:

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[].{VolumeId: VolumeId, VolumeType: VolumeType, InstanceId:  
  Attachments[0].InstanceId, State: Attachments[0].State}'  
[  
  {  
    "VolumeId": "vol-e11a5288",  
    "VolumeType": "standard",  
    "InstanceId": "i-a071c394",  
    "State": "attached"  
  },  
  {
```

```

    "VolumeId": "vol-2e410a47",
    "VolumeType": "standard",
    "InstanceId": "i-4b41a37c",
    "State": "attached"
  },
  {
    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
  }
]

```

Weitere Informationen finden Sie unter [Multiselect-Hash](#) auf der JMESPath Website.

Funktionen

Die JMESPath Syntax enthält viele Funktionen, die Sie für Ihre Abfragen verwenden können. Informationen zu JMESPath Funktionen finden Sie unter [Integrierte Funktionen](#) auf der JMESPathWebsite.

Um zu demonstrieren, wie Sie eine Funktion in Ihre Abfragen integrieren können, verwendet das folgende Beispiel die `sort_by`-Funktion. Die `sort_by`-Funktion sortiert ein Array mit einem Ausdruck als Sortierschlüssel mit der folgenden Syntax:

Syntax

```
sort_by(<listName>, <sort expression>)[].<expression>
```

Im folgenden Beispiel wird das vorherige [Multi-Auswahl-Hash-Beispiel](#) verwendet und die Ausgabe nach VolumeId sortiert.

```

$ aws ec2 describe-volumes \
  --query 'sort_by(Volumes, &VolumeId)[].{VolumeId: VolumeId, VolumeType: VolumeType,
InstanceId: Attachments[0].InstanceId, State: Attachments[0].State}'
[
  {
    "VolumeId": "vol-2e410a47",
    "VolumeType": "standard",
    "InstanceId": "i-4b41a37c",
    "State": "attached"
  },
  {

```



```

    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
  },
  {
    "VolumeId": "vol-e11a5288",
    "VolumeType": "standard",
    "InstanceId": "i-a071c394",
    "State": "attached"
  }
]

```

Weitere Informationen finden Sie unter [sort_by](#) auf der JMESPath Website.

Fortschrittliche **--query**-Beispiele

So extrahieren Sie Informationen aus einem bestimmten Element

Das folgende Beispiel verwendet den `--query`-Parameter, um ein bestimmtes Element in einer Liste zu suchen, und extrahiert anschließend Informationen aus diesem Element. Das Beispiel listet alle `AvailabilityZones` im Zusammenhang mit dem angegebenen Service-Endpunkt auf. Es extrahiert das Element aus der `ServiceDetails`-Liste mit dem angegebenen `ServiceName` und gibt dann das Feld `AvailabilityZones` aus dem ausgewählten Element aus.

```

$ aws --region us-east-1 ec2 describe-vpc-endpoint-services \
  --query 'ServiceDetails[?ServiceName==`com.amazonaws.us-
east-1.ecs`].AvailabilityZones'
[
  [
    "us-east-1a",
    "us-east-1b",
    "us-east-1c",
    "us-east-1d",
    "us-east-1e",
    "us-east-1f"
  ]
]

```

So zeigen Sie Snapshots nach dem angegebenen Erstellungsdatum an

Das folgende Beispiel zeigt, wie Sie alle Ihre Snapshots auflisten können, die nach einem bestimmten Datum erstellt wurden, einschließlich nur einiger der verfügbaren Felder in der Ausgabe.

```
$ aws ec2 describe-snapshots --owner self \
  --output json \
  --query 'Snapshots[?StartTime>=`2018-02-07`].
{Id:SnapshotId,VID:VolumeId,Size:VolumeSize}'
[
  {
    "id": "snap-0effb42b7a1b2c3d4",
    "vid": "vol-0be9bb0bf12345678",
    "Size": 8
  }
]
```

Um die neuesten zu zeigen AMIs

Das folgende Beispiel listet die fünf neuesten Amazon Machine Images (AMIs) auf, die Sie erstellt haben, sortiert vom neuesten zum ältesten.

```
$ aws ec2 describe-images \
  --owners self \
  --query 'reverse(sort_by(Images,&CreationDate))[:5].{id:ImageId,date:CreationDate}'
[
  {
    "id": "ami-0a1b2c3d4e5f60001",
    "date": "2018-11-28T17:16:38.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60002",
    "date": "2018-09-15T13:51:22.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60003",
    "date": "2018-08-19T10:22:45.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60004",
    "date": "2018-05-03T12:04:02.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60005",
    "date": "2017-12-13T17:16:38.000Z"
  }
]
```

So zeigen Sie ungesunde Auto-Scaling-Instances an

Das folgende Beispiel zeigt nur die InstanceId für alle fehlerhaften Instances in der angegebenen Auto-Scaling-Gruppe an.

```
$ aws autoscaling describe-auto-scaling-groups \  
  --auto-scaling-group-name My-AutoScaling-Group-Name \  
  --output text \  
  --query 'AutoScalingGroups[*].Instances[?HealthStatus=='`Unhealthy`'].InstanceId'
```

So schließen Sie Volumes mit dem angegebenen Tag aus

Im folgenden Beispiel werden alle Instances ohne test-Tag beschrieben. Solange dem Volume neben test ein weiteres Tag angehängt ist, wird das Volume dennoch in den Ergebnissen zurückgegeben.

Der folgende Ausdruck, um alle Tags mit dem test Tag in einem Array zurückzugeben. Alle Tags, die nicht das Tag test sind, enthalten einen null-Wert.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

So schließen Sie Volumes mit dem angegebenen Tag aus

Im folgenden Beispiel werden alle Instances ohne test-Tag beschrieben. Die Verwendung eines einfachen ?Value != `test`-Ausdrucks funktioniert nicht, um ein Volume auszuschließen, da Volumes mehrere Tags haben können. Solange dem Volume neben test ein weiteres Tag angehängt ist, wird das Volume dennoch in den Ergebnissen zurückgegeben.

Um alle Volumes mit dem Tag test auszuschließen, beginnen Sie mit dem folgenden Ausdruck, um alle Tags mit dem Tag test in einem Array zurückzugeben. Alle Tags, die nicht das Tag test sind, enthalten einen null-Wert.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

Dann filtern Sie alle positiven test-Ergebnisse mit der not_null-Funktion.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)]'
```

Leiten Sie die Ergebnisse weiter, um die Ergebnisse abzuflachen, was zu der folgenden Abfrage führt.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)] | []'
```

Serverseitige und clientseitige Filterung kombinieren

Sie können serverseitige und clientseitige Filterung zusammen verwenden. Zuerst wird die serverseitige Filterung abgeschlossen, die die Daten an den Client sendet, die der Parameter `--query` dann filtert. Wenn Sie große Datenmengen verwenden, können Sie zunächst die serverseitige Filterung verwenden, um die Datenmenge zu verringern, die bei jedem AWS CLI Aufruf an den Client gesendet wird, und gleichzeitig die leistungsstarken Anpassungsmöglichkeiten beibehalten, die die clientseitige Filterung bietet.

Das folgende Beispiel listet EC2 Amazon-Volumes auf, die sowohl serverseitige als auch clientseitige Filterung verwenden. Der Service filtert eine Liste aller angehängten Volumes in der `us-west-2a` Availability Zone. Der Parameter `--query` begrenzt die Ausgabe weiter auf die Volumes mit einem Wert `Size`, der größer als 50 ist, und zeigt nur die angegebenen Felder mit benutzerdefinierten Namen an.

```
$ aws ec2 describe-volumes \  
  --filters "Name=availability-zone,Values=us-west-2a" "Name=status,Values=attached" \  
  \  
  --query 'Volumes[?Size > `50`].{Id:VolumeId,Size:Size,Type:VolumeType}' \  
  [ \  
    { \  
      "Id": "vol-0be9bb0bf12345678", \  
      "Size": 80, \  
      "VolumeType": "gp2" \  
    } \  
  ]
```

Das folgende Beispiel ruft eine Liste von Images ab, die mehrere Kriterien erfüllen. Anschließend wird mit dem Parameter `--query` die Ausgabe nach `CreationDate` sortiert und nur das neueste ausgewählt. Schließlich wird die `ImageId` dieses ein Image angezeigt.

```
$ aws ec2 describe-images \  
  --owners amazon \  
  \  
  --query 'Images[?CreationDate == `2017-01-01T00:00:00.000Z`].ImageId'
```

```
--filters "Name=name,Values=amzn*gp2" "Name=virtualization-type,Values=hvm"
"Name=root-device-type,Values=ebs" \
--query "sort_by(Images, &CreationDate)[-1].ImageId" \
--output text
ami-00ced3122871a4921
```

Das folgende Beispiel zeigt die Anzahl der verfügbaren Volumes, die mehr als 1000 sind, an, IOPS indem gezählt wird `length`, wie viele sich in einer Liste befinden.

```
$ aws ec2 describe-volumes \
  --filters "Name=status,Values=available" \
  --query 'length(Volumes[?Iops > `1000`])'
3
```

Weitere Ressourcen

JMESPathTerminal

JMESPathTerminal ist ein interaktiver Terminalbefehl zum Experimentieren mit JMESPath Ausdrücken, die für die clientseitige Filterung verwendet werden. Mit dem Befehl `jpterm` zeigt das Terminal während der Eingabe sofort Abfrageergebnisse an. Sie können die AWS CLI Ausgabe direkt an das Terminal weiterleiten, was erweiterte Abfrageexperimente ermöglicht.

Das folgende Beispiel leitet die `aws ec2 describe-volumes` Ausgabe direkt an das Terminal weiter JMESPath.

```
$ aws ec2 describe-volumes | jpterm
```

Weitere Informationen zu JMESPath Terminal und Installationsanweisungen finden Sie unter [JMESPathTerminal](#) on GitHub.

JQ-Serviceprogramm

Das `jq`-Serviceprogramm bietet Ihnen eine Möglichkeit, Ihre Ausgabe auf der Client-Seite in ein gewünschtes Ausgabeformat umzuwandeln. Weitere Informationen `jq` und Installationsanweisungen finden Sie unter [jq](#) on GitHub.

Befehlszeilen-Rückgabecodes in der AWS CLI

Der Rückgabecode ist normalerweise ein versteckter Code, der nach der Ausführung eines AWS Command Line Interface (AWS CLI) -Befehls gesendet wird, der den Status des Befehls beschreibt. Sie können den `echo` Befehl verwenden, um den vom letzten AWS CLI Befehl gesendeten Code anzuzeigen und anhand dieser Codes festzustellen, ob ein Befehl erfolgreich war oder ob er fehlgeschlagen ist und warum ein Befehl möglicherweise einen Fehler aufweist. Zusätzlich zu den Rückgabecodes können Sie weitere Details zu einem Fehler anzeigen, indem Sie Ihre Befehle mit dem Schalter `--debug` ausführen. Durch diesen Schalter wird dann ein ausführlicher Bericht der AWS CLI -Schritte zur Verarbeitung des Befehls und ihres jeweiligen Ergebnisses erstellt.

Um den Rückgabecode eines AWS CLI Befehls zu ermitteln, führen Sie unmittelbar nach der Ausführung des CLI Befehls einen der folgenden Befehle aus.

Linux and macOS

```
$ echo $?  
0
```

Windows PowerShell

```
PS> echo $lastexitcode  
0
```

Windows Command Prompt

```
C:\> echo %errorlevel%  
0
```

Im Folgenden sind die Rückgabecodewerte aufgeführt, die am Ende der Ausführung eines AWS Command Line Interface (AWS CLI) -Befehls zurückgegeben werden können.

Code	Bedeutung
0	Der Dienst antwortete mit dem HTTP Antwortstatuscode 200, der angibt, dass keine Fehler durch den AWS CLI AWS Dienst generiert wurden, an den die Anforderung gesendet wurde.

Code	Bedeutung
1	Ein oder mehrere Amazon-S3-Übertragungsvorgänge sind fehlgeschlagen. Auf S3-Befehle beschränkt.
2	Die Bedeutung dieses Rückgabecodes hängt von dem Befehl ab: <ul style="list-style-type: none">• Gilt für alle AWS CLI Befehle — der eingegebene Befehl konnte nicht analysiert werden. Parsing-Fehler können auf fehlende erforderliche Unterbefehle oder Argumente oder die Verwendung unbekannter Befehle oder Argumente zurückzuführen sein, sind jedoch nicht hierauf beschränkt.• Begrenzt auf S3-Befehle – Eine oder mehrere für die Übertragung markierte Dateien wurden während der Übertragung übersprungen. Alle übrigen für die Übertragung markierten Dateien wurden jedoch erfolgreich übertragen. Zu den Dateien, die während des Übertragungsvorgangs übersprungen werden, gehören: Dateien, die nicht existieren; Dateien, bei denen es sich um spezielle Zeichengeräte handelt, spezielle Geräte blockieren, FIFO Warteschlangen oder Sockets; und Dateien, für die der Benutzer keine Leseberechtigung hat.
130	Der Befehl wurde durch einen unterbrochen. SIGINT Dies ist das von Ihnen gesendete Signal, um einen Befehl mit <code>Ctrl+C</code> abzubrechen.
255	Der Befehl ist fehlgeschlagen. Es wurden Fehler durch den Dienst AWS CLI oder durch den AWS Dienst verursacht, an den die Anfrage gesendet wurde.

Aliase erstellen und verwenden in AWS CLI

Aliase sind Abkürzungen, die Sie in AWS Command Line Interface (AWS CLI) erstellen können, um häufig verwendete Befehle oder Skripts zu kürzen. Sie erstellen Aliase in der `alias`-Datei in Ihrem Konfigurationsordner.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen der Aliasdatei](#)
- [Schritt 2: Erstellen eines Alias](#)
- [Schritt 3: Aufruf eines Alias](#)

- [Beispiele für das Alias-Repository](#)
- [Ressourcen](#)

Voraussetzungen

Um Alias-Befehle zu verwenden, führen Sie die folgenden Schritte aus:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Verwenden Sie eine AWS CLI Mindestversion von 1.11.24 oder 2.0.0.
- (Optional) Um AWS CLI Alias-Bash-Skripten zu verwenden, müssen Sie ein Bash-kompatibles Terminal verwenden.

Schritt 1: Erstellen der Aliasdatei

Um die `alias` Datei zu erstellen, können Sie Ihre Dateinavigation und einen Texteditor verwenden oder Ihr bevorzugtes Terminal verwenden, indem Sie das Verfahren verwenden. step-by-step Verwenden Sie den folgenden Befehlsblock, um schnell Ihre Aliasdatei zu erstellen.

Linux and macOS

```
$ mkdir -p ~/.aws/cli
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> md %USERPROFILE%\aws\cli
C:\> echo [toplevel] > %USERPROFILE%\aws\cli\alias
```

So erstellen Sie die Aliasdatei

1. Erstellen Sie einen Ordner mit dem Namen `cli` in Ihrem AWS CLI Konfigurationsordner. Standardmäßig ist der Konfigurationsordner `~/.aws/` (Linux oder macOS) und `%USERPROFILE%\aws\` (Windows). Sie können dies über Ihre Dateinavigation oder mit dem folgenden Befehl erstellen.

Linux and macOS

```
$ mkdir -p ~/.aws/cli
```

Windows

```
C:\> md %USERPROFILE%\.aws\cli
```

Der resultierende Standardpfad des `cli`-Ordners ist `~/.aws/cli/` unter Linux oder macOS und `%USERPROFILE%\.aws\cli` unter Windows.

- Erstellen Sie im `cli`-Ordner eine Textdatei namens `alias` ohne Erweiterung und fügen Sie `[toplevel]` in die erste Zeile ein. Sie können diese Datei über Ihren bevorzugten Texteditor erstellen oder den folgenden Befehl verwenden.

Linux and macOS

```
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> echo [toplevel] > %USERPROFILE%\.aws\cli\alias
```

Schritt 2: Erstellen eines Alias

Sie können einen Alias mithilfe von Basisbefehlen oder Bash-Skripten erstellen.

Erstellen eines Basisbefehls-Alias

Sie können Ihren Alias erstellen, indem Sie in der `alias`-Datei, die Sie im vorherigen Schritt erstellt haben, einen Befehl mit der folgenden Syntax hinzufügen.

Syntax

```
aliasname = command [--options]
```

Das Tool *aliasname* ist das, was du deinen Alias nennst. Das Tool *command* ist der Befehl, den Sie aufrufen möchten. Er kann auch andere Aliase enthalten. Sie können Optionen oder Parameter in Ihren Alias aufnehmen oder sie hinzufügen, wenn Sie Ihren Alias aufrufen.

Im folgenden Beispiel wird mit dem Befehl `aws sts get-caller-identity` ein Alias namens `aws whoami` erstellt. Da dieser Alias einen vorhandenen AWS CLI -Befehl aufruft, können Sie den Befehl ohne `aws`-Präfix schreiben.

```
whoami = sts get-caller-identity
```

Im folgenden Beispiel wird das vorherige `whoami`-Beispiel verwendet und die `Account-Filter-` und `Textoptionen output` hinzugefügt.

```
whoami2 = sts get-caller-identity --query Account --output text
```

Erstellen eines Alias für Unterbefehle

Note

Die Aliasfunktion für Unterbefehle erfordert eine AWS CLI Mindestversion von 1.11.24 oder 2.0.0

Sie können einen Alias für Unterbefehle erstellen, indem Sie in der `alias`-Datei, die Sie im vorherigen Schritt erstellt haben, einen Befehl mit der folgenden Syntax hinzufügen.

Syntax

```
[command commandGroup]  
aliasname = command [--options]
```

Das Tool *commandGroup* ist der Befehlsnamespace, z. B. Der Befehl `aws ec2 describe-regions` befindet sich unter der Befehlsgruppe. `ec2` Das Tool *aliasname* ist das, was du deinen Alias nennst. Das Tool *command* ist der Befehl, den Sie aufrufen möchten. Er kann auch andere Aliase enthalten. Sie können Optionen oder Parameter in Ihren Alias aufnehmen oder sie hinzufügen, wenn Sie Ihren Alias aufrufen.

Im folgenden Beispiel wird mit dem Befehl [aws ec2 describe-regions](#) ein Alias namens `aws ec2 regions` erstellt. Da dieser Alias einen vorhandenen AWS CLI -Befehl unter dem Befehls-Namespace `ec2` aufruft, können Sie den Befehl ohne das Präfix `aws ec2` schreiben.

```
[command ec2]
regions = describe-regions --query Regions[].RegionName
```

Um Aliase von Befehlen außerhalb des Befehls-Namespace zu erstellen, stellen Sie dem vollständigen Befehl ein Ausrufezeichen voran. Im folgenden Beispiel wird mit dem Befehl [aws iam list-instance-profiles](#) ein Alias namens `aws ec2 instance-profiles` erstellt.

```
[command ec2]
instance-profiles = !aws iam list-instance-profiles
```

Note

Aliase verwenden nur bestehende Befehls-Namespace und Sie können keine neuen erstellen. Sie können beispielsweise keinen Alias mit dem Abschnitt `[command johnsmith]` erstellen, da der Befehls-Namespace `johnsmith` noch nicht vorhanden ist.

Erstellen eines Bash-Scripting-Alias

Warning

Um AWS CLI Alias-Bash-Skripten zu verwenden, müssen Sie ein Bash-kompatibles Terminal verwenden

Sie können einen Alias mit Bash-Skripten für erweiterte Prozesse mit der folgenden Syntax erstellen.

Syntax

```
aliasname =
  !f() {
    script content
  }; f
```

Das Tool *aliasname* ist das, was du dein Alias nennst und *script content* ist das Skript, das Sie ausführen möchten, wenn Sie den Alias aufrufen.

Im folgenden Beispiel wird `opendns` verwendet, um Ihre aktuelle IP-Adresse auszugeben. Da Sie Aliase in anderen Aliasen verwenden können, ist der folgende `myip`-Alias nützlich, um den Zugriff für Ihre IP-Adresse aus anderen Aliasen heraus zuzulassen oder zu widerrufen.

```
myip =
  !f() {
    dig +short myip.opendns.com @resolver1.opendns.com
  }; f
```

Das folgende Skriptbeispiel ruft den vorherigen `aws myip` Alias auf, um Ihre IP-Adresse für einen EC2 Amazon-Sicherheitsgruppeneingang zu autorisieren.

```
authorize-my-ip =
  !f() {
    ip=$(aws myip)
    aws ec2 authorize-security-group-ingress --group-id ${1} --cidr $ip/32 --protocol
tcp --port 22
  }; f
```

Wenn Sie Aliase aufrufen, die Bash-Skripting verwenden, werden die Variablen immer in der Reihenfolge übergeben, in der Sie sie eingegeben haben. Beim Bash-Skripting werden die Variablennamen nicht berücksichtigt, sondern nur die Reihenfolge, in der sie erscheinen. Im folgenden `textalert`-Alias-Beispiel ist die Variable für die `--message`-Option die erste und die `--phone-number`-Option ist die zweite.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

Schritt 3: Aufruf eines Alias

Verwenden Sie die folgende Syntax, um den in Ihrer `alias`-Datei erstellten Alias auszuführen. Sie können zusätzliche Optionen hinzufügen, wenn Sie Ihren Alias aufrufen.

Syntax

```
$ aws aliasname
```

Im folgenden Beispiel wird der Befehlsalias `aws whoami` verwendet.

```
$ aws  
whoami  
{  
  "UserId": "A12BCD34E5FGHI6JKLM",  
  "Account": "1234567890987",  
  "Arn": "arn:aws:iam::1234567890987:user/userName"  
}
```

Im folgenden Beispiel wird der `aws whoami`-Alias mit zusätzlichen Optionen verwendet, um nur die Account-Zahl in der `text`-Ausgabe zurückzugeben.

```
$ aws whoami --query Account --output  
text  
1234567890987
```

Im folgenden Beispiel wird der [Unterbefehlsalias](#) `aws ec2 regions` verwendet.

```
$ aws ec2  
regions  
[  
  "ap-south-1",  
  "eu-north-1",  
  "eu-west-3",  
  "eu-west-2",  
  ...
```

Aufrufen eines Alias mit Bash-Skriptvariablen

Wenn Sie Aliase aufrufen, die Bash-Skripting verwenden, werden Variablen in der Reihenfolge übergeben, in der sie eingegeben werden. Beim Bash-Skripting wird der Name der Variablen nicht berücksichtigt, sondern nur die Reihenfolge, in der sie erscheinen. Im folgenden `textalert`-Alias beispielsweise ist die Variable für die Option `--message` die erste und `--phone-number` die zweite.

```
textalert =  
!f() {
```

```
aws sns publish --message "${1}" --phone-number ${2}
}; f
```

Wenn Sie den `textalert`-Alias aufrufen, müssen Sie Variablen in der gleichen Reihenfolge übergeben, in der sie im Alias ausgeführt werden. Im folgenden Beispiel verwenden wir die Variablen `$message` und `$phone`. Die `$message`-Variable wird als `${1}` für die `--message`-Option und die `$phone`-Variable wird als `${2}` für die `--phone-number`-Option übergeben. Dies führt zu einem erfolgreichen Aufruf des `textalert`-Alias, um eine Nachricht zu senden.

```
$ aws textalert $message
$phone
{
  "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

Im folgenden Beispiel wird die Bestellung umgeschaltet, wenn Sie den Alias in `$phone` und `$message` aufrufen. Die `$phone`-Variable wird als `${1}` für die `--message`-Option und die `$message`-Variable wird als `${2}` für die `--phone-number`-Option übergeben. Da die Variablen nicht in Ordnung sind, übergibt der Alias die Variablen falsch. Dies führt zu einem Fehler, da der Inhalt von `$message` nicht den Formatierungsanforderungen für die Telefonnummer für die Option `--phone-number` entspricht.

```
$ aws textalert $phone
$message
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help

Unknown options: text
```

Beispiele für das Alias-Repository

Das [AWS CLI Alias-Repository](#) on GitHub enthält AWS CLI Alias-Beispiele, die vom AWS CLI Entwicklerteam und der Community erstellt wurden. Sie können das gesamte `alias`-Dateibeispiel verwenden oder einzelne Aliase für Ihren eigenen Gebrauch verwenden.

⚠ Warning

Durch Ausführen der Befehle in diesem Abschnitt wird Ihre vorhandene `alias`-Datei gelöscht. Um das Überschreiben der vorhandenen Aliasdatei zu vermeiden, ändern Sie den Downloadort.

So verwenden Sie Aliase aus dem Repository

1. Installieren Sie Git. Installationsanweisungen finden Sie unter [Erste Schritte – Git installieren](#) in der Git-Dokumentation.
2. Installieren Sie den `jp`-Befehl. Der `jp`-Befehl wird in im `toString`-Alias verwendet. Installationsanweisungen finden Sie unter [JMESPath\(jp\) README .md](#) auf GitHub.
3. Installieren Sie den `jq`-Befehl. Der `jq`-Befehl wird in im `toString-with-jq`-Alias verwendet. Installationsanweisungen finden Sie unter Der [JSONProzessor \(jq\)](#) auf. GitHub
4. Laden Sie die `alias`-Datei herunter, indem Sie einen der folgenden Schritte ausführen:
 - Führen Sie die folgenden Befehle aus, die aus dem Repository heruntergeladen werden und kopieren Sie die `alias`-Datei in Ihren Konfigurationsordner.

Linux and macOS

```
$ git clone https://github.com/awslabs/awscli-aliases.git
$ mkdir -p ~/.aws/cli
$ cp awscli-aliases/alias ~/.aws/cli/alias
```

Windows

```
C:\> git clone https://github.com/awslabs/awscli-aliases.git
C:\> md %USERPROFILE%\aws\cli
C:\> copy awscli-aliases\alias %USERPROFILE%\aws\cli
```

- Laden Sie es direkt aus dem Repository herunter und speichern Sie es in dem `cli` Ordner in Ihrem AWS CLI Konfigurationsordner. Standardmäßig ist der Konfigurationsordner `~/.aws/` (Linux oder macOS) und `%USERPROFILE%\aws\` (Windows).
5. Um zu überprüfen, ob die Aliase funktionieren, führen Sie den folgenden Alias aus.

```
$ aws whoami
```

Dies zeigt die gleiche Antwort wie der `aws sts get-caller-identity`-Befehl:

```
{
  "Account": "012345678901",
  "UserId": "AIUAINBADX2VEG2TC6HD6",
  "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```

Ressourcen

- Das [AWS CLI Alias-Repository](#) on GitHub enthält AWS CLI Alias-Beispiele, die vom AWS CLI Entwicklerteam und den Beiträgen der AWS CLI Community erstellt wurden.
- Die Ankündigung der Alias-Funktion von [AWS re:Invent 2016: The Effective AWS CLI User](#) am YouTube
- [aws sts get-caller-identity](#)
- [aws ec2 describe-instances](#)
- [aws sns publish](#)

Codebeispiele für die AWS CLI

Dieses Kapitel enthält eine Sammlung von Beispielen, die Ihnen zeigen, wie Sie das AWS Command Line Interface (AWS CLI) mit verwenden können AWS-Services.

Dieses Handbuch AWS CLI enthält die folgenden Arten von Beispielen:

- [Beispiele für](#) geführte Befehle — Beispiele für geführte Befehle für das AWS CLI Benutzerhandbuch zur Verwendung von AWS CLI mit einigen AWS-Services. Dabei handelt es sich häufig um detailliertere Beispiele als die Beispiele aus dem [AWS CLI Referenzhandbuch AWS CLI](#)
- [AWS CLI Befehlsbeispiele](#) — Open-Source-Befehlsbeispiele, die auch im [AWS CLI Referenzhandbuch AWS CLI](#) verfügbar sind. Befehlsbeispiele werden im [AWS CLI](#)Repository unter gehostet GitHub.
- [AWS CLI mithilfe von Bash-Scripting-Codebeispielen](#) — [Open-Source-Bash-Skripting-Beispiele](#). Bash-Skriptbeispiele finden Sie im [Code Examples Repository](#) unter [AWS](#)GitHub

Beispiel für Feedback

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Befehlsbeispiel an, indem Sie den Link Feedback geben unten auf dieser Seite oder auf der entsprechenden Befehlsseite im [AWS CLI Referenzhandbuch AWS CLI](#) verwenden.

Sie möchten einen Beitrag leisten? Fügen AWS CLI Sie Befehlsbeispiele im [AWS Codebeispiel-Repository](#) bei GitHub. Weitere Informationen zur Mitarbeit finden Sie auf den GitHub-Seiten unter [Kurzanleitung für Beiträge zu AWS CLI Codebeispielen](#).

Geführte Befehlsbeispiele für die AWS CLI

The AWS Command Line Interface (AWS CLI) ist ein Open-Source-Tool, mit dem Sie in Ihrer Befehlszeilen-Shell mit einer Vielzahl von AWS-Services Befehlen interagieren können. Dieser Abschnitt enthält angeleitete Beispiele, die zeigen, wie Sie mithilfe von AWS CLI auf einige der zugreifen können. AWS-Services Dazu gehören einige der benutzerdefinierten AWS CLI Befehle, z. B. die `aws s3` Befehle auf hoher Ebene. Diese Befehlsbeispiele veranschaulichen allgemeine Aktionen, die für einige verwendet werden, AWS-Services und bieten zusätzliche Ressourcen für weitere Informationen.

Unabhängig davon, ob Sie ein erfahrener AWS Benutzer oder Neuling sind, dienen diese geführten Beispiele als Ressource zur Optimierung Ihrer AWS Abläufe.

Eine vollständige Referenz aller verfügbaren Befehle für jeden AWS-Service-Befehl finden Sie im [AWS CLI Referenzhandbuch AWS CLI](#). Darüber hinaus können Sie die [integrierte Befehlszeilenhilfe](#) verwenden, um sich mit den Befehlen AWS-Services, Optionen und Funktionen vertraut zu machen. `aws cli`.

Weitere Befehlsbeispiele, die in diesem Abschnitt nicht verfügbar sind, finden Sie im [AWS CLI Befehlsbeispiele](#) Abschnitt. Dies sind Open-Source-Befehlsbeispiele, die auch im [AWS CLI Referenzhandbuch AWS CLI](#) verfügbar sind. Befehlsbeispiele werden im [AWS CLI Repository](#) unter gehostet GitHub.

Beispiele für Open-Source-Bash-Scripting finden Sie unter [the section called "Bash-Skript-Beispiele"](#). Bash-Skriptbeispiele finden Sie im [AWS Code Examples Repository](#) unter [GitHub](#).

Services

- [Verwendung von Amazon DynamoDB in AWS CLI](#)
- [Verwendung von Amazon EC2 in der AWS CLI](#)
- [Verwendung von Amazon S3 Glacier in der AWS CLI](#)
- [Verwendung IAM in der AWS CLI](#)
- [Verwendung von Amazon S3 in der AWS CLI](#)
- [Zugriff auf Amazon SNS in der AWS CLI](#)

Verwendung von Amazon DynamoDB in AWS CLI

Eine Einführung in Amazon DynamoDB

[Was ist Amazon DynamoDB?](#)

Die AWS Command Line Interface (AWS CLI) bietet Unterstützung für alle AWS Datenbankdienste, einschließlich Amazon DynamoDB. Sie können das AWS CLI für spontane Operationen verwenden, z. B. das Erstellen einer Tabelle. Sie können damit auch DynamoDB-Vorgänge in Hilfsprogrammskripten einbetten.

Weitere Informationen zur Verwendung von AWS CLI mit DynamoDB finden Sie [dynamodb](#) in der AWS CLI Befehlsreferenz.

Verwenden Sie den folgenden Befehl, um die AWS CLI Befehle für DynamoDB aufzulisten.

```
$ aws dynamodb help
```

Themen

- [Voraussetzungen](#)
- [Erstellen und Verwenden von DynamoDB-Tabellen](#)
- [Verwenden von DynamoDB Local](#)
- [Ressourcen](#)

Voraussetzungen

Zur Ausführung von dynamodb-Befehlen ist Folgendes erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).

Erstellen und Verwenden von DynamoDB-Tabellen

Das Befehlszeilenformat besteht aus einem DynamoDB-Befehlsnamen, gefolgt von den Parametern für diesen Befehl. Der AWS CLI unterstützt die CLI [Kurzsyntax für die Parameterwerte und die vollständige Syntax](#). JSON

Im folgenden Beispiel wird eine Tabelle mit dem Namen MusicCollection erstellt.

```
$ aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=Artist,AttributeType=S  
  AttributeName=SongTitle,AttributeType=S \  
  --key-schema AttributeName=Artist,KeyType=HASH  
  AttributeName=SongTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1
```

Sie können neue Zeilen zur Tabelle hinzufügen mithilfe von Befehlen, die denen im folgenden Beispiel ähneln. In diesen Beispielen wird eine Kombination aus Kurzsyntax und verwendet. JSON

```
$ aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item '{  
    "Artist": {"S": "No One You Know"},  
    "SongTitle": {"S": "Call Me Today"} ,  
    "AlbumTitle": {"S": "Somewhat Famous"}  
  }' \  
  --return-consumed-capacity TOTAL  
{  
  "ConsumedCapacity": {  
    "CapacityUnits": 1.0,  
    "TableName": "MusicCollection"  
  }  
}
```

```
$ aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item '{  
    "Artist": {"S": "Acme Band"},  
    "SongTitle": {"S": "Happy Day"} ,  
    "AlbumTitle": {"S": "Songs About Life"}  
  }' \  
  --return-consumed-capacity TOTAL  
{  
  "ConsumedCapacity": {  
    "CapacityUnits": 1.0,  
    "TableName": "MusicCollection"  
  }  
}
```

Es kann schwierig sein, gültige Befehle JSON in einem einzeiligen Befehl zu verfassen. Um das einfacher zu machen, AWS CLI können sie JSON Dateien lesen. Stellen Sie sich zum Beispiel den folgenden JSON Ausschnitt vor, der in einer Datei mit dem Namen `expression-attributes.json` gespeichert ist.

```
{  
  ":v1": {"S": "No One You Know"},  
  ":v2": {"S": "Call Me Today"}  
}
```

Sie können diese Datei verwenden, um eine query-Anfrage mithilfe der AWS CLI auszugeben. Im folgenden Beispiel wird der Inhalt der Datei `expression-attributes.json` für den Wert des Parameters `--expression-attribute-values` verwendet.

```
$ aws dynamodb query --table-name MusicCollection \  
  --key-condition-expression "Artist = :v1 AND SongTitle = :v2" \  
  --expression-attribute-values file://expression-attributes.json  
{  
  "Count": 1,  
  "Items": [  
    {  
      "AlbumTitle": {  
        "S": "Somewhat Famous"  
      },  
      "SongTitle": {  
        "S": "Call Me Today"  
      },  
      "Artist": {  
        "S": "No One You Know"  
      }  
    }  
  ],  
  "ScannedCount": 1,  
  "ConsumedCapacity": null  
}
```

Verwenden von DynamoDB Local

Zusätzlich zu DynamoDB können Sie den AWS CLI mit DynamoDB Local verwenden. DynamoDB Local ist eine kleine clientseitige Datenbank, die nach dem Vorbild des DynamoDB-Service funktioniert. DynamoDB Local ermöglicht es Ihnen, Anwendungen zu schreiben, die DynamoDB verwendenAPI, ohne Tabellen oder Daten im DynamoDB-Webdienst zu manipulieren. Stattdessen werden alle API Aktionen in eine lokale Datenbank umgeleitet. Dies ermöglicht Ihnen Einsparungen, die den bereitgestellten Durchsatz, die Datenspeicherung und Datenübertragungsgebühren betreffen.

Weitere Informationen zu DynamoDB Local und zur Verwendung mit dem finden Sie in den AWS CLI folgenden Abschnitten des [Amazon DynamoDB Developer Guide](#):

- [DynamoDB Local](#)
- [Verwendung der AWS CLI mit DynamoDB Local](#)

Ressourcen

AWS CLI Referenz:

- [aws dynamodb](#)
- [aws dynamodb create-table](#)
- [aws dynamodb put-item](#)
- [aws dynamodb query](#)

Service-Referenz:

- [DynamoDB Local](#) im Entwicklerhandbuch für Amazon DynamoDB
- [Verwendung mit AWS CLI DynamoDB Local](#) im Entwicklerhandbuch für Amazon DynamoDB

Verwendung von Amazon EC2 in der AWS CLI

Eine Einführung in Amazon Elastic Compute Cloud

[Einführung in Amazon EC2 — Elastic Cloud Server und Hosting mit AWS](#)

Amazon Elastic Compute Cloud (AmazonEC2) bietet hoch skalierbare und flexible virtuelle Computerumgebungen. EC2Mit Amazon können Sie virtuelle Server, sogenannte EC2 Amazon-Instances, bereitstellen und verwalten, um eine Vielzahl von Computeranforderungen zu erfüllen.

EC2Amazon-Instances sind virtuelle Maschinen, die mit verschiedenen Konfigurationen von ArbeitsspeicherCPU, Speicher und Netzwerkfunktionen angepasst werden können. Sie können aus einer Vielzahl von Instance-Typen wählen, die je nach Ihren Anwendungsanforderungen von leichten, kostengünstigen Optionen bis hin zu leistungsstarken Instances mit hoher Leistung reichen. Diese Flexibilität ermöglicht es Ihnen, Ihren Computeranforderungen gerecht zu werden, um Leistung und Kosteneffizienz zu optimieren.

Darüber hinaus EC2 bietet Amazon eine Reihe von Funktionen, mit denen Sie Ihre Rechenressourcen effektiv verwalten können. Dazu gehören die Möglichkeit, schnell neue Instances zu starten, benutzerdefinierte Maschinen-Images (AMIs) für eine schnelle Bereitstellung zu erstellen und die Rechenkapazität nach Bedarf zu erhöhen oder zu reduzieren.

Mit dem AWS Command Line Interface (AWS CLI) können Sie auf die Funktionen von Amazon EC2 zugreifen. Verwenden Sie den folgenden Befehl `aws ec2 help`, um die AWS CLI Befehle für Amazon aufzulisten.

```
aws ec2 help
```

Bevor Sie Befehle ausführen, richten Sie die Standardanmeldeinformationen ein. Weitere Informationen finden Sie unter [Konfiguration der Einstellungen für den AWS CLI](#).

Dieses Thema enthält kurze Beispiele für AWS CLI Befehle, die allgemeine Aufgaben für Amazon EC2 ausführen.

Ausführliche Beispiele für AWS CLI Befehle finden Sie unter [AWS CLI Codebeispiele im Repository](#) unter GitHub.

Themen

- [EC2Amazon-Schlüsselpaare erstellen, anzeigen und löschen in AWS CLI](#)
- [Erstellen, Konfigurieren und Löschen von EC2 Amazon-Sicherheitsgruppen in der AWS CLI](#)
- [Starten, Auflisten und Schließen von EC2 Amazon-Instances in der AWS CLI](#)
- [Ändern Sie einen EC2 Amazon-Instance-Typ mit einem Bash-Skript in AWS CLI](#)

EC2Amazon-Schlüsselpaare erstellen, anzeigen und löschen in AWS CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Ihre Schlüsselpaare für Amazon Elastic Compute Cloud (AmazonEC2) zu erstellen, anzuzeigen und zu löschen. Sie verwenden Schlüsselpaare, um eine Verbindung zu einer EC2 Amazon-Instance herzustellen.

Sie müssen Amazon das key pair zur Verfügung stellen, EC2 wenn Sie die Instance erstellen, und dieses key pair dann zur Authentifizierung verwenden, wenn Sie eine Verbindung mit der Instance herstellen.

Note

Weitere Befehlsbeispiele finden Sie im [AWS CLI AWS CLI Referenzhandbuch](#)

Themen

- [Voraussetzungen](#)

- [Erstellen eines Schlüsselpaars](#)
- [Anzeigen Ihres Schlüsselpaars](#)
- [Löschen Ihres Schlüsselpaars](#)
- [Referenzen](#)

Voraussetzungen

Zur Ausführung von ec2-Befehlen ist Folgendes erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Stellen Sie Ihre IAM Berechtigungen so ein, dass Sie EC2 Zugriff auf Amazon haben. Weitere Informationen zu IAM Berechtigungen für Amazon EC2 finden Sie in den [IAMRichtlinien für Amazon EC2](#) im EC2Amazon-Benutzerhandbuch.

Erstellen eines Schlüsselpaars

Zum Erstellen eines Schlüsselpaars verwenden Sie den Befehl `aws ec2 create-key-pair` mit den Optionen `--query` und `--output text`, um den privaten Schlüssel direkt in eine Datei weiterzureichen.

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
> MyKeyPair.pem
```

Denn PowerShell die `> file` Umleitung verwendet standardmäßig die Kodierung UTF -8, die bei einigen Clients nicht verwendet werden kann. SSH Sie müssen die Ausgabe konvertieren, indem Sie sie an den Befehl `out-file` weiterreichen und die Kodierung explizit auf `ascii` festlegen.

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
| out-file -encoding ascii -filepath MyKeyPair.pem
```

Die resultierende Datei `MyKeyPair.pem` sollte wie folgt aussehen.

```
-----BEGIN RSA PRIVATE KEY-----  
EXAMPLEKEYKCAQEAY7WZhaDsra1W3mRlQtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/  
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7i05dSrvC7dQkW2duV5QuUdE0QW  
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZwQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449i1x9X1F  
G50TCFe0zfl8dqqCP6GzbPaIjiU19xX/az0R9V+tpU0zEL+wmXnZt3/nHPQ5xvD20JH67km6SuPW
```



```
oPzev/D8V+x4+bHthfSjR9Y7DvQFjfBVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrqu
/uler7vgIn5m71N5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/ufGxbL1
mb5qwMGUnEpJaZD6QSSs3kICLwWUYUiGfc0uiSbmJoap/GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2
bahyWyJNfjLe4M86yd2YK3V2CmK+X/B0sShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/tJWSD9
81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA30zdXzMQexXVJ1TLZVEH0E7bh1Y9d801ozR
oQs/FiZNAx2iijCWyv0lpjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfq1+lIp1
YkriL0DbLXlvRAH+yHPRit2hH0jtUNZ4Axx+cpq09qbUI3+43eEy24B7G/Uh+GTfbjsXs0xQx/x
p9otyVvc7hsQ5TA5PZb+mvkJ50BEKzet9XcKw0NBVELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
ayav0elc5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URipzWV0Eihvm+xTtmaZ1Sp//1kq75XDwnU
WA8gkn603QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
gYBjb0+OZk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjlDp9PfN295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY70zd5wQewBQ4AdS1WSX4nGDtsiFxiI5sKuAAe0CbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vwwKBgF+09VI/1wJBirsDGz9whVwFFPrTkJNvJZzYt69qezx1sjgFKshy
WBhd4xHZtmCqpBP1AymEjr/T0lboxyARMXmIOWIANXMGb4KGSy11mzSVAoQ+fqR+cJ3d0dyP1j
jjb0Ed/NY8fr1NDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0i0egLda
NWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvTW+nY2jVhv7UGd8MjwUTNGItbdb6nsYqM2asrnF3qS
VRkAKKKYeGjKpUfVTiW0YFjXkfcR/V+QFL50ndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----
```

Ihr privater Schlüssel wird nicht gespeichert AWS und kann erst abgerufen werden, wenn er erstellt wurde. Später kann er nicht mehr wiederhergestellt werden. Wenn Sie den privaten Schlüssel verlieren, müssen Sie ein neues Schlüsselpaar erstellen.

Wenn Sie über einen Linux-Computer eine Verbindung zu Ihrer Instance herstellen, sollten Sie den folgenden Befehl verwenden, um die Berechtigungen für Ihre private Schlüsseldatei festzulegen, sodass nur Sie diese lesen können.

```
$ chmod 400 MyKeyPair.pem
```

Anzeigen Ihres Schlüsselpaars

Aus dem Schlüsselpaar wird ein „Fingerabdruck“ generiert, mit dem Sie überprüfen können, ob der private Schlüssel auf Ihrem lokalen Computer dem öffentlichen Schlüssel entspricht, der in AWS gespeichert ist.

Der Fingerabdruck ist ein SHA1 Hash, der aus einer DER -codierten Kopie des privaten Schlüssels stammt. Dieser Wert wird bei der Erstellung des key pair erfasst und AWS zusammen mit dem öffentlichen Schlüssel gespeichert. Sie können den Fingerabdruck in der EC2 Amazon-Konsole oder durch Ausführen des AWS CLI Befehls anzeigen [aws ec2 describe-key-pairs](#).

Das folgende Beispiel zeigt den Fingerabdruck für MyKeyPair.

```
$ aws ec2 describe-key-pairs --key-name MyKeyPair
{
  "KeyPairs": [
    {
      "KeyName": "MyKeyPair",
      "KeyFingerprint":
        "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f"
    }
  ]
}
```

Weitere Informationen zu Schlüsseln und Fingerabdrücken finden Sie unter [Amazon EC2 Key Pairs](#) im EC2Amazon-Benutzerhandbuch.

Löschen Ihres Schlüsselpaars

Um ein key pair zu löschen, führen Sie den [aws ec2 delete-key-pair](#) Befehl aus und ersetzen Sie *MyKeyPair* mit dem Namen des Paares, das gelöscht werden soll.

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

Referenzen

AWS CLI Referenz:

- [aws ec2](#)
- [aws ec2 create-key-pair](#)
- [aws ec2 delete-key-pair](#)
- [aws ec2 describe-key-pairs](#)

Andere Referenz:

- [Amazon Elastic Compute Cloud-Dokumentation](#)
- AWS CLI Codebeispiele und Beiträge dazu AWS SDK finden Sie im [AWS Codebeispiel-Repository](#) unter GitHub.

Erstellen, Konfigurieren und Löschen von EC2 Amazon-Sicherheitsgruppen in der AWS CLI

Sie können eine Sicherheitsgruppe für Ihre Amazon Elastic Compute Cloud (AmazonEC2) -Instances erstellen, die im Wesentlichen als Firewall fungiert und Regeln enthält, die festlegen, welcher Netzwerkverkehr ein- und ausreisen kann.

Verwenden Sie AWS Command Line Interface (AWS CLI), um eine Sicherheitsgruppe zu erstellen, Regeln zu vorhandenen Sicherheitsgruppen hinzuzufügen und Sicherheitsgruppen zu löschen.

Note

Weitere Befehlsbeispiele finden Sie im [AWS CLI](#).

Themen

- [Voraussetzungen](#)
- [Eine Sicherheitsgruppe erstellen](#)
- [Hinzufügen von Regeln zu Ihrer Sicherheitsgruppe](#)
- [Löschen Ihrer Sicherheitsgruppe](#)
- [Referenzen](#)

Voraussetzungen

Zur Ausführung von ec2-Befehlen ist Folgendes erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Stellen Sie Ihre IAM Berechtigungen so ein, dass Sie Amazon EC2 Zugriff gewähren. Weitere Informationen zu IAM Berechtigungen für Amazon EC2 finden Sie unter [IAM Richtlinien für Amazon EC2](#) im EC2Amazon-Benutzerhandbuch.

Eine Sicherheitsgruppe erstellen

Sie können Sicherheitsgruppen erstellen, die virtuellen privaten Clouds zugeordnet sind (VPCs).

Das folgende [aws ec2 create-security-group](#) Beispiel zeigt, wie Sie eine Sicherheitsgruppe für eine bestimmte Gruppe erstellenVPC.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --  
vpc-id vpc-1a2b3c4d  
{  
  "GroupId": "sg-903004f8"  
}
```

Zum Anzeigen der Anfangsinformationen für eine Sicherheitsgruppe führen Sie den Befehl [aws ec2 describe-security-groups](#) aus. Sie können auf eine VPC Sicherheitsgruppe vom Typ EC2 - nur anhand ihres Namens verweisen `vpc-id`, nicht anhand ihres Namens.

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8  
{  
  "SecurityGroups": [  
    {  
      "IpPermissionsEgress": [  
        {  
          "IpProtocol": "-1",  
          "IpRanges": [  
            {  
              "CidrIp": "0.0.0.0/0"  
            }  
          ],  
          "UserIdGroupPairs": []  
        }  
      ],  
      "Description": "My security group"  
      "IpPermissions": [],  
      "GroupName": "my-sg",  
      "VpcId": "vpc-1a2b3c4d",  
      "OwnerId": "123456789012",  
      "GroupId": "sg-903004f8"  
    }  
  ]  
}
```

Hinzufügen von Regeln zu Ihrer Sicherheitsgruppe

Wenn Sie eine EC2 Amazon-Instance ausführen, müssen Sie Regeln in der Sicherheitsgruppe aktivieren, um eingehenden Netzwerkverkehr für Ihre Verbindungsmöglichkeiten mit dem Image zuzulassen.

Wenn Sie beispielsweise eine Windows-Instance starten, fügen Sie normalerweise eine Regel hinzu, die eingehenden Datenverkehr auf TCP Port 3389 zulässt, um das Remote Desktop Protocol (RDP) zu unterstützen. Wenn Sie eine Linux-Instance starten, fügen Sie in der Regel eine Regel hinzu, die eingehenden Datenverkehr auf TCP Port 22 zulässt, um Verbindungen zu unterstützen. SSH

Fügen Sie mit dem Befehl [aws ec2 authorize-security-group-ingress](#) eine Regel zu Ihrer Sicherheitsgruppe hinzu. Ein erforderlicher Parameter dieses Befehls ist die öffentliche IP-Adresse Ihres Computers oder das Netzwerk (in Form eines Adressbereichs), an das Ihr Computer angeschlossen ist, in [CIDR](#)Notation.

Note

Wir bieten den folgenden Dienst, <https://checkip.amazonaws.com/>, an, damit Sie Ihre öffentliche IP-Adresse ermitteln können. Zum Finden weiterer Services zur Identifizierung Ihrer IP-Adresse geben Sie in Ihren Browser "wie lautet meine IP-Adresse" ein. Wenn Sie über eine ISP oder hinter Ihrer Firewall eine Verbindung mit einer dynamischen IP-Adresse herstellen (über ein NAT Gateway aus einem privaten Netzwerk), kann sich Ihre Adresse regelmäßig ändern. In diesem Fall müssen Sie den IP-Adressbereich herausfinden, der von Client-Computern verwendet wird.

Das folgende Beispiel zeigt, wie Sie eine Regel für RDP (TCP Port 3389) zu einer EC2 VPC Sicherheitsgruppe mit der ID hinzufügen, die Ihre IP-Adresse `sg-903004f8` verwendet.

Suchen Sie zunächst Ihre IP-Adresse.

```
$ curl https://checkip.amazonaws.com
x.x.x.x
```

Sie können die IP-Adresse dann zur Sicherheitsgruppe hinzufügen, indem Sie den [aws ec2 authorize-security-group-ingress](#)-Befehl ausführen.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 3389 --cidr x.x.x.x/x
```

Der folgende Befehl fügt eine weitere Regel hinzu, die SSH für Instanzen in derselben Sicherheitsgruppe aktiviert werden soll.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 22 --cidr x.x.x.x/x
```

Zum Anzeigen der Änderungen der Sicherheitsgruppe führen Sie den Befehl [aws ec2 describe-security-groups](#) aus.

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8
{
  "SecurityGroups": [
    {
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "UserIdGroupPairs": []
        }
      ],
      "Description": "My security group"
      "IpPermissions": [
        {
          "ToPort": 22,
          "IpProtocol": "tcp",
          "IpRanges": [
            {
              "CidrIp": "x.x.x.x/x"
            }
          ],
          "UserIdGroupPairs": [],
          "FromPort": 22
        }
      ],
      "GroupName": "my-sg",
      "OwnerId": "123456789012",
      "GroupId": "sg-903004f8"
    }
  ]
}
```

Löschen Ihrer Sicherheitsgruppe

Um eine Sicherheitsgruppe zu löschen, führen Sie den Befehl [aws ec2 delete-security-group](#) aus.

Note

Sie können eine Sicherheitsgruppe nicht löschen, wenn sie aktuell an eine Umgebung angefügt ist.

Mit dem folgenden Befehlsbeispiel wird eine VPC Sicherheitsgruppe vom EC2 Typ - gelöscht.

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

Referenzen

AWS CLI Referenz:


- [aws ec2](#)
- [aws ec2 authorize-security-group-ingress](#)
- [aws ec2 create-security-group](#)
- [aws ec2 delete-security-group](#)
- [aws ec2 describe-security-groups](#)

Andere Referenz:

- [Amazon Elastic Compute Cloud-Dokumentation](#)
- AWS CLI Codebeispiele und Beiträge dazu AWS SDK finden Sie im [AWS Codebeispiel-Repository](#) unter GitHub.

Starten, Auflisten und Schließen von EC2 Amazon-Instances in der AWS CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Amazon Elastic Compute Cloud (AmazonEC2) -Instances zu starten, aufzulisten und zu beenden. Wenn Sie eine Instance starten, die nicht unter das AWS kostenlose Kontingent fällt, wird Ihnen nach dem Start der Instance eine Rechnung gestellt, und zwar für die Zeit, in der die Instance läuft, auch wenn sie inaktiv bleibt.

 Note

Weitere Befehlsbeispiele finden Sie im [AWS CLI AWS CLI Referenzhandbuch](#)

Themen

- [Voraussetzungen](#)
- [Starten Ihrer Instance](#)
- [Hinzufügen eines Blockgeräts zu Ihrer Instance](#)
- [Hinzufügen eines Tags zu Ihrer Instance](#)
- [Herstellen einer Verbindung zu Ihrer Instance](#)
- [Auflisten Ihrer Instances](#)
- [Beenden Ihrer Instance](#)
- [Referenzen](#)

Voraussetzungen

Um die ec2-Befehle in diesem Thema auszuführen, sind folgende Schritte erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Stellen Sie Ihre IAM Berechtigungen so ein, dass Sie Amazon EC2 Zugriff gewähren. Weitere Informationen zu IAM Berechtigungen für Amazon EC2 finden Sie unter [IAM Richtlinien für Amazon EC2](#) im EC2Amazon-Benutzerhandbuch.
- Erstellen Sie ein [Schlüsselpaar](#) und eine [Sicherheitsgruppe](#).
- Wählen Sie ein Amazon Machine Image (AMI) aus und notieren Sie sich die AMI ID. Weitere Informationen [finden Sie unter Suchen eines geeigneten AMI](#) Produkts im EC2Amazon-Benutzerhandbuch.

Starten Ihrer Instance

Verwenden Sie den [aws ec2 run-instances](#) Befehl, um eine EC2 Amazon-Instance mit der von AMI Ihnen ausgewählten zu starten. Sie können die Instance in einer virtuellen privaten Cloud (VPC) starten.

Die Instance weist zu Beginn den Status `pending` auf. Sie wechselt aber nach wenigen Minuten in den Status `running`.

Das folgende Beispiel zeigt, wie eine `t2.micro` Instanz im angegebenen Subnetz von a VPC gestartet wird. Ersetzen Sie das *italicized* Parameterwerte durch Ihre eigenen.

```
$ aws ec2 run-instances --image-id ami-xxxxxxx --count 1 --instance-type t2.micro --
key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
{
  "OwnerId": "123456789012",
  "ReservationId": "r-5875ca20",
  "Groups": [
    {
      "GroupName": "my-sg",
      "GroupId": "sg-903004f8"
    }
  ],
  "Instances": [
    {
      "Monitoring": {
        "State": "disabled"
      },
      "PublicDnsName": null,
      "Platform": "windows",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "EbsOptimized": false,
      "LaunchTime": "2013-07-19T02:42:39.000Z",
      "PrivateIpAddress": "10.0.1.114",
      "ProductCodes": [],
      "VpcId": "vpc-1a2b3c4d",
      "InstanceId": "i-5203422c",
      "ImageId": "ami-173d747e",
      "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
      "KeyName": "MyKeyPair",
      "SecurityGroups": [
        {
          "GroupName": "my-sg",
          "GroupId": "sg-903004f8"
        }
      ]
    }
  ],
}
```

```
"ClientToken": null,
"SubnetId": "subnet-6e7f829e",
"InstanceType": "t2.micro",
"NetworkInterfaces": [
  {
    "Status": "in-use",
    "SourceDestCheck": true,
    "VpcId": "vpc-1a2b3c4d",
    "Description": "Primary network interface",
    "NetworkInterfaceId": "eni-a7edb1c9",
    "PrivateIpAddresses": [
      {
        "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
        "Primary": true,
        "PrivateIpAddress": "10.0.1.114"
      }
    ],
    "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
    "Attachment": {
      "Status": "attached",
      "DeviceIndex": 0,
      "DeleteOnTermination": true,
      "AttachmentId": "eni-attach-52193138",
      "AttachTime": "2013-07-19T02:42:39.000Z"
    },
    "Groups": [
      {
        "GroupName": "my-sg",
        "GroupId": "sg-903004f8"
      }
    ],
    "SubnetId": "subnet-6e7f829e",
    "OwnerId": "123456789012",
    "PrivateIpAddress": "10.0.1.114"
  }
],
"SourceDestCheck": true,
"Placement": {
  "Tenancy": "default",
  "GroupName": null,
  "AvailabilityZone": "us-west-2b"
},
"Hypervisor": "xen",
"BlockDeviceMappings": [
```

```
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": true,
        "VolumeId": "vol-877166c8",
        "AttachTime": "2013-07-19T02:42:39.000Z"
      }
    }
  ],
  "Architecture": "x86_64",
  "StateReason": {
    "Message": "pending",
    "Code": "pending"
  },
  "RootDeviceName": "/dev/sda1",
  "VirtualizationType": "hvm",
  "RootDeviceType": "ebs",
  "Tags": [
    {
      "Value": "MyInstance",
      "Key": "Name"
    }
  ],
  "AmiLaunchIndex": 0
}
]
```

Hinzufügen eines Blockgeräts zu Ihrer Instance

Jede gestartete Instance hat ein zugehöriges Root-Gerät-Volume. Sie können die Blockgerätezuweisung verwenden, um zusätzliche Amazon Elastic Block Store (AmazonEBS) - Volumes oder Instance-Speicher-Volumes anzugeben, die an eine Instance angehängt werden sollen, wenn sie gestartet wird.

Zum Hinzufügen eines Blockgeräts zu Ihrer Instance geben Sie die Option `--block-device-mappings` an, wenn Sie `run-instances` verwenden.

Der folgende Beispielparameter stellt ein EBS Amazon-Standardvolume mit einer Größe von 20 GB bereit und ordnet es mithilfe der Kennung Ihrer Instance zu `/dev/sdf`.

```
--block-device-mappings "[{"DeviceName":"/dev/sdf","Ebs":{"VolumeSize":20,
"DeleteOnTermination":false}]"
```

Das folgende Beispiel fügt ein EBS Amazon-Volume hinzu, dem auf der Grundlage eines vorhandenen Snapshots zugeordnet wurde. /dev/sdf Ein Snapshot stellt ein Image dar, das für Sie in das Volume geladen wurde. Wenn Sie einen Snapshot angeben, müssen Sie keine Volume-Größe angeben; es wird groß genug für Ihr Image sein. Wenn Sie aber einen Größenwert angeben, muss dieser größer oder gleich der Snapshot-Größe sein.

```
--block-device-mappings [{"DeviceName":"/dev/sdf","Ebs":{"SnapshotId":"snap-
a1b2c3d4"}}]
```

Im folgenden Beispiel werden zwei Volumes zu Ihrer Instance hinzugefügt. Die Anzahl der Volumes, die für Ihre Instance verfügbar sind, hängt vom Instance-Typ ab.

```
--block-device-mappings [{"DeviceName":"/dev/sdf","VirtualName":"ephemeral0"},
{"DeviceName":"/dev/sdg","VirtualName":"ephemeral1"}]
```

Das folgende Beispiel erstellt das Mapping (/dev/sdj), stellt aber kein Volume für die Instance bereit.

```
--block-device-mappings [{"DeviceName":"/dev/sdj","NoDevice":""}]"
```

Weitere Informationen finden Sie unter [Block Device Mapping](#) im EC2Amazon-Benutzerhandbuch.

Hinzufügen eines Tags zu Ihrer Instance

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource zuweisen. Damit können Sie Metadaten zu Ressourcen hinzufügen, die Sie für eine Vielzahl von Zwecken einsetzen können. Weitere Informationen finden Sie unter [Tagging Your Resources](#) im EC2Amazon-Benutzerhandbuch.

Das folgende Beispiel zeigt, wie Sie ein Tag mit dem Schlüsselnamen „Name“ und den Wert „MyInstance“ zur angegebenen Instance hinzufügen, in dem Sie den Befehl [aws ec2 create-tags](#) verwenden.

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

Herstellen einer Verbindung zu Ihrer Instance

Wenn die Instance ausgeführt wird, können Sie eine Verbindung mit ihr herstellen und sie genau so verwenden wie einen Computer, der sich direkt vor Ihnen befindet. Weitere Informationen finden Sie unter [Connect to Your Amazon EC2 Instance](#) im EC2Amazon-Benutzerhandbuch.

Auflisten Ihrer Instances

Sie können den verwenden AWS CLI , um Ihre Instances aufzulisten und Informationen zu ihnen einzusehen. Sie können alle Ihre Instances auflisten oder die Ergebnisse auf der Grundlage der Instances, die für Sie von Interesse sind, filtern.

Die folgenden Beispiele demonstrieren die Verwendung des [aws ec2 describe-instances](#)-Befehls.

Mit dem folgenden Befehl werden alle Ihre Instances aufgelistet.

```
$ aws ec2 describe-instances
```

Der folgende Befehl filtert die Liste nur nach Ihren t2.micro-Instances und gibt nur die InstanceId-Werte für die einzelnen Übereinstimmungen aus.

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query  
"Reservations[].Instances[].InstanceId"  
[  
  "i-05e998023d9c69f9a"  
]
```

Der folgende Befehl listet alle Ihre Instances mit dem Tag Name=MyInstance auf.

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

Der folgende Befehl listet Ihre Instances auf, die mit einer der folgenden Methoden gestartet wurden AMIs: ami-x0123456ami-y0123456, undami-z0123456.

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-  
y0123456,ami-z0123456"
```

Beenden Ihrer Instance

Das Beenden einer Instance löscht diese. Sie können mit einer Instance keine Verbindung mehr herstellen, nachdem Sie sie beendet haben.

Sobald der Status der Instance zu `shutting-down` oder `terminated` wechselt, fallen für diese Instance keine Gebühren mehr an. Wenn Sie später eine erneute Verbindung zu einer Instance herstellen möchten, verwenden Sie [stop-instances](#) statt `terminate-instances`. Weitere Informationen finden Sie unter [Terminate Your Instance](#) im EC2Amazon-Benutzerhandbuch.

Zum Löschen einer Instance verwenden Sie den Befehl [aws ec2 terminate-instances](#).

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-5203422c",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

Referenzen

AWS CLI Referenz:

- [aws ec2](#)
- [aws ec2 create-tags](#)
- [aws ec2 describe-instances](#)
- [aws ec2 run-instances](#)
- [aws ec2 terminate-instances](#)

Andere Referenz:

- [Amazon Elastic Compute Cloud-Dokumentation](#)
- AWS CLI Codebeispiele und Beiträge dazu AWS SDK finden Sie im [AWS Codebeispiel-Repository](#) unter GitHub.

Ändern Sie einen EC2 Amazon-Instance-Typ mit einem Bash-Skript in AWS CLI

Dieses Bash-Scripting-Beispiel für Amazon EC2 ändert den Instance-Typ für eine EC2 Amazon-Instance mithilfe von AWS Command Line Interface (AWS CLI). Dabei wird die Instance gestoppt, wenn sie ausgeführt wird, der Instance-Typ wird geändert und dann wird die Instance, falls angefordert, neu gestartet. Shell-Skripte sind Programme, die in einer Befehlszeilenschnittstelle ausgeführt werden sollen.

Note

Weitere Befehlsbeispiele finden Sie im [AWS CLI AWS CLI Referenzhandbuch](#)

Themen

- [Bevor Sie beginnen](#)
- [Über das Beispiel](#)
- [Parameter](#)
- [Dateien](#)
- [Referenzen](#)

Bevor Sie beginnen

Bevor Sie eines der folgenden Beispiele ausführen können, müssen die folgenden Schritte abgeschlossen werden.

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Das von Ihnen verwendete Profil muss über Berechtigungen verfügen, die die in den Beispielen ausgeführten AWS Operationen zulassen.
- Eine laufende EC2 Amazon-Instance in dem Konto, für das Sie die Erlaubnis haben, sie zu beenden und zu ändern. Wenn Sie das Testskript ausführen, startet es eine Instance, testet die Änderung des Typs und beendet dann die Instance.

- Es hat sich AWS bewährt, diesem Code die geringsten Rechte oder nur die für die Ausführung einer Aufgabe erforderlichen Berechtigungen zu gewähren. Weitere Informationen finden Sie unter [Grant Least Privilege](#) im AWS Identity and Access Management (IAM) -Benutzerhandbuch.
- Dieser Code wurde nicht in allen AWS Regionen getestet. Einige AWS Dienste sind nur in bestimmten Regionen verfügbar. Weitere Informationen finden Sie unter [Service-Endpunkte und Kontingente](#) im allgemeinen AWS -Referenzhandbuch.
- Wenn Sie diesen Code ausführen, kann dies zu Gebühren für Ihr AWS Konto führen. Es liegt in Ihrer Verantwortung sicherzustellen, dass alle durch dieses Skript erstellten Ressourcen entfernt werden, wenn Sie mit ihnen fertig sind.

Über das Beispiel

Dieses Beispiel ist als Funktion in der Shell-Skriptdatei `change_ec2_instance_type.sh` geschrieben, die Sie von einem anderen Skript oder von der Befehlszeile aus `source` können. Jede Skriptdatei enthält Kommentare, die jede der Funktionen beschreiben. Sobald sich die Funktion im Speicher befindet, können Sie sie über die Befehlszeile aufrufen. Mit den folgenden Befehlen wird beispielsweise der Typ der angegebenen Instance in `t2.nano` geändert:

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

Das vollständige Beispiel und die herunterladbaren Skriptdateien finden Sie unter [Ändern des EC2 Amazon-Instance-Typs](#) im AWS Codebeispiel-Repository auf GitHub.

Parameter

`-i` – (String) Gibt die zu ändernde Instance-ID an.

`-t` — (string) Gibt den EC2 Amazon-Instance-Typ an, zu dem gewechselt werden soll.

`-r` – (Switch) Ist standardmäßig nicht festgelegt. Wenn `-r` festgelegt ist, startet die Instance nach dem Typwechsel neu.

`-f` – (Switch) Standardmäßig fordert das Skript den Benutzer auf, das Herunterfahren der Instance zu bestätigen, bevor der Wechsel vorgenommen wird. Wenn `-f` festgelegt ist, wird der Benutzer nicht aufgefordert, das Herunterfahren der Instance zu bestätigen, bevor der Typwechsel vorgenommen wird.

`-V` – (Switch) Standardmäßig arbeitet das Skript im Hintergrund und zeigt die Ausgabe nur im Fehlerfall an. Wenn `-v` festgelegt ist, zeigt die Funktion während der gesamten Ausführungszeit den Status an.

Dateien

change_ec2_instance_type.sh

Die Hauptskriptdatei enthält die `change_ec2_instance_type()`-Funktion, die die folgenden Aufgaben ausführt:

- Überprüft, ob die angegebene EC2 Amazon-Instance existiert.
- Warnt den Benutzer, bevor die Instance gestoppt wird, sofern `-f` nicht ausgewählt ist.
- Ändert den Instance-Typ
- Wenn Sie `-r` festlegen, wird die Instance neu gestartet und bestätigt, dass die Instance ausgeführt wird

Sehen Sie sich den Code für [change_ec2_instance_type.sh](#) on an. GitHub

test_change_ec2_instance_type.sh

Das `test_change_ec2_instance_type.sh`-Dateiskript testet die verschiedenen Codepfade für die `change_ec2_instance_type`-Funktion. Wenn alle Schritte im Testskript ordnungsgemäß funktionieren, entfernt das Testskript alle von ihm erstellten Ressourcen.

Sie können das Testskript mit den folgenden Parametern ausführen:

- `-v` — (Schalter) Jeder Test zeigt einen `pass/failure` status as they run. By default, the tests runs silently and the output includes only the final overall `pass/failure` Status an.
- `-i` – (Schalter) Das Skript wird nach jedem Test angehalten, damit Sie die Zwischenergebnisse jedes Schritts durchsuchen können. Ermöglicht es Ihnen, den aktuellen Status der Instance mithilfe der EC2 Amazon-Konsole zu überprüfen. Das Skript fährt mit dem nächsten Schritt fort, nachdem Sie ENTER an der Eingabeaufforderung gedrückt haben.

Sehen Sie sich den Code für [test_change_ec2_instance_type.sh](#) on an GitHub.

awsdocs_general.sh

Die Skriptdatei `awsdocs_general.sh` enthält allgemeine Funktionen, die in erweiterten Beispielen für die AWS CLI verwendet werden.

Sehen Sie sich den Code für [awsdocs_general.sh](#) on an GitHub.

Referenzen

AWS CLI Referenz:

- [aws ec2](#)
- [aws ec2 describe-instances](#)
- [aws ec2 modify-instance-attribute](#)
- [aws ec2 start-instances](#)
- [aws ec2 stop-instances](#)
- [aws ec2 wait instance-running](#)
- [aws ec2 wait instance-stopped](#)

Andere Referenz:

- [Amazon Elastic Compute Cloud-Dokumentation](#)
- AWS CLI Codebeispiele und Beiträge dazu AWS SDK finden Sie im [AWS Codebeispiel-Repository](#) unter GitHub.

Verwendung von Amazon S3 Glacier in der AWS CLI

Einführung in Amazon S3 Glacier

[Einführung in Amazon S3 Glacier](#)

Dieses Thema zeigt Beispiele für AWS CLI Befehle, die allgemeine Aufgaben für S3 Glacier ausführen. Die Beispiele zeigen, wie Sie mithilfe von eine große Datei AWS CLI auf S3 Glacier hochladen können, indem Sie sie in kleinere Teile aufteilen und diese über die Befehlszeile hochladen.

Mit dem AWS Command Line Interface (AWS CLI) können Sie auf die Funktionen von Amazon S3 Glacier zugreifen. Verwenden Sie den folgenden Befehl, um die AWS CLI Befehle für S3 Glacier aufzulisten.

```
aws glacier help
```

Note

Weitere Befehlsreferenzen und -beispiele finden Sie unter [aws glacier](#) in der AWS CLI - Befehlsreferenz..

Themen

- [Voraussetzungen](#)
- [Erstellen eines Amazon-S3-Glacier-Tresors](#)
- [Vorbereiten einer Datei zum Hochladen](#)
- [Starten eines mehrteiligen Uploads und Hochladen der Dateien](#)
- [Abschließen des Uploads](#)
- [Ressourcen](#)

Voraussetzungen

Zur Ausführung von `glacier`-Befehlen ist Folgendes erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Dieses Tutorial verwendet mehrere Befehlszeilentools, die normalerweise auf Unix-ähnlichen Betriebssystemen vorinstalliert sind, einschließlich Linux und macOS. Windows-Benutzer können die gleichen Tools verwenden, indem Sie [Cygwin](#) installieren und die Befehle vom Cygwin-Terminal ausführen. Wir geben systemeigene Windows-Befehle und -Hilfsprogramme an, die die gleichen Funktionen ausführen, sofern verfügbar.

Erstellen eines Amazon-S3-Glacier-Tresors

Erstellen Sie einen Tresor mit dem Befehl [create-vault](#).

```
$ aws glacier create-vault --account-id - --vault-name myvault
{
  "location": "/123456789012/vaults/myvault"
```

```
}
```

Note

Alle S3-Glacier-Befehle erfordern einen Konto-ID-Parameter. Verwenden Sie den Bindestrich (`--account-id -`), um das aktuelle Konto zu verwenden.

Vorbereiten einer Datei zum Hochladen

Erstellen Sie eine Datei für einen Probe-Upload. Mit den folgenden Befehlen wird eine Datei mit dem Namen erstellt *largefile* das enthält genau 3 MiB an Zufallsdaten.

Linux oder macOS

```
$ dd if=/dev/urandom of=largefile bs=3145728 count=1
1+0 records in
1+0 records out
3145728 bytes (3.1 MB) copied, 0.205813 s, 15.3 MB/s
```

dd ist ein Hilfsprogramm, das eine Anzahl von Bytes aus einer Eingabedatei in eine Ausgabedatei kopiert. Im vorigen Beispiel wird die Systemgerätedatei `/dev/urandom` als Quelle für Zufallsdaten verwendet. `fsutil` führt eine ähnliche Funktion in Windows aus.

Windows

```
C:\> fsutil file createnew largefile 3145728
File C:\temp\largefile is created
```

Teilen Sie als Nächstes die Datei mithilfe eines Dateisplitters in Blöcke mit 1 MiB (1.048.576 Byte) auf.

```
$ split -b 1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

Starten eines mehrteiligen Uploads und Hochladen der Dateien

Erstellen Sie in Amazon S3 Glacier einen mehrteiligen Upload, indem Sie den Befehl [initiate-multipart-upload](#) verwenden.

```
$ aws glacier initiate-multipart-upload --account-id - --archive-description "multipart
upload test" --part-size 1048576 --vault-name myvault
{
  "uploadId": "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ",
  "location": "/123456789012/vaults/myvault/multipart-
uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
}
```

S3 Glacier benötigt die Größe jedes Teils in Byte (in diesem Beispiel 1 MiB), Ihren Tresornamen und die Konto-ID, um den mehrteiligen Upload zu konfigurieren. Der AWS CLI gibt eine Upload-ID aus, wenn der Vorgang abgeschlossen ist. Speichern Sie diesen Upload-ID zur späteren Verwendung in einer Shell-Variablen.

Linux oder macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
```

Verwenden Sie als Nächstes den Befehl [upload-multipart-part](#), um die drei Teile hochzuladen.

```
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkaa --range 'bytes
0-1048575/*' --account-id - --vault-name myvault
{
  "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkab --range 'bytes
1048576-2097151/*' --account-id - --vault-name myvault
{
```

```
"checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkac --range 'bytes
2097152-3145727/*' --account-id - --vault-name myvault
{
  "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
```

Note

Im vorigen Beispiel wird das Dollarzeichen (\$) verwendet, um die Inhalte der UPLOADID Shell-Variablen unter Linux zu referenzieren. Verwenden Sie in der Windows-Befehlszeile ein Prozentzeichen (%) auf beiden Seiten des Variablennamens (z. B. %UPLOADID%).

Sie müssen den Byte-Bereich für jeden einzelnen Teil angeben, wenn Sie diese hochladen, damit S3 Glacier die Teile in der richtigen Reihenfolge wieder zusammensetzen kann. Jeder Teil hat 1 048 576 Bytes. Der erste Teil belegt somit Byte 0-1048575, der zweite 1048576-2097151 und der dritte 2097152-3145727.

Abschließen des Uploads

Amazon S3 Glacier benötigt einen Tree-Hash der Originaldatei, um zu bestätigen, dass alle hochgeladenen Teile AWS unversehrt erreicht wurden.

Um einen Tree-Hash zu berechnen, müssen Sie die Datei in 1 MiB-Teile aufteilen und für jedes Stück einen binären SHA -256-Hash berechnen. Dann teilen Sie die Liste der Hashes in Paare auf. Sie kombinieren die beiden binären Hashes in jedem Paar und ermitteln die Hashes der Ergebnisse. Wiederholen Sie diesen Prozess, bis nur ein Hash übrig ist. Wenn es auf einer beliebigen Ebene eine ungerade Anzahl von Hashes gibt, verschieben Sie diese auf die nächste Ebene, ohne sie zu bearbeiten.

Das Wichtigste bei der korrekten Berechnung von einem Struktur-Hash unter Verwendung von Befehlszeilen-Hilfsprogrammen ist, den jeweiligen Hash im Binärformat zu speichern und erst im letzten Schritt in einen Hexadezimalwert zu konvertieren. Wenn Sie eine Hexadezimalversion von einem beliebigen Hash in der Struktur kombinieren oder hashen, erhalten Sie ein falsches Ergebnis.

Note

Windows-Benutzer können den Befehl `type` anstelle von `cat` verwenden. Open SSL ist für Windows unter [SSLOpen.org](https://www.openssl.org) verfügbar.

So berechnen Sie einen Struktur-Hash:

1. Falls noch nicht geschehen, teilen Sie die Originaldatei in 1-MiB-Teile auf.

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

2. Berechnet und speichert den binären SHA -256-Hash jedes Chunks.

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. Kombinieren Sie die ersten beiden Hashes und ermitteln Sie den binären Hash des Ergebnisses.

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. Kombinieren Sie den übergeordnete Hash der Blöcke aa und ab mit dem Hash ac und hashen Sie das Ergebnis, dieses Mal mit einer hexadezimalen Ausgabe. Speichern Sie das Ergebnis in einer Shell-Variablen.

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

Schließen Sie den Upload mit dem Befehl [complete-multipart-upload](#) ab. Dieser Befehl akzeptiert die Größe der ursprünglichen Datei in Bytes, den endgültigen Hash-Wert in Hexadezimalformat und Ihre Konto-ID sowie den Tresornamen.

```
$ aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728 --
upload-id $UPLOADID --account-id - --vault-name myvault
{
  "archiveId": "d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrALLGAA0NJAzo5QdP-
N83MKqd96Unspoa5H51ItWX-sK8-QS0ZhwsyGiu9-R-
kwWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
  "checksum": "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
  "location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrALLGAA0NJAzo5QdP-N83MKqd96Unspoa5H51ItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

Sie können auch den Status des Tresors mithilfe des Befehls [describe-vault](#) überprüfen.

```
$ aws glacier describe-vault --account-id - --vault-name myvault
{
  "SizeInBytes": 3178496,
  "VaultARN": "arn:aws:glacier:us-west-2:123456789012:vaults/myvault",
  "LastInventoryDate": "2018-12-07T00:26:19.028Z",
  "NumberOfArchives": 1,
  "CreationDate": "2018-12-06T21:23:45.708Z",
  "VaultName": "myvault"
}
```

Note

Der Tresor-Status wird etwa einmal pro Tag aktualisiert. Weitere Informationen finden Sie unter [Arbeiten mit Tresoren](#).

Sie können jetzt die von Ihnen erstellten Chunk- und Hash-Dateien entfernen.

```
$ rm chunk* hash*
```

Weitere Informationen über mehrteilige Uploads finden Sie unter [Hochladen großer Archive in Teilen](#) und [Berechnen von Prüfsummen](#) im Amazon-S3-Glacier-Entwicklerhandbuch.

Ressourcen

AWS CLI Referenz:

- [aws glacier](#)
- [aws glacier complete-multipart-upload](#)
- [aws glacier create-vault](#)
- [aws glacier describe-vault](#)
- [aws glacier initiate-multipart-upload](#)

Service-Referenz:

- [Entwicklerhandbuch für Amazon S3 Glacier](#)
- [Hochladen von großen Archiven in Teilen](#) im Entwicklerhandbuch für Amazon S3 Glacier
- [Berechnen von Prüfsummen](#) im Entwicklerhandbuch für Amazon S3 Glacier
- [Arbeiten mit Tresoren](#) im Entwicklerhandbuch für Amazon S3 Glacier

Verwendung IAM in der AWS CLI

Eine Einführung in AWS Identity and Access Management

[Einführung in AWS Identity and Access Management](#)

Sie können mit dem AWS Identity and Access Management (IAM) auf die Funktionen von AWS Command Line Interface (AWS CLI) zugreifen. Verwenden Sie den folgenden Befehl IAM, um die AWS CLI Befehle für aufzulisten.

```
aws iam help
```

Dieses Thema enthält Beispiele für AWS CLI Befehle, mit denen allgemeine Aufgaben für ausgeführt IAM werden.

Bevor Sie Befehle ausführen, richten Sie die Standardanmeldeinformationen ein. Weitere Informationen finden Sie unter [Konfiguration der Einstellungen für den AWS CLI](#).

Weitere Informationen zum IAM Dienst finden Sie im [AWS Identity and Access Management Benutzerhandbuch](#).

Themen

- [Erstellen von IAM-Benutzern und -Gruppen](#)
- [Einem Benutzer eine IAM verwaltete Richtlinie zuordnen](#)
- [Ein erstes Passwort für einen IAM Benutzer einrichten](#)
- [Einen Zugriffsschlüssel für einen IAM Benutzer erstellen](#)

Erstellen von IAM-Benutzern und -Gruppen

So erstellen Sie eine Gruppe und fügen dieser einen neuen Benutzer hinzu

1. Verwenden Sie den Befehl [create-group](#), um die Gruppe zu erstellen.

```
$ aws iam create-group --group-name MyIamGroup
{
  "Group": {
    "GroupName": "MyIamGroup",
    "CreateDate": "2018-12-14T03:03:52.834Z",
    "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
    "Path": "/"
  }
}
```

2. Verwenden Sie den Befehl [create-user](#), um den Benutzer zu erstellen.

```
$ aws iam create-user --user-name MyUser
{
  "User": {
    "UserName": "MyUser",
    "Path": "/",
    "CreateDate": "2018-12-14T03:13:02.581Z",
    "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/MyUser"
  }
}
```

3. Fügen Sie mit dem Befehl [add-user-to-group](#) den Benutzer zur Gruppe hinzu.

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. Vergewissern Sie sich, dass die MyIamGroup-Gruppe MyUser enthält. Nutzen Sie dazu den Befehl [get-group](#).

```
$ aws iam get-group --group-name MyIamGroup
{
  "Group": {
    "GroupName": "MyIamGroup",
    "CreateDate": "2018-12-14T03:03:52Z",
    "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
    "Path": "/"
  },
  "Users": [
    {
      "UserName": "MyUser",
      "Path": "/",
      "CreateDate": "2018-12-14T03:13:02Z",
      "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/MyUser"
    }
  ],
  "IsTruncated": "false"
}
```

Einem Benutzer eine IAM verwaltete Richtlinie zuordnen

Die Richtlinie dieses Beispiels stattet den Benutzer mit "Hauptbenutzerberechtigungen" aus.

Um eine IAM verwaltete Richtlinie an einen Benutzer anzuhängen

1. Ermitteln Sie den Amazon-Ressourcennamen (ARN) der anzuhängenden Richtlinie. Der folgende Befehl verwendet `list-policies`, um ARN die Richtlinie mit dem Namen zu finden `PowerUserAccess`. Das wird dann ARN in einer Umgebungsvariablen gespeichert.

```
$ export POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text) ~
$ echo $POLICYARN
arn:aws:iam::aws:policy/PowerUserAccess
```

2. Um die Richtlinie anzuhängen, verwenden Sie den [attach-user-policy](#) Befehl und verweisen Sie auf die Umgebungsvariable, die die Richtlinie enthält ARN.

```
$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN
```

- Überprüfen Sie, ob die Richtlinie dem Benutzer zugewiesen ist, indem Sie den Befehl [list-attached-user-policies](#) ausführen.

```
$ aws iam list-attached-user-policies --user-name MyUser
{
  "AttachedPolicies": [
    {
      "PolicyName": "PowerUserAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Zugriffsmanagementressourcen](#). Dieses Thema enthält Links zu einer Übersicht über Berechtigungen und Richtlinien sowie Links zu Beispielen für Richtlinien für den Zugriff auf Amazon S3EC2, Amazon und andere Services.

Ein erstes Passwort für einen IAM Benutzer einrichten

Im folgenden Befehl wird [create-login-profile](#) verwendet, um ein Anfangspasswort für den angegebenen Benutzer festzulegen. Wenn sich der Benutzer zum ersten Mal anmeldet, wird er aufgefordert, das Passwort zu einem Passwort zu ändern, das nur er kennt.

```
$ aws iam create-login-profile --user-name MyUser --password My!User1Login8P@ssword --password-reset-required
{
  "LoginProfile": {
    "UserName": "MyUser",
    "CreateDate": "2018-12-14T17:27:18Z",
    "PasswordResetRequired": true
  }
}
```

Verwenden Sie den Befehl `update-login-profile` zum Ändern des Passworts für einen Benutzer.

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

Einen Zugriffsschlüssel für einen IAM Benutzer erstellen

Sie können den Befehl [create-access-key](#) verwenden, um einen Zugriffsschlüssel für einen Benutzer zu erstellen. Ein Zugriffsschlüssel ist ein Satz von Sicherheits-Anmeldeinformationen bestehend aus einer Zugriffsschlüssel-ID und einem geheimen Schlüssel.

Ein Benutzer kann gleichzeitig maximal zwei Zugriffsschlüssel erstellen. Wenn Sie versuchen, einen dritten Satz zu erstellen, gibt der Befehl einen `LimitExceeded`-Fehler aus.

```
$ aws iam create-access-key --user-name MyUser
{
  "AccessKey": {
    "UserName": "MyUser",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "Status": "Active",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "CreateDate": "2018-12-14T17:34:16Z"
  }
}
```

Verwenden Sie den Befehl [delete-access-key](#), um einen Zugriffsschlüssel für einen Benutzer zu löschen. Geben Sie mit der Zugriffsschlüssel-ID an, welcher Zugriffsschlüssel gelöscht werden soll.

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

Verwendung von Amazon S3 in der AWS CLI

Eine Einführung in Amazon Simple Storage Service (Amazon S3)

[Einführung in Amazon Simple Storage Service \(Amazon S3\) – Cloud-Speicher auf AWS](#)

Sie können auf die Funktionen von Amazon Simple Storage Service (Amazon S3) zugreifen, indem Sie AWS Command Line Interface (AWS CLI) verwenden. Amazon S3 ist ein hoch skalierbarer und langlebiger Objektspeicherservice. Amazon S3 ist so konzipiert, dass es praktisch unbegrenzte Speicherkapazität bietet, was es zu einer idealen Lösung für eine Vielzahl von Datenspeicher- und Verwaltungsanforderungen macht.

Amazon S3 ermöglicht Ihnen das Speichern und Abrufen beliebiger Datenmengen, von kleinen Dateien bis hin zu großen Datensätzen, in Form von Objekten. Jedes Objekt wird in einem Container

gespeichert, der als Bucket bezeichnet wird. Auf diesen kann über die Tools und AWS Management Console oder programmatisch über die Tools und zugegriffen und AWS SDKs diese verwaltet werden. AWS CLI

Amazon S3 bietet neben Basisspeicher auch eine Reihe von Funktionen wie Lebenszyklusmanagement, Versionierung, Skalierbarkeit und Sicherheit. Diese lassen sich in andere integrieren, AWS-Services sodass Sie Cloud-basierte Lösungen entwickeln können, die sich an Ihre Bedürfnisse anpassen lassen.

Das AWS CLI bietet zwei Befehlsebenen für den Zugriff auf Amazon S3:

- `s3` — Benutzerdefinierte Befehle auf hoher Ebene, die speziell für AWS CLI entwickelt wurden und die Ausführung allgemeiner Aufgaben wie das Erstellen, Bearbeiten, Löschen und Synchronisieren von Objekten und Buckets vereinfachen.
- `s3api` — Bietet direkten Zugriff auf alle Amazon S3 API S3-Operationen, sodass Sie erweiterte Operationen ausführen können.

Themen in diesem Leitfaden:

- [Verwenden von Befehlen auf hoher Ebene \(s3\) in der AWS CLI](#)
- [Verwenden von API -Level \(s3api\) -Befehlen in AWS CLI](#)
- [Skriptbeispiel für den Amazon S3 S3-Bucket-Lebenszyklus in der AWS CLI](#)

Verwenden von Befehlen auf hoher Ebene (s3) in der AWS CLI

In diesem Thema wird beschrieben, wie Sie Amazon-S3-Buckets und -Objekte mit den [aws s3](#)-Befehlen in der AWS CLI verwalten. Befehle, die in diesem Thema nicht behandelt werden, und weitere Befehlsbeispiele finden Sie unter [aws s3](#)-Befehle in der AWS CLI -Referenz.

Die `aws s3`-High-Level-Befehle vereinfachen die Verwaltung von Amazon-S3-Objekten. Mit diesen Befehlen können Sie den Inhalt von Amazon S3 intern und mit lokalen Verzeichnissen verwalten.

Themen

- [Voraussetzungen](#)
- [Bevor Sie beginnen](#)
- [Erstellen eines Buckets](#)
- [Auflisten von Buckets und Objekten](#)

- [Buckets löschen](#)
- [Objekte löschen](#)
- [Verschieben von Objekten](#)
- [Kopieren von Objekten](#)
- [Synchronisieren von Objekten](#)
- [Häufig verwendete Optionen für s3-Befehle](#)
- [Ressourcen](#)

Voraussetzungen

Zur Ausführung von s3-Befehlen ist Folgendes erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Das von Ihnen verwendete Profil muss über Berechtigungen verfügen, die die in den Beispielen ausgeführten AWS Operationen zulassen.
- Sie müssen diese Amazon-S3-Begriffe verstehen:
 - Bucket – Ein Amazon-S3-Ordner der obersten Ebene.
 - Präfix – Ein Amazon-S3-Ordner in einem Bucket.
 - Objekt – Jedes Element, das in einem Amazon-S3-Bucket gehostet wird.

Bevor Sie beginnen

In diesem Abschnitt finden Sie einige Hinweise zur Verwendung von `aws s3`-Befehlen.

Uploads großer Objekte

Wenn Sie `aws s3`-Befehle zum Hochladen großer Objekte in einen Amazon-S3-Bucket verwenden, führt die AWS CLI automatisch einen mehrteiligen Upload durch. Sie können einen fehlgeschlagenen Upload nicht fortsetzen, wenn Sie diese `aws s3`-Befehle verwenden.

Wenn der mehrteilige Upload aufgrund eines Timeouts fehlschlägt oder wenn Sie den Vorgang manuell abgebrochen haben, AWS CLI stoppt der Upload und bereinigt alle erstellten Dateien. Dieser Vorgang kann einige Minuten dauern.

Wenn der mehrteilige Upload oder Bereinigungsprozess durch einen Kill-Befehl oder einen Systemfehler abgebrochen wird, verbleiben die erstellten Dateien im Amazon-S3-Bucket. <https://docs.aws.amazon.com/cli/latest/reference/s3api/abort-multipart-upload.html>

Dateieigenschaften und Tags in mehrteiligen Kopien

Wenn Sie die AWS CLI Version 1 der Befehle im `aws s3` Namespace verwenden, um eine Datei von einem Amazon S3 S3-Bucket-Speicherort an einen anderen Amazon S3 S3-Bucket-Speicherort zu kopieren, und dieser Vorgang [mehrteiliges Kopieren](#) verwendet, werden keine Dateieigenschaften vom Quellobjekt in das Zielobjekt kopiert.

Erstellen eines Buckets

Verwenden Sie den Befehl [s3 mb](#), um einen Bucket zu erstellen. Bucket-Namen müssen global eindeutig sein (einzigartig in ganz Amazon S3) und sollten DNS konform sein.

Bucket-Namen können Kleinbuchstaben, Zahlen, Bindestriche und Punkte enthalten. Bucket-Namen können nur mit einem Buchstaben oder einer Zahl beginnen und enden. Ein Punkt neben einem Bindestrich oder einem weiteren Punkt ist unzulässig.

Syntax

```
$ aws s3 mb <target> [--options]
```

s3-mb-Beispiele

Im folgenden Beispiel wird der Bucket `s3://amzn-s3-demo-bucket` erstellt.

```
$ aws s3 mb s3://amzn-s3-demo-bucket
```

Auflisten von Buckets und Objekten

Um Ihre Buckets, Ordner oder Objekte aufzulisten, verwenden Sie den Befehl [s3 ls](#). Wenn Sie den Befehl ohne Ziel oder Optionen verwenden, werden alle Buckets aufgelistet.

Syntax

```
$ aws s3 ls <target> [--options]
```


Ein paar gängige Optionen für diesen Befehl und Beispiele finden Sie unter [Häufig verwendete Optionen für s3-Befehle](#). Eine vollständige Liste der verfügbaren Optionen finden Sie unter [s3 ls](#) in der AWS CLI -Befehlsreferenz.

s3-ls-Beispiele

Das folgende Beispiel listet alle Amazon-S3-Buckets auf.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket/amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

Mit dem Befehl unten werden alle Objekte und Präfixe in einem Bucket aufgeführt. In dieser Beispielausgabe enthält das Präfix `example/` eine Datei namens `MyFile1.txt`.

```
$ aws s3 ls s3://amzn-s3-demo-bucket
                PRE example/
2018-12-04 19:05:48          3 MyFile1.txt
```

Sie können die Ausgabe nach einem bestimmten Präfix filtern, indem Sie das Präfix in den Befehl einschließen. Der folgende Befehl listet die Objekte auf in `bucket-name/example/` (das heißt, Objekte in `bucket-name` gefiltert nach dem Präfix `example/`).

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32          3 MyFile1.txt
```

Buckets löschen

Verwenden Sie zum Löschen eines Buckets den Befehl [s3 rb](#).

Syntax

```
$ aws s3 rb <target> [--options]
```

s3-rb-Beispiele

Im folgenden Beispiel wird der Bucket `s3://amzn-s3-demo-bucket` entfernt.

```
$ aws s3 rb s3://amzn-s3-demo-bucket
```

Standardmäßig muss der Bucket leer sein, damit der Vorgang erfolgreich ist. Zum Entfernen eines Buckets, der nicht leer ist, müssen Sie die Option `--force` hinzufügen. Wenn Sie einen versionsgesteuerten Bucket verwenden, der bereits gelöschte–aber aufbewahrte–Objekte enthält, lässt dieser Befehl nicht zu, dass Sie den Bucket entfernen. Sie müssen zunächst alle Inhalte entfernen.

Im folgenden Beispiel werden alle Objekte und Präfixe im Bucket gelöscht und anschließend der Bucket gelöscht.

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

Objekte löschen

Um Objekte in einem Bucket oder in Ihrem lokalen Verzeichnis zu löschen, verwenden Sie den Befehl [s3 rm](#).

Syntax

```
$ aws s3 rm <target> [--options]
```

Ein paar gängige Optionen für diesen Befehl und Beispiele finden Sie unter [Häufig verwendete Optionen für s3-Befehle](#). Eine vollständige Liste der Optionen finden Sie unter [s3 rm](#) in der AWS CLI -Befehlsreferenz.

s3-rm-Beispiele

Im folgenden Beispiel wird `filename.txt` aus `s3://amzn-s3-demo-bucket/example` gelöscht.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example/filename.txt
```

Im folgenden Beispiel werden alle Objekte aus `s3://amzn-s3-demo-bucket/example` mit der Option `--recursive` gelöscht.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example --recursive
```

Verschieben von Objekten

Verwenden Sie den Befehl [s3 mv](#), um Objekte aus einem Bucket oder einem lokalen Verzeichnis zu verschieben. Der `s3 mv` Befehl kopiert das Quellobjekt oder die Quelldatei an das angegebene Ziel und löscht dann das Quellobjekt oder die Quelldatei.

Syntax

```
$ aws s3 mv <source> <target> [--options]
```

Ein paar gängige Optionen für diesen Befehl und Beispiele finden Sie unter [Häufig verwendete Optionen für s3-Befehle](#). Eine vollständige Liste der verfügbaren Optionen finden Sie unter [s3 mv](#) in der AWS CLI -Befehlsreferenz.

Warning

Wenn Sie in Ihrer Amazon S3-Quelle ARNs oder Ihrem Amazon S3-Ziel irgendeine Art von Access Point- oder Access Point-Aliasnamen verwenden, müssen Sie besonders darauf achten, dass Ihr Amazon S3 S3-Quell- und Ziel-Alias in verschiedene zugrunde liegende Buckets URLs aufgelöst werden. Wenn Quell- und Ziel-Bucket identisch sind, kann die Quelldatei oder das Objekt auf sich selbst verschoben werden, was zu einem versehentlichen Löschen Ihrer Quelldatei oder Ihres Quellobjekts führen kann. Um zu überprüfen, ob Quell- und Ziel-Bucket nicht identisch sind, verwenden Sie den `--validate-same-s3-paths` Parameter oder setzen Sie die Umgebungsvariable [AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS](#) auf `true`.

s3-mv-Beispiele

Im folgenden Beispiel werden alle Objekte von `s3://amzn-s3-demo-bucket/example` nach `s3://amzn-s3-demo-bucket/` verschoben.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

Im folgenden Beispiel wird eine lokale Datei mit dem `s3 mv`-Befehl aus Ihrem aktuellen Arbeitsverzeichnis in den Amazon-S3-Bucket verschoben.

```
$ aws s3 mv filename.txt s3://amzn-s3-demo-bucket
```

Im folgenden Beispiel wird eine Datei aus Ihrem Amazon-S3-Bucket in Ihr aktuelles Arbeitsverzeichnis verschoben, wobei `./` Ihr aktuelles Arbeitsverzeichnis angibt.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/filename.txt ./
```

Kopieren von Objekten

Verwenden Sie den Befehl [s3 cp](#), um Objekte aus einem Bucket oder einem lokalen Verzeichnis zu verschieben.

Syntax

```
$ aws s3 cp <source> <target> [--options]
```

Sie können den Bindestrich-Parameter für das Dateistreaming an die Standardeingabe (`stdin`) oder die Standardausgabe (`stdout`) verwenden.

Warning

Wenn Sie verwenden PowerShell, kann die Shell die Kodierung einer Eingabe oder Ausgabe ändern CRLF oder eine an die Pipeline CRLF weitergeleitete Eingabe oder Ausgabe oder eine umgeleitete Ausgabe hinzufügen.

Der Befehl `s3 cp` verwendet die folgende Syntax, um einen Dateistream von `stdin` in einen angegebenen Bucket hochzuladen.

Syntax

```
$ aws s3 cp - <target> [--options]
```

Der `s3 cp`-Befehl verwendet die folgende Syntax, um einen Amazon-S3-Dateistream für `stdout` herunterzuladen.

Syntax

```
$ aws s3 cp <target> [--options] -
```

Ein paar gängige Optionen für diesen Befehl und Beispiele finden Sie unter [Häufig verwendete Optionen für s3-Befehle](#). Eine vollständige Liste der Optionen finden Sie unter [s3 cp](#) in der AWS CLI -Befehlsreferenz.

Beispiele für `s3 cp`

Im folgenden Beispiel werden alle Objekte von `s3://amzn-s3-demo-bucket/example` nach `s3://amzn-s3-demo-bucket/` kopiert.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

Im folgenden Beispiel wird eine lokale Datei mit dem `s3 cp`-Befehl aus Ihrem aktuellen Arbeitsverzeichnis in den Amazon-S3-Bucket kopiert.

```
$ aws s3 cp filename.txt s3://amzn-s3-demo-bucket
```

Im folgenden Beispiel wird eine Datei aus Ihrem Amazon-S3-Bucket in Ihr aktuelles Arbeitsverzeichnis kopiert, wobei `./` Ihr aktuelles Arbeitsverzeichnis angibt.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt ./
```

Im folgenden Beispiel wird `echo` verwendet, um den Text „hello world“ in die Datei `s3://bucket-name/filename.txt` zu streamen.

```
$ echo "hello world" | aws s3 cp - s3://amzn-s3-demo-bucket/filename.txt
```

Im folgenden Beispiel wird die `s3://amzn-s3-demo-bucket/filename.txt`-Datei nach `stdout` gestreamt und der Inhalt an die Konsole ausgegeben.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt -  
hello world
```

Das folgende Beispiel streamt den Inhalt von `s3://bucket-name/pre` nach `stdout`, verwendet den Befehl `bzip2` zum Komprimieren der Dateien und lädt die neue komprimierte Datei namens `key.bz2` nach `s3://bucket-name` hoch.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/pre - | bzip2 --best | aws s3 cp - s3://amzn-s3-  
demo-bucket/key.bz2
```

Synchronisieren von Objekten

Der Befehl [s3 sync](#) synchronisiert die Inhalte von einem Bucket und einem Verzeichnis oder die Inhalte von zwei Buckets. Normalerweise werden mit `s3 sync` fehlende oder veraltete Dateien bzw. Objekte zwischen Quelle und Ziel kopiert. Sie können aber auch die Option `--delete` hinzufügen, um Dateien oder Objekte, die nicht in der Quelldatei vorhanden sind, aus dem Ziel zu entfernen.

Syntax

```
$ aws s3 sync <source> <target> [--options]
```

Ein paar gängige Optionen für diesen Befehl und Beispiele finden Sie unter [Häufig verwendete Optionen für s3-Befehle](#). Eine vollständige Liste der Optionen finden Sie unter [s3 sync](#) in der AWS CLI -Befehlsreferenz.

Beispiele für die s3-Synchronisierung

Das folgende Beispiel synchronisiert den Inhalt eines Amazon S3 S3-Präfixes mit dem Namen path im Bucket amzn-s3-demo-bucket mit dem aktuellen Arbeitsverzeichnis.

s3 sync aktualisiert alle Dateien mit einer anderen Größe oder geänderten Zeit als Dateien mit demselben Namen am Ziel. Die Ausgabe enthält bestimmte Vorgänge, die während der Synchronisierung ausgeführt wurden. Beachten Sie, dass die Operation rekursiv das Unterverzeichnis MySubdirectory und seinen Inhalt mit s3://amzn-s3-demo-bucket/path/MySubdirectory synchronisiert.

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
upload: MyFile2.txt to s3://amzn-s3-demo-bucket/path/MyFile2.txt
upload: MyFile1.txt to s3://amzn-s3-demo-bucket/path/MyFile1.txt
```

Das folgende Beispiel, das das vorherige erweitert, zeigt die Verwendung der Option --delete.

```
// Delete local file
$ rm ./MyFile1.txt

// Attempt sync without --delete option - nothing happens
$ aws s3 sync . s3://amzn-s3-demo-bucket/path

// Sync with deletion - object is deleted from bucket
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete
delete: s3://amzn-s3-demo-bucket/path/MyFile1.txt

// Delete object from bucket
$ aws s3 rm s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
delete: s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt

// Sync with deletion - local file is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
```

```
delete: MySubdirectory\MyFile3.txt

// Sync with Infrequent Access storage class
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --storage-class STANDARD_IA
```

Bei Verwendung der Option `--delete` können mit den Optionen `--exclude` und `--include` die Dateien oder Objekte gefiltert werden, die während einer `s3 sync`-Operation gelöscht werden sollen. In diesem Fall muss die Parameterzeichenfolge Dateien angeben, die für das Zielverzeichnis oder den Bucket vom Löschen ausgenommen oder zum Löschen hinzugefügt werden. Es folgt ein Beispiel.

```
Assume local directory and s3://amzn-s3-demo-bucket/path currently in sync and each
contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt
...

// Sync with delete, excluding files that match a pattern. MyFile88.txt is deleted,
while remote MyFile1.txt is not.
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete --exclude "path/MyFile?.txt"
delete: s3://amzn-s3-demo-bucket/path/MyFile88.txt
...

// Sync with delete, excluding MyFile2.rtf - local file is NOT deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete --exclude "./MyFile2.rtf"
download: s3://amzn-s3-demo-bucket/path/MyFile1.txt to MyFile1.txt
...

// Sync with delete, local copy of MyFile2.rtf is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MyFile2.rtf
```

Häufig verwendete Optionen für s3-Befehle

Die folgenden Optionen werden häufig für die in diesem Thema beschriebenen Befehle verwendet.

<https://docs.aws.amazon.com/cli/latest/reference/>

acl

`s3 sync` und `s3 cp` können die Option `--acl` verwenden. Auf diese Weise können Sie die Zugriffsberechtigungen für Dateien festlegen, die nach Amazon S3 kopiert werden. Die Option

`--acl` akzeptiert die Werte `private`, `public-read` und `public-read-write`. Weitere Informationen finden Sie unter [Canned ACL](#) im Amazon S3 S3-Benutzerhandbuch.

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --acl public-read
```

exclude

Wenn Sie die Befehle `s3 cp`, `s3 mv`, `s3 sync` oder `s3 rm` verwenden, können Sie die Ergebnisse mit der Option `--exclude` oder `--include` filtern. Die Option `--exclude` legt Regeln fest, um nur Objekte vom Befehl auszuschließen, und die Optionen gelten in der angegebenen Reihenfolge. Dies wird im folgenden Beispiel veranschaulicht.

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Exclude all .txt files, resulting in only MyFile2.rtf being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt"

// Exclude all .txt files but include all files with the "MyFile*.txt" format,
resulting in, MyFile1.txt, MyFile2.rtf, MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
"MyFile*.txt"

// Exclude all .txt files, but include all files with the "MyFile*.txt" format,
but exclude all files with the "MyFile?.txt" format resulting in, MyFile2.rtf and
MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
"MyFile*.txt" --exclude "MyFile?.txt"
```

include

Wenn Sie die Befehle `s3 cp`, `s3 mv`, `s3 sync` oder `s3 rm` verwenden, können Sie die Ergebnisse mit der Option `--exclude` oder `--include` filtern. Die Option `--include` legt Regeln fest, um nur die für den Befehl angegebenen Objekte einzuschließen, und die Optionen gelten in der angegebenen Reihenfolge. Dies wird im folgenden Beispiel veranschaulicht.

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
```



```
MyFile88.txt

// Include all .txt files, resulting in MyFile1.txt and MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt"

// Include all .txt files but exclude all files with the "MyFile*.txt" format,
resulting in no files being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
"MyFile*.txt"

// Include all .txt files, but exclude all files with the "MyFile*.txt" format, but
include all files with the "MyFile?.txt" format resulting in MyFile1.txt being
copied

$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
"MyFile*.txt" --include "MyFile?.txt"
```

grant

Die Befehle `s3 cp`, `s3 mv` und `s3 sync` enthalten die Option `--grants`. Diese kann genutzt werden, um Berechtigungen für das Objekt an bestimmte Benutzer oder Gruppen zu erteilen. Mithilfe der folgenden Syntax legen Sie für die Option `--grants` eine Liste von Berechtigungen fest. Ersetzen Sie `Permission`, `Grantee_Type` und `Grantee_ID` durch Ihre eigenen Werte.

Syntax

```
--grants Permission=Grantee_Type=Grantee_ID
        [Permission=Grantee_Type=Grantee_ID ...]
```

Jeder Wert enthält die folgenden Elemente:

- *Permission* — Gibt die erteilten Berechtigungen an. Mögliche Einstellungen sind `read`, `readacl`, `writeacl` oder `full`.
- *Grantee_Type* — Gibt an, wie der Empfänger identifiziert werden kann. Mögliche Einstellungen sind `uri`, `emailaddress` oder `id`.
- *Grantee_ID* — Spezifiziert den Empfänger anhand von *Grantee_Type*.
 - `uri`— Die der Gruppe. URI Weitere Informationen finden Sie unter [Wer ist ein Berechtigungsempfänger?](#)
 - `emailaddress` – Die E-Mail-Adresse des Kontos.
 - `id` – Die kanonische ID des Kontos.

Weitere Informationen zur Amazon-S3-Zugriffssteuerung finden Sie unter [Zugriffssteuerung](#).

Im folgenden Beispiel wird ein Objekt in einen Bucket kopiert. Es werden read-Berechtigungen für das Objekt für alle erteilt. Das Konto, das zu full gehört, erhält read-Berechtigungen (readacl, writeacl und user@example.com).

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://  
acs.amazonaws.com/groups/global/AllUsers full=emailaddress=user@example.com
```

Sie können auch eine nicht standardmäßige Speicherklasse (REDUCED_REDUNDANCY oder STANDARD_IA) für Objekte angeben, die Sie in Amazon S3 hochladen. Verwenden Sie dazu die Option `--storage-class`.

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --storage-class REDUCED_REDUNDANCY
```

recursive

Wenn Sie diese Option verwenden, wird der Befehl für alle Dateien oder Objekte unter dem angegebenen Verzeichnis oder Präfix ausgeführt. Das folgende Beispiel löscht `s3://amzn-s3-demo-bucket/path` und seinen gesamten Inhalt.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/path --recursive
```

Ressourcen

AWS CLI Referenz:

- [aws s3](#)
- [aws s3 cp](#)
- [aws s3 mb](#)
- [aws s3 mv](#)
- [aws s3 ls](#)
- [aws s3 rb](#)
- [aws s3 rm](#)
- [aws s3 sync](#)

Service-Referenz:

- [Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon S3 S3-Benutzerhandbuch
- [Arbeiten mit Amazon S3 S3-Objekten](#) im Amazon S3 S3-Benutzerhandbuch
- [Schlüssel mit einem Präfix und einem Trennzeichen hierarchisch auflisten](#) im Amazon S3 S3-Benutzerhandbuch
- [Berechnen Sie mehrteilige Uploads in einen S3-Bucket mithilfe der AWS SDK for .NET \(Low-Level\) im Amazon S3 S3-Benutzerhandbuch](#) ab

Verwenden von API -Level (s3api) -Befehlen in AWS CLI

Die Befehle API auf -Ebene (im `s3api` Befehlssatz enthalten) bieten direkten Zugriff auf den Amazon Simple Storage Service (Amazon S3) und ermöglichen einige Operationen APIs, die in den `s3` High-Level-Befehlen nicht verfügbar sind. Diese Befehle entsprechen den anderen AWS Diensten, die Zugriff auf die Funktionen API der Dienste auf der Ebene bieten. Weitere Informationen zu den Befehlen `s3` finden Sie unter [Verwenden von Befehlen auf hoher Ebene \(s3\) in der AWS CLI](#).

Dieses Thema enthält Beispiele, die zeigen, wie die Befehle auf niedrigerer Ebene verwendet werden, die Amazon S3 zugeordnet sind. APIs Darüber hinaus finden Sie Beispiele für jeden API S3-Befehl im `s3api` Abschnitt des [AWS CLI Referenzhandbuchs AWS CLI](#)

Themen

- [Voraussetzungen](#)
- [Wenden Sie ein benutzerdefiniertes an ACL](#)
- [Konfigurieren einer Protokollierungsrichtlinie](#)
- [Ressourcen](#)

Voraussetzungen

Zur Ausführung von `s3api`-Befehlen ist Folgendes erforderlich:

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Das von Ihnen verwendete Profil muss über Berechtigungen verfügen, die die anhand der Beispiele ausgeführten AWS Operationen zulassen.
- Sie müssen diese Amazon-S3-Begriffe verstehen:
 - Bucket – Ein Amazon-S3-Ordner der obersten Ebene.

- Präfix – Ein Amazon-S3-Ordner in einem Bucket.
- Objekt – Jedes Element, das in einem Amazon-S3-Bucket gehostet wird.

Wenden Sie ein benutzerdefiniertes ACL

Mit Befehlen auf hoher Ebene können Sie die `--acl` Option verwenden, um vordefinierte Zugriffskontrolllisten (ACLs) auf Amazon S3 S3-Objekte anzuwenden. Sie können diesen Befehl jedoch nicht verwenden, um Einstellungen für den gesamten Bucket festzulegen. ACLs Sie können dies jedoch mit dem Befehl [put-bucket-acl](#) API -level tun.

Das folgende Beispiel zeigt, wie Sie zwei AWS Benutzern (`user1@example.com` und `user2@example.com`) Vollzugriff und allen Benutzern Leseberechtigungen gewähren. Der Bezeichner für „alle“ stammt aus einem SpecialURI, das Sie als Parameter übergeben.

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-control  
'emailaddress="user1@example.com",emailaddress="user2@example.com"' --grant-read  
'uri="http://acs.amazonaws.com/groups/global/AllUsers"'
```

Einzelheiten zur Erstellung von finden Sie unter [PUTBucket acl](#) in der Amazon Simple Storage Service API Reference. ACLs Die `s3api` ACL Befehle inCLI, z. B. `put-bucket-acl`, verwenden dieselbe [Kurzschreibweise für Argumente](#).

Konfigurieren einer Protokollierungsrichtlinie

Der API Befehl `put-bucket-logging` konfiguriert eine Bucket-Logging-Richtlinie.

Im folgenden Beispiel erhält der AWS Benutzer `user@example.com` die volle Kontrolle über die Protokolldateien, und alle Benutzer haben Lesezugriff darauf. Beachten Sie, dass der `put-bucket-acl` Befehl auch erforderlich ist, um dem Amazon S3 S3-Protokollbereitstellungssystem (angegeben durch `aURI`) die Berechtigungen zu erteilen, die zum Lesen und Schreiben der Protokolle in den Bucket erforderlich sind.

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-read-acp 'URI="http://  
acs.amazonaws.com/groups/s3/LogDelivery"' --grant-write 'URI="http://acs.amazonaws.com/  
groups/s3/LogDelivery"'
```

```
$ aws s3api put-bucket-logging --bucket amzn-s3-demo-bucket --bucket-logging-status  
file://logging.json
```

Die Datei `logging.json` aus dem vorherigen Befehl hat folgenden Inhalt.

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-bucket",
    "TargetPrefix": "amzn-s3-demo-bucketLogs/",
    "TargetGrants": [
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "Type": "Group",
          "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
        },
        "Permission": "READ"
      }
    ]
  }
}
```

Ressourcen

AWS CLI Referenz:

- [aws s3api](#)
- [aws s3api put-bucket-acl](#)
- [aws s3api put-bucket-logging](#)

Service-Referenz:

- [Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon S3 S3-Benutzerhandbuch
- [Arbeiten mit Amazon S3 S3-Objekten](#) im Amazon S3 S3-Benutzerhandbuch
- [Schlüssel mit einem Präfix und einem Trennzeichen hierarchisch auflisten](#) im Amazon S3 S3-Benutzerhandbuch
- [Brechen Sie mehrteilige Uploads in einen S3-Bucket mithilfe der AWS SDK for .NET \(Low-Level\) im Amazon S3 S3-Benutzerhandbuch](#) ab

Skriptbeispiel für den Amazon S3 S3-Bucket-Lebenszyklus in der AWS CLI

In diesem Thema wird ein Beispiel der Bash-Skriptsprache für Amazon-S3-Bucket-Lebenszyklusoperationen mithilfe der AWS Command Line Interface (AWS CLI) verwendet. Dieses Skriptsprachebeispiel verwendet den Befehlssatz [aws s3api](#). Shell-Skripte sind Programme, die in einer Befehlszeilenschnittstelle ausgeführt werden sollen.

Themen

- [Bevor Sie beginnen](#)
- [Über das Beispiel](#)
- [Dateien](#)
- [Referenzen](#)

Bevor Sie beginnen

Bevor Sie eines der folgenden Beispiele ausführen können, müssen die folgenden Schritte abgeschlossen werden.

- Installieren und Konfigurieren der AWS CLI. Weitere Informationen erhalten Sie unter [Installieren Sie das AWS CLI](#) und [Authentifizierungs- und Zugangsdaten für AWS CLI](#).
- Das Profil, das Sie verwenden, muss über Berechtigungen verfügen, die die in den Beispielen ausgeführten AWS Operationen zulassen.
- Es hat sich AWS bewährt, diesem Code die geringsten Rechte oder nur die zum Ausführen einer Aufgabe erforderlichen Berechtigungen zu gewähren. Weitere Informationen finden Sie unter [Grant Least Privilege](#) im IAMBenutzerhandbuch.
- Dieser Code wurde nicht in allen AWS Regionen getestet. Einige AWS Dienste sind nur in bestimmten Regionen verfügbar. Weitere Informationen finden Sie unter [Service-Endpunkte und Kontingente](#) im allgemeinen AWS -Referenzhandbuch.
- Wenn Sie diesen Code ausführen, kann dies zu Gebühren für Ihr AWS Konto führen. Es liegt in Ihrer Verantwortung sicherzustellen, dass alle durch dieses Skript erstellten Ressourcen entfernt werden, wenn Sie mit ihnen fertig sind.

Der Amazon-S3-Service verwendet die folgenden Begriffe:

- Bucket – Ein Amazon-S3-Ordner der obersten Ebene.

- Präfix – Ein Amazon-S3-Ordner in einem Bucket.
- Objekt – Jedes Element, das in einem Amazon-S3-Bucket gehostet wird.

Über das Beispiel

Dieses Beispiel zeigt, wie Sie mit einigen grundlegenden Amazon-S3-Operationen interagieren, indem Sie eine Reihe von Funktionen in Shell-Skriptdateien verwenden. Die Funktionen befinden sich in der Shell-Skriptdatei namens `bucket-operations.sh`. Sie können diese Funktionen in einer anderen Datei aufrufen. Jede Skriptdatei enthält Kommentare, die jede der Funktionen beschreiben.

Um die Zwischenergebnisse jedes Schritts anzuzeigen, führen Sie das Skript mit einem `-i`-Parameter aus. Sie können den aktuellen Status des Buckets oder seinen Inhalt mithilfe der Amazon-S3-Konsole anzeigen. Das Skript fährt nur dann mit dem nächsten Schritt fort, wenn Sie an der Eingabeaufforderung die Eingabetaste drücken.

Das vollständige Beispiel und die herunterladbaren Skriptdateien finden Sie unter [Amazon S3 Bucket Lifecycle Operations](#) im AWS Code Examples Repository unter GitHub.

Dateien

Das Beispiel enthält die folgenden Dateien:

`bucket-operations.sh`

Diese Hauptskriptdatei kann aus einer anderen Datei bezogen werden. Sie umfasst Funktionen, die die folgenden Aufgaben ausführen:

- Erstellen eines Buckets und Überprüfen, ob er vorhanden ist
- Kopieren einer Datei vom lokalen Computer in einen Bucket
- Kopieren einer Datei von einem Bucket-Speicherort an einen anderen Bucket-Speicherort
- Auflisten der Inhalte eines Buckets
- Löschen einer Datei aus einem Bucket
- Löschen eines Buckets

Sehen Sie sich den Code für [bucket-operations.sh](#) on an GitHub.

test-bucket-operations.sh

Die Shell-Skriptdatei `test-bucket-operations.sh` zeigt, wie die Funktionen aufgerufen werden, indem die Datei `bucket-operations.sh` bezogen und jede der Funktionen aufgerufen wird. Nach dem Aufrufen von Funktionen entfernt das Testskript alle Ressourcen, die es erstellt hat.

Sehen Sie sich den Code für [test-bucket-operations.sh](#) on an. GitHub

awsdocs-general.sh

Die Skriptdatei `awsdocs-general.sh` enthält allgemeine Funktionen, die in erweiterten Code-Beispielen für die AWS CLI verwendet werden.

Sehen Sie sich den Code für [awsdocs-general.sh](#) on an GitHub.

Referenzen

AWS CLI Referenz:

- [aws s3api](#)
- [aws s3api create-bucket](#)
- [aws s3api copy-object](#)
- [aws s3api delete-bucket](#)
- [aws s3api delete-object](#)
- [aws s3api head-bucket](#)
- [aws s3api list-objects](#)
- [aws s3api put-object](#)

Andere Referenz:

- [Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon S3 S3-Benutzerhandbuch
- [Arbeiten mit Amazon S3 S3-Objekten](#) im Amazon S3 S3-Benutzerhandbuch
- Um AWS CLI Codebeispiele anzusehen und Beiträge dazu zu AWS SDK leisten, finden Sie im [AWS Codebeispiel-Repository](#) unter GitHub.

Zugriff auf Amazon SNS in der AWS CLI

Sie können auf die Funktionen von Amazon Simple Notification Service (AmazonSNS) zugreifen, indem Sie AWS Command Line Interface (AWS CLI) verwenden. Verwenden Sie den folgenden Befehl `sns`, um die AWS CLI Befehle für Amazon aufzulisten.

```
aws sns help
```

Bevor Sie Befehle ausführen, richten Sie die Standardanmeldeinformationen ein. Weitere Informationen finden Sie unter [Konfiguration der Einstellungen für den AWS CLI](#).

Dieses Thema zeigt Beispiele für AWS CLI Befehle, mit denen allgemeine Aufgaben für Amazon ausgeführt werden können.

Themen

- [Erstellen eines Themas](#)
- [Abonnieren eines Themas](#)
- [Veröffentlichung für ein Thema](#)
- [Abbestellen eines Themas](#)
- [Löschen eines Themas](#)

Erstellen eines Themas

Zum Erstellen eines Themas verwenden Sie den Befehl `sns create-topic` und geben den Namen an, den Sie dem Thema zuweisen möchten.

```
$ aws sns create-topic --name my-topic
{
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

Notieren Sie sich den `TopicArn` der Antwort, den Sie später zum Veröffentlichern einer Nachricht benötigen.

Abonnieren eines Themas

Verwenden Sie zum Abonnieren eines Themas den Befehl `sns subscribe`.

Das folgende Beispiel gibt das email-Protokoll und eine E-Mail-Adresse für den notification-endpoint an.

```
$ aws sns subscribe --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
protocol email --notification-endpoint saanvi@example.com
{
  "SubscriptionArn": "pending confirmation"
}
```

AWS sendet sofort eine Bestätigungsnachricht per E-Mail an die Adresse, die Sie im subscribe Befehl angegeben haben. Die E-Mail-Nachricht enthält den folgenden Text.

```
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:123456789012:my-topic
To confirm this subscription, click or visit the following link (If this was in error
no action is necessary):
Confirm subscription
```

Nachdem der Empfänger auf den Link für Confirm subscription (Abonnement bestätigen) geklickt hat, zeigt der Browser des Empfängers eine Benachrichtigung mit Informationen ähnlich der folgenden an.

```
Subscription confirmed!

You have subscribed saanvi@example.com to the topic:my-topic.

Your subscription's id is:
arn:aws:sns:us-west-2:123456789012:my-topic:1328f057-de93-4c15-512e-8bb22EXAMPLE

If it was not your intention to subscribe, click here to unsubscribe.
```

Veröffentlichung für ein Thema

Zum Senden einer Nachricht an alle Abonnenten eines Themas verwenden Sie den Befehl [sns publish](#).

Das folgende Beispiel sendet die Meldung „Hallo Welt!“ an alle Abonnenten des angegebenen Themas.

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
message "Hello World!"
{
```

```
"MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"  
}
```

AWS Sendet in diesem Beispiel eine E-Mail-Nachricht mit dem Text „Hello World!“
ansaanvi@example.com.

Abbestellen eines Themas

Um sich von einem Thema abzumelden und keine Nachrichten mehr zu diesem Thema zu erhalten, verwenden Sie den [sns unsubscribe](#) Befehl und geben Sie das ARN Thema an, für das Sie sich abmelden möchten.

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-  
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

Um zu überprüfen, ob Sie sich erfolgreich abgemeldet haben, verwenden Sie den [sns list-subscriptions](#) Befehl, um zu bestätigen, dass das nicht ARN mehr in der Liste angezeigt wird.

```
$ aws sns list-subscriptions
```

Löschen eines Themas

Mit dem Befehl [sns delete-topic](#) können Sie ein Thema löschen.

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

Um zu überprüfen, ob das Thema AWS erfolgreich gelöscht wurde, verwenden Sie den [sns list-topics](#) Befehl, um zu bestätigen, dass das Thema nicht mehr in der Liste erscheint.

```
$ aws sns list-topics
```

AWS CLI Befehlsbeispiele

Die Codebeispiele in diesem Thema zeigen Ihnen, wie Sie AWS Command Line Interface with verwenden AWS.

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wichtigsten Operationen innerhalb eines Dienstes ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Dienstes oder in Kombination mit anderen aufrufen AWS-Services.

Services

- [ACMBeispiele mit AWS CLI](#)
- [APIGateway-Beispiele mit AWS CLI](#)
- [APIGateway HTTP und WebSocket API Beispiele mit AWS CLI](#)
- [APIAPIGateway-Management-Beispiele mit AWS CLI](#)
- [App Mesh Mesh-Beispiele mit AWS CLI](#)
- [App Runner-Beispiele mit AWS CLI](#)
- [AWS AppConfig Beispiele mit AWS CLI](#)
- [Beispiele für Application Auto Scaling mit AWS CLI](#)
- [Beispiele für den Application Discovery Service mit AWS CLI](#)
- [AppRegistry Beispiele mit AWS CLI](#)
- [Athena-Beispiele mit AWS CLI](#)
- [Auto Scaling Scaling-Beispiele mit AWS CLI](#)
- [Beispiele für Auto Scaling-Pläne mit AWS CLI](#)
- [AWS Backup Beispiele mit AWS CLI](#)
- [AWS Batch Beispiele mit AWS CLI](#)
- [AWS Budgets Beispiele mit AWS CLI](#)
- [Amazon Chime Chime-Beispiele mit AWS CLI](#)
- [Cloud API Control-Beispiele mit AWS CLI](#)
- [AWS Cloud Map Beispiele mit AWS CLI](#)
- [AWS Cloud9 Beispiele mit AWS CLI](#)
- [AWS CloudFormation Beispiele mit AWS CLI](#)
- [CloudFront Beispiele mit AWS CLI](#)
- [CloudSearch Amazon-Beispiele mit AWS CLI](#)

- [CloudTrail Beispiele mit AWS CLI](#)
- [CloudWatch Beispiele mit AWS CLI](#)
- [CloudWatch Log-Beispiele mit AWS CLI](#)
- [CloudWatch Beispiele für Netzwerküberwachung mit AWS CLI](#)
- [CodeArtifact Beispiele mit AWS CLI](#)
- [CodeBuild Beispiele mit AWS CLI](#)
- [CodeCommit Beispiele mit AWS CLI](#)
- [CodeDeploy Beispiele mit AWS CLI](#)
- [CodeGuru Beispiele für Gutachter mit AWS CLI](#)
- [CodePipeline Beispiele mit AWS CLI](#)
- [AWS CodeStar Beispiele für Benachrichtigungen mit AWS CLI](#)
- [CodeConnections Beispiele mit AWS CLI](#)
- [Beispiele für Amazon Cognito Identity mit AWS CLI](#)
- [Beispiele für Amazon Cognito Identity Provider mit AWS CLI](#)
- [Amazon Comprehend Comprehend-Beispiele mit AWS CLI](#)
- [Beispiele von Amazon Comprehend Medical mit AWS CLI](#)
- [AWS Config Beispiele mit AWS CLI](#)
- [Amazon Connect Connect-Beispiele mit AWS CLI](#)
- [AWS Cost and Usage Report Beispiele mit AWS CLI](#)
- [Beispiele für den Cost Explorer Explorer-Service mit AWS CLI](#)
- [Firehose-Beispiele mit AWS CLI](#)
- [Amazon Data Lifecycle Manager Manager-Beispiele mit AWS CLI](#)
- [AWS Data Pipeline Beispiele mit AWS CLI](#)
- [DataSync Beispiele mit AWS CLI](#)
- [DAXBeispiele mit AWS CLI](#)
- [Beispiele für Detective mit AWS CLI](#)
- [Beispiele für Device Farm mit AWS CLI](#)
- [AWS Direct Connect Beispiele mit AWS CLI](#)
- [AWS Directory Service Beispiele mit AWS CLI](#)

- [AWS DMS Beispiele mit AWS CLI](#)
- [Amazon DocumentDB DocumentDB-Beispiele mit AWS CLI](#)
- [DynamoDB-Beispiele mit AWS CLI](#)
- [Beispiele für DynamoDB Streams mit AWS CLI](#)
- [EC2Amazon-Beispiele mit AWS CLI](#)
- [Amazon EC2 Instance Connect-Beispiele mit AWS CLI](#)
- [ECRAmazon-Beispiele mit AWS CLI](#)
- [Amazon ECR Public-Beispiele mit AWS CLI](#)
- [ECSAmazon-Beispiele mit AWS CLI](#)
- [EFSAmazon-Beispiele mit AWS CLI](#)
- [EKSAmazon-Beispiele mit AWS CLI](#)
- [Elastic Beanstalk Beanstalk-Beispiele mit AWS CLI](#)
- [Elastic Load Balancing — Version 1, Beispiele mit AWS CLI](#)
- [Elastic Load Balancing — Version 2-Beispiele mit AWS CLI](#)
- [Elastic Transcoder Transcoder-Beispiele mit AWS CLI](#)
- [ElastiCache Beispiele mit AWS CLI](#)
- [MediaStore Beispiele mit AWS CLI](#)
- [EMRAmazon-Beispiele mit AWS CLI](#)
- [Amazon EMR an EKS Beispielen mit AWS CLI](#)
- [EventBridge Beispiele mit AWS CLI](#)
- [Beispiele für Firewall Manager mit AWS CLI](#)
- [AWS FIS Beispiele mit AWS CLI](#)
- [GameLift Amazon-Beispiele mit AWS CLI](#)
- [Global Accelerator-Beispiele mit AWS CLI](#)
- [AWS Glue Beispiele mit AWS CLI](#)
- [GuardDuty Beispiele mit AWS CLI](#)
- [AWS Health Beispiele mit AWS CLI](#)
- [HealthImaging Beispiele mit AWS CLI](#)
- [HealthLake Beispiele mit AWS CLI](#)

- [HealthOmics Beispiele mit AWS CLI](#)
- [IAMBeispiele mit AWS CLI](#)
- [IAMAccess Analyzer-Beispiele mit AWS CLI](#)
- [Image Builder Builder-Beispiele mit AWS CLI](#)
- [Beispiele für Incident Manager mit AWS CLI](#)
- [Beispiele für Incident Manager-Kontakte mit AWS CLI](#)
- [Amazon Inspector Inspector-Beispiele mit AWS CLI](#)
- [AWS IoT Beispiele mit AWS CLI](#)
- [AWS IoT 1-Click Gerätebeispiele mit AWS CLI](#)
- [AWS IoT 1-Click Beispiele für Projekte mit AWS CLI](#)
- [AWS IoT Analytics Beispiele mit AWS CLI](#)
- [Device Advisor-Beispiele mit AWS CLI](#)
- [AWS IoT data Beispiele mit AWS CLI](#)
- [AWS IoT Events Beispiele mit AWS CLI](#)
- [AWS IoT Events-Data Beispiele mit AWS CLI](#)
- [AWS IoT Greengrass Beispiele mit AWS CLI](#)
- [AWS IoT Greengrass V2 Beispiele mit AWS CLI](#)
- [AWS IoT Jobs SDK release Beispiele mit AWS CLI](#)
- [AWS IoT SiteWise Beispiele mit AWS CLI](#)
- [AWS IoT Things Graph Beispiele mit AWS CLI](#)
- [AWS IoT Wireless Beispiele mit AWS CLI](#)
- [IVSAmazon-Beispiele mit AWS CLI](#)
- [Amazon IVS Chat-Beispiele mit AWS CLI](#)
- [Beispiele für Amazon IVS Real-Time Streaming mit AWS CLI](#)
- [Amazon Kendra Beispiele mit AWS CLI](#)
- [Kinesis-Beispiele mit AWS CLI](#)
- [AWS KMS Beispiele mit AWS CLI](#)
- [Beispiele für Lake Formation mit AWS CLI](#)
- [Lambda-Beispiele mit AWS CLI](#)

- [License Manager Manager-Beispiele mit AWS CLI](#)
- [Lightsail-Beispiele mit AWS CLI](#)
- [Macie-Beispiele mit AWS CLI](#)
- [Amazon Managed Grafana-Beispiele mit AWS CLI](#)
- [MediaConnect Beispiele mit AWS CLI](#)
- [MediaConvert Beispiele mit AWS CLI](#)
- [MediaLive Beispiele mit AWS CLI](#)
- [MediaPackage Beispiele mit AWS CLI](#)
- [MediaPackage VODBeispiele mit AWS CLI](#)
- [MediaStore Beispiele für Datenebene mit AWS CLI](#)
- [MediaTailor Beispiele mit AWS CLI](#)
- [MemoryDB-Beispiele mit AWS CLI](#)
- [MSKAmazon-Beispiele mit AWS CLI](#)
- [Network Manager-Beispiele mit AWS CLI](#)
- [Nimble Studio-Beispiele mit AWS CLI](#)
- [OpenSearch Servicebeispiele mit AWS CLI](#)
- [AWS OpsWorks Beispiele mit AWS CLI](#)
- [AWS OpsWorks CM Beispiele mit AWS CLI](#)
- [Beispiele für Organizations, die AWS CLI](#)
- [AWS Outposts Beispiele mit AWS CLI](#)
- [AWS Payment Cryptography Beispiele mit AWS CLI](#)
- [AWS Payment Cryptography Beispiele für Datenebene mit AWS CLI](#)
- [Amazon Pinpoint Pinpoint-Beispiele mit AWS CLI](#)
- [Beispiele für Amazon Polly mit AWS CLI](#)
- [AWS-Preisliste Beispiele mit AWS CLI](#)
- [AWS Private CA Beispiele mit AWS CLI](#)
- [AWS Proton Beispiele mit AWS CLI](#)
- [QLDBBeispiele mit AWS CLI](#)
- [RDSAmazon-Beispiele mit AWS CLI](#)

- [Beispiele für Amazon RDS Data Service mit AWS CLI](#)
- [Beispiele für Amazon RDS Performance Insights mit AWS CLI](#)
- [Amazon Redshift Redshift-Beispiele mit AWS CLI](#)
- [Amazon Rekognition Rekognition-Beispiele mit AWS CLI](#)
- [AWS RAM Beispiele mit AWS CLI](#)
- [Resource Explorer-Beispiele mit AWS CLI](#)
- [Beispiele für Resource Groups mit AWS CLI](#)
- [API-Beispiele für das Tagging von Resource Groups mithilfe von AWS CLI](#)
- [AWS RoboMaker Beispiele mit AWS CLI](#)
- [Route 53-Beispiele mit AWS CLI](#)
- [Beispiele für die Route 53-Domainregistrierung mit AWS CLI](#)
- [Beispiele für Route 53 53-Profile mit AWS CLI](#)
- [Beispiele für Route 53 Resolver mit AWS CLI](#)
- [Amazon S3 S3-Beispiele mit AWS CLI](#)
- [Beispiele für Amazon S3 Control mit AWS CLI](#)
- [S3 Glacier-Beispiele mit AWS CLI](#)
- [Secrets Manager Manager-Beispiele mit AWS CLI](#)
- [Security Hub Hub-Beispiele mit AWS CLI](#)
- [Security Lake-Beispiele mit AWS CLI](#)
- [AWS Serverless Application Repository Beispiele mit AWS CLI](#)
- [Servicekatalog-Beispiele mit AWS CLI](#)
- [Beispiele für Service Quotas mit AWS CLI](#)
- [SESA-Beispiele mit AWS CLI](#)
- [Shield-Beispiele mit AWS CLI](#)
- [Beispiele für Unterzeichner mit AWS CLI](#)
- [Schneeball-Beispiele mit AWS CLI](#)
- [SNS-Beispiele mit AWS CLI](#)
- [SQS-Beispiele mit AWS CLI](#)
- [Storage Gateway Gateway-Beispiele mit AWS CLI](#)

- [AWS STS Beispiele mit AWS CLI](#)
- [AWS Support Beispiele mit AWS CLI](#)
- [SWFAmazon-Beispiele mit AWS CLI](#)
- [Systems Manager Manager-Beispiele mit AWS CLI](#)
- [Amazon Textract Textract-Beispiele mit AWS CLI](#)
- [Amazon Transcribe Transcribe-Beispiele mit AWS CLI](#)
- [Amazon Translate Translate-Beispiele mit AWS CLI](#)
- [Trusted Advisor Beispiele mit AWS CLI](#)
- [Beispiele für verifizierte Berechtigungen mit AWS CLI](#)
- [VPCBeispiele für Gitter unter Verwendung AWS CLI](#)
- [AWS WAF Classic Beispiele mit AWS CLI](#)
- [AWS WAF Classic Regional Beispiele mit AWS CLI](#)
- [AWS WAFV2 Beispiele mit AWS CLI](#)
- [WorkDocs Amazon-Beispiele mit AWS CLI](#)
- [WorkMail Amazon-Beispiele mit AWS CLI](#)
- [Amazon WorkMail Message Flow-Beispiele mit AWS CLI](#)
- [WorkSpaces Beispiele mit AWS CLI](#)
- [Röntgenbeispiele mit AWS CLI](#)

ACMBeispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren ACM.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-certificate

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-certificate`.

AWS CLI

Um einem vorhandenen ACM Zertifikat Tags hinzuzufügen

Der folgende `add-tags-to-certificate` Befehl fügt dem angegebenen Zertifikat zwei Tags hinzu. Verwenden Sie ein Leerzeichen, um mehrere Tags voneinander zu trennen:

```
aws acm add-tags-to-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- API-Einheiten finden Sie [AddTagsToCertificate](#) in der AWS CLI Befehlsreferenz.

delete-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-certificate`.

AWS CLI

Um ein ACM Zertifikat aus Ihrem Konto zu löschen

Der folgende `delete-certificate` Befehl löscht das Zertifikat mit dem angegebenen ARN Wert:

```
aws acm delete-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- API-Einheiten finden Sie [DeleteCertificate](#) in der AWS CLI Befehlsreferenz.

describe-certificate

Das folgende Codebeispiel zeigt die Verwendung `describe-certificate`.

AWS CLI

Um die in einem ACM Zertifikat enthaltenen Felder abzurufen

Der folgende `describe-certificate` Befehl ruft alle Felder für das Zertifikat mit den angegebenen ARN Werten ab:

```
aws acm describe-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Es wird eine Ausgabe ähnlich der folgenden angezeigt:

```
{
  "Certificate": {
    "CertificateArn":
"arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
    "CreatedAt": 1446835267.0,
    "DomainName": "www.example.com",
    "DomainValidationOptions": [
      {
        "DomainName": "www.example.com",
        "ValidationDomain": "www.example.com",
        "ValidationEmails": [
          "hostmaster@example.com",
          "admin@example.com",
          "owner@example.com.whoisprivacyservice.org",
          "tech@example.com.whoisprivacyservice.org",
          "admin@example.com.whoisprivacyservice.org",
          "postmaster@example.com",
          "webmaster@example.com",
          "administrator@example.com"
        ]
      }
    ],
  },
  {
    "DomainName": "www.example.net",
    "ValidationDomain": "www.example.net",
    "ValidationEmails": [
      "postmaster@example.net",
      "admin@example.net",
      "owner@example.net.whoisprivacyservice.org",
      "tech@example.net.whoisprivacyservice.org",
      "admin@example.net.whoisprivacyservice.org",
      "hostmaster@example.net",
      "administrator@example.net",
      "webmaster@example.net"
    ]
  }
}
```

```

    ],
    "InUseBy": [],
    "IssuedAt": 1446835815.0,
    "Issuer": "Amazon",
    "KeyAlgorithm": "RSA-2048",
    "NotAfter": 1478433600.0,
    "NotBefore": 1446768000.0,
    "Serial": "0f:ac:b0:a3:8d:ea:65:52:2d:7d:01:3a:39:36:db:d6",
    "SignatureAlgorithm": "SHA256WITHRSA",
    "Status": "ISSUED",
    "Subject": "CN=www.example.com",
    "SubjectAlternativeNames": [
      "www.example.com",
      "www.example.net"
    ]
  }
}

```

- API-Einheiten finden Sie [DescribeCertificate](#) in der AWS CLI Befehlsreferenz.

export-certificate

Das folgende Codebeispiel zeigt die Verwendung `export-certificate`.

AWS CLI

Um ein privates Zertifikat zu exportieren, das von einer privaten Zertifizierungsstelle ausgestellt wurde.

Der folgende `export-certificate` Befehl exportiert ein privates Zertifikat, eine Zertifikatskette und einen privaten Schlüssel auf Ihr Display:

```

aws acm export-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file

```

Verwenden Sie den folgenden Befehl, um das Zertifikat, die Kette und den privaten Schlüssel in eine lokale Datei zu exportieren:

```

aws acm export-certificate --certificate-
arn arn:aws:acm:region:sccount:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file > c:\temp\export.txt

```

- API-Einheiten finden Sie [ExportCertificate](#) unter AWS CLI Befehlsreferenz.

get-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-certificate`.

AWS CLI

Um ein ACM Zertifikat abzurufen

Der folgende `get-certificate` Befehl ruft das Zertifikat für das angegebene Zertifikat ARN und die Zertifikatskette ab:

```
aws acm get-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Es wird eine Ausgabe ähnlich der folgenden angezeigt:

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCcAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYW1mXzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYW1mXzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCcAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYW1mXzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
```

```

b2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCcAfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxGzAJBgNVBAGTAldBMRAwDgYDZDQHEwDZWF0dGx1MQ8wDQYDZDQKEwZBbWF6
b24xZDASBgNVBASTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxGzAJBgNVBAGTAldBMRAwDgYD
VQHEwDZWF0dGx1MQ8wDQYDZDQKEwZBbWF6b24xZDASBgNVBASTC0lBTSBDb25z
b2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCcAfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxGzAJBgNVBAGTAldBMRAwDgYDZDQHEwDZWF0dGx1MQ8wDQYDZDQKEwZBbWF6
b24xZDASBgNVBASTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxGzAJBgNVBAGTAldBMRAwDgYD
VQHEwDZWF0dGx1MQ8wDQYDZDQKEwZBbWF6b24xZDASBgNVBASTC0lBTSBDb25z
b2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
}

```

- API-Einheiten finden Sie [GetCertificate](#) in der AWS CLI Befehlsreferenz.

import-certificate

Das folgende Codebeispiel zeigt die Verwendung `import-certificate`.

AWS CLI

Um ein Zertifikat zu importieren ACM.

Der folgende `import-certificate` Befehl importiert ein Zertifikat in ACM. Ersetzen Sie die Dateinamen durch Ihre eigenen:

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- API-Einheiten finden Sie [ImportCertificate](#) in der AWS CLI Befehlsreferenz.

list-certificates

Das folgende Codebeispiel zeigt die Verwendung `list-certificates`.

AWS CLI

Um die ACM Zertifikate für ein AWS Konto aufzulisten

Der folgende `list-certificates` Befehl listet ARNs der Zertifikate in Ihrem Konto auf:

```
aws acm list-certificates
```

Der vorhergehende Befehl erzeugt eine Ausgabe, die der folgenden ähnelt:

```
{
  "CertificateSummaryList": [
    {
      "CertificateArn":
"arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
      "DomainName": "www.example.com"
    },
    {
      "CertificateArn": "arn:aws:acm:region:account:certificate/aaaaaaaa-bbbb-
cccc-dddd-eeeeeeeeeeee",
      "DomainName": "www.example.net"
    }
  ]
}
```



```
]
}
```

Sie können entscheiden, wie viele Zertifikate Sie bei jedem Anruf anzeigen möchten `list-certificates`. Wenn Sie beispielsweise über vier Zertifikate verfügen und nicht mehr als zwei gleichzeitig anzeigen möchten, legen Sie das `max-items` Argument wie im folgenden Beispiel auf 2 fest:

```
aws acm list-certificates --max-items 2
```

Zwei Zertifikate ARNs und ein `NextToken` Wert werden angezeigt:

```
"CertificateSummaryList": [
  {
    "CertificateArn": "arn:aws:acm:region:account: \
      certificate/12345678-1234-1234-1234-123456789012",
    "DomainName": "www.example.com"
  },
  {
    "CertificateArn": "arn:aws:acm:region:account: \
      certificate/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "DomainName": "www.example.net"
  }
],
"NextToken": "9f4d9f69-275a-41fe-b58e-2b837bd9ba48"
```

Um die nächsten beiden Zertifikate in Ihrem Konto anzuzeigen, legen Sie bei Ihrem nächsten Anruf diesen `NextToken` Wert fest:

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-
b58e-2b837bd9ba48
```

Sie können Ihre Ausgabe mithilfe des `certificate-statuses` Arguments filtern. Der folgende Befehl zeigt Zertifikate an, die `VALIDATION` den Status `PENDING` _ haben:

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

Sie können Ihre Ausgabe auch mithilfe des `includes` Arguments filtern. Mit dem folgenden Befehl werden Zertifikate angezeigt, die nach den folgenden Eigenschaften gefiltert wurden. Die anzuzeigenden Zertifikate:

- Specify that the RSA algorithm and a 2048 bit key are used to generate key pairs.
- Contain a Key Usage extension that specifies that the certificates can be used to create digital signatures.
- Contain an Extended Key Usage extension that specifies that the certificates can be used for code signing.

```
aws acm list-certificates --max-items 10 --includes
extendedKeyUsage=CODE_SIGNING,keyUsage=DIGITAL_SIGNATURE,keyTypes=RSA_2048
```

- API-Einheiten finden Sie [ListCertificates](#) in der AWS CLI Befehlsreferenz.

list-tags-for-certificate

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-certificate`.

AWS CLI

Um die auf ein ACM Zertifikat angewendeten Tags aufzulisten

Der folgende `list-tags-for-certificate` Befehl listet die Tags auf, die auf ein Zertifikat in Ihrem Konto angewendet wurden:

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Der vorhergehende Befehl erzeugt eine Ausgabe, die der folgenden ähnelt:

```
{
  "Tags": [
    {
      "Value": "Website",
      "Key": "Purpose"
    },
    {
      "Value": "Alice",
      "Key": "Admin"
    }
  ]
}
```

- API-Einheiten finden Sie [ListTagsForCertificate](#) in der AWS CLI Befehlsreferenz.

remove-tags-from-certificate

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-certificate`.

AWS CLI

Um ein Tag aus einem ACM Zertifikat zu entfernen

Der folgende `remove-tags-from-certificate` Befehl entfernt zwei Tags aus dem angegebenen Zertifikat. Verwenden Sie ein Leerzeichen, um mehrere Tags voneinander zu trennen:

```
aws acm remove-tags-from-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- API-Einheiten finden Sie [RemoveTagsFromCertificate](#) in der AWS CLI Befehlsreferenz.

request-certificate

Das folgende Codebeispiel zeigt die Verwendung `request-certificate`.

AWS CLI

Um ein neues ACM Zertifikat anzufordern

Mit dem folgenden `request-certificate` Befehl wird mithilfe einer Validierung ein neues Zertifikat für die Domain `www.example.com` angefordert. DNS:

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

Sie können ein Idempotenz-Token eingeben, um zwischen Aufrufen zu unterscheiden: `request-certificate`

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --idempotency-token 91adc45q
```

Sie können einen oder mehrere alternative Betreffnamen eingeben, um ein Zertifikat anzufordern, das mehr als eine Apex-Domain schützt:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.net
```

Sie können einen alternativen Namen eingeben, der auch für den Zugriff auf Ihre Website verwendet werden kann:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

Sie können ein Sternchen (*) als Platzhalter verwenden, um ein Zertifikat für mehrere Subdomains in derselben Domain zu erstellen:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

Sie können auch mehrere alternative Namen eingeben:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
subject-alternative-names b.example.com c.example.com d.example.com
```

Wenn Sie E-Mail für die Validierung verwenden, können Sie Optionen für die Domainvalidierung eingeben, um die Domain anzugeben, an die die Bestätigungs-E-Mail gesendet werden soll:

```
aws acm request-certificate --domain-name example.com --validation-  
method EMAIL --subject-alternative-names www.example.com --domain-validation-  
options DomainName=example.com,ValidationDomain=example.com
```

Mit dem folgenden Befehl wird die Protokollierung der Zertifikatstransparenz deaktiviert, wenn Sie ein neues Zertifikat anfordern:

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --  
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- API-Einheiten finden Sie [RequestCertificate](#) in der AWS CLI Befehlsreferenz.

resend-validation-email

Das folgende Codebeispiel zeigt die Verwendung `resend-validation-email`.

AWS CLI

Um die Bestätigungs-E-Mail für Ihre ACM Zertifikatsanfrage erneut zu senden

Der folgende `resend-validation-email` Befehl weist die Amazon-Zertifizierungsstelle an, eine Bestätigungs-E-Mail an die entsprechenden Adressen zu senden:

```
aws acm resend-validation-email --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --domain www.example.com --validation-domain example.com
```

- API-Einheiten finden Sie [ResendValidationEmail](#) in der AWS CLI Befehlsreferenz.

update-certificate-options

Das folgende Codebeispiel zeigt die Verwendung `update-certificate-options`.

AWS CLI

Um die Zertifikatsoptionen zu aktualisieren

Mit dem folgenden `update-certificate-options` Befehl wird die Protokollierung der Zertifikatstransparenz deaktiviert:

```
aws acm update-certificate-options --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --options CertificateTransparencyLoggingPreference=DISABLED
```

- API-Einheiten finden Sie [UpdateCertificateOptions](#) in der AWS CLI Befehlsreferenz.

APIGateway-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit API Gateway Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-api-key

Das folgende Codebeispiel zeigt die Verwendung `create-api-key`.

AWS CLI

Um einen API Schlüssel zu erstellen, der für eine bestehende API und eine Stage aktiviert ist

Befehl:

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for
development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- API Einheiten finden Sie [CreateApiKey](#) unter AWS CLI Befehlsreferenz.

create-authorizer

Das folgende Codebeispiel zeigt die Verwendung `create-authorizer`.

AWS CLI

Beispiel 1: So erstellen Sie einen tokenbasierten API Gateway Custom Authorizer für API

Im folgenden `create-authorizer` Beispiel wird ein tokenbasierter Authorizer erstellt.

```
aws apigateway create-authorizer \
  --rest-api-id 1234123412 \
  --name 'First-Token-Custom-Authorizer' \
  --type TOKEN \
  --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
  --identity-source 'method.request.header.Authorization' \
  --authorizer-result-ttl-in-seconds 300
```

Ausgabe:

```
{
  "authType": "custom",
  "name": "First-Token-Custom-Authorizer",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
  "authorizerResultTtlInSeconds": 300,
  "identitySource": "method.request.header.Authorization",
  "type": "TOKEN",
  "id": "z40xj0"
}
```

Beispiel 2: So erstellen Sie einen auf Cognito User Pools basierenden API Gateway Custom Authorizer für API

Im folgenden `create-authorizer` Beispiel wird ein auf Cognito User Pools basierender API Gateway Custom Authorizer erstellt.

```
aws apigateway create-authorizer \
  --rest-api-id 1234123412 \
  --name 'First_Cognito_Custom_Authorizer' \
  --type COGNITO_USER_POOLS \
  --provider-arns 'arn:aws:cognito-idp:us-east-1:123412341234:userpool/us-east-1_aWcZeQbuD' \
  --identity-source 'method.request.header.Authorization'
```

Ausgabe:

```
{
  "authType": "cognito_user_pools",
  "identitySource": "method.request.header.Authorization",
  "name": "First_Cognito_Custom_Authorizer",
  "providerARNs": [
    "arn:aws:cognito-idp:us-east-1:342398297714:userpool/us-east-1_qWbZzQhzE"
  ],
  "type": "COGNITO_USER_POOLS",
  "id": "5yid1t"
}
```

Beispiel 3: Um einen anforderungsbasierten API Gateway Custom Authorizer für das zu erstellen API

Im folgenden `create-authorizer` Beispiel wird ein anforderungsbasierter Authorizer erstellt.

```
aws apigateway create-authorizer \  
  --rest-api-id 1234123412 \  
  --name 'First_Request_Custom_Authorizer' \  
  --type REQUEST \  
  --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \  
  --identity-source 'method.request.header.Authorization,context.accountId' \  
  --authorizer-result-ttl-in-seconds 300
```

Ausgabe:

```
{  
  "id": "z40xj0",  
  "name": "First_Request_Custom_Authorizer",  
  "type": "REQUEST",  
  "authType": "custom",  
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",  
  "identitySource": "method.request.header.Authorization,context.accountId",  
  "authorizerResultTtlInSeconds": 300  
}
```

- API-Einheiten finden Sie [CreateAuthorizer](#) in der AWS CLI Befehlsreferenz.

create-base-path-mapping

Das folgende Codebeispiel zeigt die Verwendung `create-base-path-mapping`.

AWS CLI

Um die Basispfadzuordnung für einen benutzerdefinierten Domainnamen zu erstellen

Befehl:

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-  
api-id 1234123412 --stage prod --base-path v1
```

- API-Einheiten finden Sie [CreateBasePathMapping](#) in der AWS CLI Befehlsreferenz.

create-deployment

Das folgende Codebeispiel zeigt die Verwendung `create-deployment`.

AWS CLI

Um die konfigurierten Ressourcen für API eine neue Phase bereitzustellen

Befehl:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-description 'Development Stage' --description 'First deployment to the dev stage'
```

Um die konfigurierten Ressourcen für API eine bestehende Phase bereitzustellen

Befehl:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Second deployment to the dev stage'
```

Um die konfigurierten Ressourcen für eine API bestehende Phase mit Stufenvariablen bereitzustellen

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Drittes Deployment in der Entwicklungsphase' --variables key='value',='otherKeyotherValue'
```

- API-Einheiten finden [CreateDeployment AWS CLI](#) Sie in der Befehlsreferenz.

create-domain-name

Das folgende Codebeispiel zeigt die Verwendung `create-domain-name`.

AWS CLI

Um den benutzerdefinierten Domainnamen zu erstellen

Befehl:

```
aws apigateway create-domain-name --domain-name 'my.domain.tld' --certificate-name 'my.domain.tld cert' --certificate-arn 'arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3'
```

- API-Einheiten finden Sie [CreateDomainName](#) in der AWS CLI Befehlsreferenz.

create-model

Das folgende Codebeispiel zeigt die Verwendung `create-model`.

AWS CLI

Um ein Modell für ein zu erstellen API

Befehl:

```
aws apigateway create-model --rest-api-id 1234123412 --name 'firstModel' --
description 'The First Model' --content-type 'application/json' --schema
'{"$schema": "http://json-schema.org/draft-04/schema#", "title": "firstModel",
"type": "object", "properties": { "firstProperty" : { "type": "object",
"properties": { "key": { "type": "string" } } } } }'
```

Ausgabe:

```
{
  "contentType": "application/json",
  "description": "The First Model",
  "name": "firstModel",
  "id": "2rzg01",
  "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title
\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty
\": { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string
\" } } } } }"
```

- API-Einheiten finden Sie [CreateModel](#) in der AWS CLI Befehlsreferenz.

create-resource

Das folgende Codebeispiel zeigt die Verwendung `create-resource`.

AWS CLI

Um eine Ressource in einem zu erstellen API

Befehl:

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-part 'new-resource'
```

- API-Einheiten finden Sie [CreateResource](#) in der AWS CLI Befehlsreferenz.

create-rest-api

Das folgende Codebeispiel zeigt die Verwendung `create-rest-api`.

AWS CLI

Um ein zu erstellen API

Befehl:

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first API'
```

Um ein Duplikat API aus einem vorhandenen zu erstellen API

Befehl:

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is a copy of my first API' --clone-from 1234123412
```

- API-Einheiten finden Sie [CreateRestApi](#) in der AWS CLI Befehlsreferenz.

create-stage

Das folgende Codebeispiel zeigt die Verwendung `create-stage`.

AWS CLI

Um eine Phase in einer Phase zu erstellen API, die eine bestehende Bereitstellung enthalten wird

Befehl:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --description 'Development stage' --deployment-id a1b2c3
```

Um eine Phase in einer Phase zu erstellen API, die eine bestehende Bereitstellung und benutzerdefinierte Stufenvariablen enthält

Befehl:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'  
--description 'Development stage' --deployment-id a1b2c3 --variables  
key='value',otherKey='otherValue'
```

- API Einheiten finden Sie [CreateStage](#) in der AWS CLI Befehlsreferenz.

create-usage-plan-key

Das folgende Codebeispiel zeigt die Verwendung `create-usage-plan-key`.

AWS CLI

Ordnen Sie einen vorhandenen API Schlüssel einem Nutzungsplan zu

Befehl:

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --  
key-id 4vq3yryqm5
```

- API Einheiten finden Sie [CreateUsagePlanKey](#) in der AWS CLI Befehlsreferenz.

create-usage-plan

Das folgende Codebeispiel zeigt die Verwendung `create-usage-plan`.

AWS CLI

Um einen Nutzungsplan mit Drosselungs- und Kontingentlimits zu erstellen, der zu Beginn des Monats zurückgesetzt wird

Befehl:

```
aws apigateway create-usage-plan --name "New Usage Plan" --description "A new usage  
plan" --throttle burstLimit=10,rateLimit=5 --quota limit=500,offset=0,period=MONTH
```

- API Einheiten finden Sie unter [CreateUsagePlan AWS CLI](#) Befehlsreferenz.

delete-api-key

Das folgende Codebeispiel zeigt die Verwendung `delete-api-key`.

AWS CLI

Um einen API Schlüssel zu löschen

Befehl:

```
aws apigateway delete-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k
```

- APIEinzelheiten finden Sie [DeleteApiKey](#) in der AWS CLI Befehlsreferenz.

delete-authorizer

Das folgende Codebeispiel zeigt die Verwendung `delete-authorizer`.

AWS CLI

Um einen Custom Authorizer in einem zu löschen API

Befehl:

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- APIEinzelheiten finden Sie [DeleteAuthorizer](#) in der AWS CLI Befehlsreferenz.

delete-base-path-mapping

Das folgende Codebeispiel zeigt die Verwendung `delete-base-path-mapping`.

AWS CLI

Um eine Basispfadzuordnung für einen benutzerdefinierten Domainnamen zu löschen

Befehl:

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path 'dev'
```

- APIEinzelheiten finden Sie [DeleteBasePathMapping](#) in der AWS CLI Befehlsreferenz.

delete-client-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-client-certificate`.

AWS CLI

Um ein Client-Zertifikat zu löschen

Befehl:

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- API-Einheiten finden Sie [DeleteClientCertificate](#) in der AWS CLI Befehlsreferenz.

delete-deployment

Das folgende Codebeispiel zeigt die Verwendung `delete-deployment`.

AWS CLI

Um eine Bereitstellung in einem zu löschen API

Befehl:

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- API-Einheiten finden Sie [DeleteDeployment](#) in der AWS CLI Befehlsreferenz.

delete-domain-name

Das folgende Codebeispiel zeigt die Verwendung `delete-domain-name`.

AWS CLI

Um einen benutzerdefinierten Domainnamen zu löschen

Befehl:

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- API-Einheiten finden Sie [DeleteDomainName](#) in der AWS CLI Befehlsreferenz.

delete-integration-response

Das folgende Codebeispiel zeigt die Verwendung `delete-integration-response`.

AWS CLI

Um eine Integrationsantwort für eine bestimmte Ressource, Methode und einen Statuscode in einem API

Befehl:

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200
```

- API-Einheiten finden Sie [DeleteIntegrationResponse](#) unter AWS CLI Befehlsreferenz.

delete-integration

Das folgende Codebeispiel zeigt die Verwendung `delete-integration`.

AWS CLI

Um eine Integration für eine bestimmte Ressource und Methode in einem zu löschen API

Befehl:

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- API-Einheiten finden Sie [DeleteIntegration](#) in der AWS CLI Befehlsreferenz.

delete-method-response

Das folgende Codebeispiel zeigt die Verwendung `delete-method-response`.

AWS CLI

Um eine Methodenantwort für die angegebene Ressource, Methode und den Statuscode in einem API

Befehl:

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3  
--http-method GET --status-code 200
```

- API-Einheiten finden Sie [DeleteMethodResponse](#) in der AWS CLI Befehlsreferenz.

delete-method

Das folgende Codebeispiel zeigt die Verwendung `delete-method`.

AWS CLI

Um eine Methode für die angegebene Ressource in einem zu löschen API

Befehl:

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-  
method GET
```

- API-Einheiten finden Sie [DeleteMethod](#) in der AWS CLI Befehlsreferenz.

delete-model

Das folgende Codebeispiel zeigt die Verwendung `delete-model`.

AWS CLI

Um ein Modell in der angegebenen Datei zu löschen API

Befehl:

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- API-Einheiten finden Sie [DeleteModel](#) in der AWS CLI Befehlsreferenz.

delete-resource

Das folgende Codebeispiel zeigt die Verwendung `delete-resource`.

AWS CLI

Um eine Ressource in einem zu löschen API

Befehl:

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- API-Einheiten finden Sie [DeleteResource](#) in der AWS CLI Befehlsreferenz.

delete-rest-api

Das folgende Codebeispiel zeigt die Verwendung `delete-rest-api`.

AWS CLI

Um ein zu löschen API

Befehl:

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- API-Einheiten finden Sie [DeleteRestApi](#) in der AWS CLI Befehlsreferenz.

delete-stage

Das folgende Codebeispiel zeigt die Verwendung `delete-stage`.

AWS CLI

Um eine Phase in einem zu löschen API

Befehl:

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- API-Einheiten finden Sie [DeleteStage](#) in der AWS CLI Befehlsreferenz.

delete-usage-plan-key

Das folgende Codebeispiel zeigt die Verwendung `delete-usage-plan-key`.

AWS CLI

Um einen API Schlüssel aus einem Nutzungsplan zu entfernen

Befehl:

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- APIEinzelheiten finden Sie [DeleteUsagePlanKey](#) in der AWS CLI Befehlsreferenz.

delete-usage-plan

Das folgende Codebeispiel zeigt die Verwendung `delete-usage-plan`.

AWS CLI

Um einen Nutzungsplan zu löschen

Befehl:

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- APIEinzelheiten finden Sie [DeleteUsagePlan](#) in der AWS CLI Befehlsreferenz.

flush-stage-authorizers-cache

Das folgende Codebeispiel zeigt die Verwendung `flush-stage-authorizers-cache`.

AWS CLI

Um alle Autorisierungs-Cache-Einträge auf einer Bühne zu leeren

Befehl:

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-name dev
```

- APIEinzelheiten finden Sie [FlushStageAuthorizersCache](#) in der AWS CLI Befehlsreferenz.

flush-stage-cache

Das folgende Codebeispiel zeigt die Verwendung `flush-stage-cache`.

AWS CLI

Um den Cache für eine Phase zu leeren API

Befehl:

```
aws apigateway flush-stage-cache --rest-api-id 1234123412 --stage-name dev
```

- API-Einheiten finden Sie [FlushStageCache](#) in der AWS CLI Befehlsreferenz.

generate-client-certificate

Das folgende Codebeispiel zeigt die Verwendung `generate-client-certificate`.

AWS CLI

Um ein clientseitiges Zertifikat SSL zu erstellen

Befehl:

```
aws apigateway generate-client-certificate --description 'My First Client Certificate'
```

- API-Einheiten finden Sie [GenerateClientCertificate](#) in der AWS CLI Befehlsreferenz.

get-account

Das folgende Codebeispiel zeigt die Verwendung `get-account`.

AWS CLI

Um die API Gateway-Kontoeinstellungen abzurufen

Befehl:

```
aws apigateway get-account
```

Ausgabe:

```
{
```

```
"cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogsRole",
  "throttleSettings": {
    "rateLimit": 500.0,
    "burstLimit": 1000
  }
}
```

- APIEinzelheiten finden Sie [GetAccount](#) in der AWS CLI Befehlsreferenz.

get-api-key

Das folgende Codebeispiel zeigt die Verwendung `get-api-key`.

AWS CLI

Um die Informationen zu einem bestimmten API Schlüssel zu erhalten

Befehl:

```
aws apigateway get-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B30lkq0b1k
```

Ausgabe:

```
{
  "description": "My first key",
  "enabled": true,
  "stageKeys": [
    "a1b2c3d4e5/dev",
    "e5d4c3b2a1/dev"
  ],
  "lastUpdatedDate": 1456184515,
  "createdDate": 1456184452,
  "id": "8bk1k8b11k3sB38D9B310enyWT8c09B30lkq0b1k",
  "name": "My key"
}
```

- APIEinzelheiten finden Sie [GetApiKey](#) in der AWS CLI Befehlsreferenz.

get-api-keys

Das folgende Codebeispiel zeigt die Verwendung `get-api-keys`.

AWS CLI

Um die Liste der API Schlüssel abzurufen

Befehl:

```
aws apigateway get-api-keys
```

Ausgabe:

```
{
  "items": [
    {
      "description": "My first key",
      "enabled": true,
      "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
      ],
      "lastUpdatedDate": 1456184515,
      "createdDate": 1456184452,
      "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0blk",
      "name": "My key"
    }
  ]
}
```

- APIEinzelheiten finden Sie [GetApiKeys](#) in der AWS CLI Befehlsreferenz.

get-authorizer

Das folgende Codebeispiel zeigt die Verwendung `get-authorizer`.

AWS CLI

Um die API API Gateway-Per-Authorizer-Einstellungen abzurufen

Befehl:

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```

Ausgabe:

```
{
  "authorizerResultTtlInSeconds": 300,
  "name": "MyAuthorizer",
  "type": "TOKEN",
  "identitySource": "method.request.header.Authorization",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:authorizer_function/invocations",
  "id": "gfi4n3"
}
```

- API-Einheiten finden Sie [GetAuthorizer](#) in der AWS CLI Befehlsreferenz.

get-authorizers

Das folgende Codebeispiel zeigt die Verwendung `get-authorizers`.

AWS CLI

Um die Liste der Autorisierer für ein abzurufen REST API

Befehl:

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

Ausgabe:

```
{
  "items": [
    {
      "name": "MyAuthorizer",
      "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123412341234:function:My_Authorizer_Function/
invocations",
      "authorizerResultTtlInSeconds": 300,
      "identitySource": "method.request.header.Authorization",
      "type": "TOKEN",
      "id": "gfi4n3"
    }
  ]
}
```

- API-Einheiten finden Sie [GetAuthorizers](#) in der AWS CLI Befehlsreferenz.

get-base-path-mapping

Das folgende Codebeispiel zeigt die Verwendung `get-base-path-mapping`.

AWS CLI

Um die Basispfadzuordnung für einen benutzerdefinierten Domainnamen abzurufen

Befehl:

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-path v1
```

Ausgabe:

```
{
  "basePath": "v1",
  "restApiId": "1234w4321e",
  "stage": "api"
}
```

- API-Einheiten finden Sie [GetBasePathMapping](#) in der AWS CLI Befehlsreferenz.

get-base-path-mappings

Das folgende Codebeispiel zeigt die Verwendung `get-base-path-mappings`.

AWS CLI

Um die Basispfadzuordnungen für einen benutzerdefinierten Domainnamen abzurufen

Befehl:

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

Ausgabe:

```
{
  "items": [
    {
      "basePath": "(none)",
      "restApiId": "1234w4321e",
    }
  ]
}
```

```
        "stage": "dev"
      },
      {
        "basePath": "v1",
        "restApiId": "1234w4321e",
        "stage": "api"
      }
    ]
  }
}
```

- API-Einheiten finden Sie [GetBasePathMappings](#) in der AWS CLI Befehlsreferenz.

get-client-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-client-certificate`.

AWS CLI

Um ein Client-Zertifikat zu erhalten

Befehl:

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- API-Einheiten finden Sie [GetClientCertificate](#) in der AWS CLI Befehlsreferenz.

get-client-certificates

Das folgende Codebeispiel zeigt die Verwendung `get-client-certificates`.

AWS CLI

Um eine Liste von Client-Zertifikaten abzurufen

Befehl:

```
aws apigateway get-client-certificates
```

Ausgabe:

```
{
  "items": [
```



```
{
  "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
content> -----END CERTIFICATE-----",
  "clientCertificateId": "a1b2c3",
  "expirationDate": 1483556561,
  "description": "My Client Certificate",
  "createdDate": 1452020561
}
```

- API-Einheiten finden Sie [GetClientCertificates](#) in der AWS CLI Befehlsreferenz.

get-deployment

Das folgende Codebeispiel zeigt die Verwendung `get-deployment`.

AWS CLI

Um Informationen über eine Bereitstellung zu erhalten

Befehl:

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

Ausgabe:

```
{
  "description": "myDeployment",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

- API-Einheiten finden Sie [GetDeployment](#) in der AWS CLI Befehlsreferenz.

get-deployments

Das folgende Codebeispiel zeigt die Verwendung `get-deployments`.

AWS CLI

Um eine Liste der Bereitstellungen für ein abzurufen REST API

Befehl:

```
aws apigateway get-deployments --rest-api-id 1234123412
```

Ausgabe:

```
{
  "items": [
    {
      "createdDate": 1453797217,
      "id": "0a2b4c",
      "description": "Deployed my API for the first time"
    }
  ]
}
```

- APIEinzelheiten finden Sie [GetDeployments](#) in der AWS CLI Befehlsreferenz.

get-domain-name

Das folgende Codebeispiel zeigt die Verwendung `get-domain-name`.

AWS CLI

Um Informationen zu einem benutzerdefinierten Domainnamen zu erhalten

Befehl:

```
aws apigateway get-domain-name --domain-name api.domain.tld
```

Ausgabe:

```
{
  "domainName": "api.domain.tld",
  "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
  "certificateName": "uploadedCertificate",
  "certificateUploadDate": 1462565487
}
```

- APIEinzelheiten finden Sie [GetDomainName](#) in der AWS CLI Befehlsreferenz.

get-domain-names

Das folgende Codebeispiel zeigt die Verwendung `get-domain-names`.

AWS CLI

Um eine Liste mit benutzerdefinierten Domainnamen zu erhalten

Befehl:

```
aws apigateway get-domain-names
```

Ausgabe:

```
{
  "items": [
    {
      "distributionDomainName": "d9511k3109bkd.cloudfront.net",
      "certificateUploadDate": 1452812505,
      "certificateName": "my_custom_domain-certificate",
      "domainName": "subdomain.domain.tld"
    }
  ]
}
```

- API-Einheiten finden Sie [GetDomainNames](#) in der AWS CLI Befehlsreferenz.

get-export

Das folgende Codebeispiel zeigt die Verwendung `get-export`.

AWS CLI

Um die JSON Swagger-Vorlage für eine Phase abzurufen

Befehl:

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Um die JSON Swagger-Vorlage + API Gateway-Erweiterungen für eine Phase zu erhalten

Befehl:

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Um die JSON Swagger-Vorlage + Postman-Erweiterungen für eine Phase zu erhalten

Befehl:

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

- APIEinzelheiten finden Sie [GetExport](#) in AWS CLI der Befehlsreferenz.

get-integration-response

Das folgende Codebeispiel zeigt die Verwendung `get-integration-response`.

AWS CLI

Um die Konfiguration der Integrationsantwort für eine HTTP Methode abzurufen, die unter REST API der Ressource `a` definiert ist

Befehl:

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

Ausgabe:

```
{
  "statusCode": "200",
  "responseTemplates": {
    "application/json": null
  }
}
```

- APIEinzelheiten finden Sie [GetIntegrationResponse](#) in der AWS CLI Befehlsreferenz.

get-integration

Das folgende Codebeispiel zeigt die Verwendung `get-integration`.

AWS CLI

Um die Integrationskonfiguration für eine HTTP Methode abzurufen, die unter REST API der Ressource a definiert ist

Befehl:

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

Ausgabe:

```
{
  "httpMethod": "POST",
  "integrationResponses": {
    "200": {
      "responseTemplates": {
        "application/json": null
      },
      "statusCode": "200"
    }
  },
  "cacheKeyParameters": [],
  "type": "AWS",
  "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
  "cacheNamespace": "y9h6rt"
}
```

- APIEinzelheiten finden Sie [GetIntegration](#) in der AWS CLI Befehlsreferenz.

get-method-response

Das folgende Codebeispiel zeigt die Verwendung get-method-response.

AWS CLI

Um die Ressourcenkonfiguration der Methodenantwort für eine HTTP Methode abzurufen, die unter REST API der Ressource a definiert ist

Befehl:

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

Ausgabe:

```
{
  "responseModels": {
    "application/json": "Empty"
  },
  "statusCode": "200"
}
```

- API-Einheiten finden Sie [GetMethodResponse](#) in der AWS CLI Befehlsreferenz.

get-method

Das folgende Codebeispiel zeigt die Verwendung `get-method`.

AWS CLI

Um die Methodenressourcenkonfiguration für eine HTTP-Methode abzurufen, die unter REST API der Ressource `a` definiert ist

Befehl:

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

Ausgabe:

```
{
  "apiKeyRequired": false,
  "httpMethod": "GET",
  "methodIntegration": {
    "integrationResponses": {
      "200": {
        "responseTemplates": {
          "application/json": null
        },
        "statusCode": "200"
      }
    }
  }
}
```

```

    }
  },
  "cacheKeyParameters": [],
  "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
  "httpMethod": "POST",
  "cacheNamespace": "y9h6rt",
  "type": "AWS"
},
"requestParameters": {},
"methodResponses": {
  "200": {
    "responseModels": {
      "application/json": "Empty"
    },
    "statusCode": "200"
  }
},
"authorizationType": "NONE"
}

```

- APIEinzelheiten finden Sie [GetMethod](#) in der AWS CLI Befehlsreferenz.

get-model-template

Das folgende Codebeispiel zeigt die Verwendung `get-model-template`.

AWS CLI

Um die Zuordnungsvorlage für ein Modell abzurufen, das unter `a` definiert ist REST API

Befehl:

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

Ausgabe:

```
{
  "value": "#set($inputRoot = $input.path('$'))\n{ }"
}
```

- APIEinzelheiten finden Sie [GetModelTemplate](#) in der AWS CLI Befehlsreferenz.

get-model

Das folgende Codebeispiel zeigt die Verwendung `get-model`.

AWS CLI

Um die Konfiguration für ein Modell abzurufen, das unter a definiert ist REST API

Befehl:

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

Ausgabe:

```
{
  "contentType": "application/json",
  "description": "This is a default empty schema model",
  "name": "Empty",
  "id": "etd5w5",
  "schema": "{\n  \"$schema\": \"http://json-schema.org/draft-04/schema#\",\n  \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}"
}
```

- API-Einheiten finden Sie [GetModel](#) in der AWS CLI Befehlsreferenz.

get-models

Das folgende Codebeispiel zeigt die Verwendung `get-models`.

AWS CLI

Um eine Liste von Modellen für ein zu erhalten REST API

Befehl:

```
aws apigateway get-models --rest-api-id 1234123412
```

Ausgabe:

```
{
  "items": [
```



```

    {
      "description": "This is a default error schema model",
      "schema": "{\n  \"\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#\n\",,\n  \"title\" : \"Error Schema\",,\n  \"type\" : \"object\",,\n  \"properties\" :\n  {\n    \"message\" : { \"type\" : \"string\" }\n  }\n}",
      "contentType": "application/json",
      "id": "7tpbze",
      "name": "Error"
    },
    {
      "description": "This is a default empty schema model",
      "schema": "{\n  \"\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#\n\",,\n  \"title\" : \"Empty Schema\",,\n  \"type\" : \"object\"\n}",
      "contentType": "application/json",
      "id": "etd5w5",
      "name": "Empty"
    }
  ]
}

```

- API-Einheiten finden Sie [GetModels](#) in der AWS CLI Befehlsreferenz.

get-resource

Das folgende Codebeispiel zeigt die Verwendung `get-resource`.

AWS CLI

Um Informationen über eine Ressource zu erhalten

Befehl:

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

Ausgabe:

```

{
  "path": "/path",
  "pathPart": "path",
  "id": "zwo0y3",
  "parentId": "uyokt6ij2g"
}

```

- API-Einheiten finden Sie [GetResource](#) unter AWS CLI Befehlsreferenz.

get-resources

Das folgende Codebeispiel zeigt die Verwendung `get-resources`.

AWS CLI

Um eine Liste von Ressourcen für ein zu erhalten REST API

Befehl:

```
aws apigateway get-resources --rest-api-id 1234123412
```

Ausgabe:

```
{
  "items": [
    {
      "path": "/resource/subresource",
      "resourceMethods": {
        "POST": {}
      },
      "id": "024ace",
      "pathPart": "subresource",
      "parentId": "ai5b02"
    }
  ]
}
```

- API-Einheiten finden Sie [GetResources](#) in der AWS CLI Befehlsreferenz.

get-rest-api

Das folgende Codebeispiel zeigt die Verwendung `get-rest-api`.

AWS CLI

Um Informationen zu erhalten über API

Befehl:

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

Ausgabe:

```
{
  "name": "myAPI",
  "id": "o1y243m4f5",
  "createdDate": 1453416433
}
```

- APIEinzelheiten finden Sie [GetRestApi](#) in der AWS CLI Befehlsreferenz.

get-rest-apis

Das folgende Codebeispiel zeigt die Verwendung `get-rest-apis`.

AWS CLI

Um eine Liste von zu erhalten REST APIs

Befehl:

```
aws apigateway get-rest-apis
```

Ausgabe:

```
{
  "items": [
    {
      "createdDate": 1438884790,
      "id": "12s44z21rb",
      "name": "My First API"
    }
  ]
}
```

- APIEinzelheiten finden Sie [GetRestApis](#) in der AWS CLI Befehlsreferenz.

get-sdk

Das folgende Codebeispiel zeigt die Verwendung `get-sdk`.

AWS CLI

Um das Android SDK für eine REST API Phase zu bekommen

Befehl:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-type android
--parameters
  groupId='com.mycompany',invokerPackage='com.mycompany.clientsdk',artifactId='Mycompany-
  client',artifactVersion='1.0.0' /path/to/android_sdk.zip
```

Ausgabe:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}
```

Um das IOS SDK für eine REST API Bühne zu bekommen

Befehl:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type objectivec --parameters classPrefix='myprefix' /path/to/iOS_sdk.zip
```

Ausgabe:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip
  \"\"
}
```

Um das Javascript SDK für eine REST API Stufe zu bekommen

Befehl:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type javascript /path/to/javascript_sdk.zip
```

Ausgabe:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip\"
}
```

- API-Einheiten finden Sie [GetSdk](#) in der AWS CLI Befehlsreferenz.

get-stage

Das folgende Codebeispiel zeigt die Verwendung `get-stage`.

AWS CLI

Um Informationen über die Phase eines API Mannes zu erhalten

Befehl:

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

Ausgabe:

```
{
  "stageName": "dev",
  "cacheClusterSize": "0.5",
  "cacheClusterEnabled": false,
  "cacheClusterStatus": "NOT_AVAILABLE",
  "deploymentId": "rbh1fj",
  "lastUpdatedDate": 1466802961,
  "createdDate": 1460682074,
  "methodSettings": {
    "*/*": {
      "cacheTtlInSeconds": 300,
      "loggingLevel": "INFO",
      "dataTraceEnabled": false,
      "metricsEnabled": true,
      "unauthorizedCacheControlHeaderStrategy":
        "SUCCEED_WITH_RESPONSE_HEADER",
      "throttlingRateLimit": 500.0,
    }
  }
}
```

```

        "cacheDataEncrypted": false,
        "cachingEnabled": false,
        "throttlingBurstLimit": 1000,
        "requireAuthorizationForCacheControl": true
    },
    "~1resource/GET": {
        "cacheTtlInSeconds": 300,
        "loggingLevel": "INFO",
        "dataTraceEnabled": false,
        "metricsEnabled": true,
        "unauthorizedCacheControlHeaderStrategy":
"SUCCEED_WITH_RESPONSE_HEADER",
        "throttlingRateLimit": 500.0,
        "cacheDataEncrypted": false,
        "cachingEnabled": false,
        "throttlingBurstLimit": 1000,
        "requireAuthorizationForCacheControl": true
    }
}
}
}

```

- API-Einheiten finden Sie [GetStage](#) in der AWS CLI Befehlsreferenz.

get-stages

Das folgende Codebeispiel zeigt die Verwendung `get-stages`.

AWS CLI

Um die Liste der Stufen für ein zu erhalten REST API

Befehl:

```
aws apigateway get-stages --rest-api-id 1234123412
```

Ausgabe:

```
{
  "item": [
    {
      "stageName": "dev",
      "cacheClusterSize": "0.5",

```

```

    "cacheClusterEnabled": true,
    "cacheClusterStatus": "AVAILABLE",
    "deploymentId": "123h64",
    "lastUpdatedDate": 1456185138,
    "createdDate": 1453589092,
    "methodSettings": {
      "~1resource~1subresource/POST": {
        "cacheTtlInSeconds": 300,
        "loggingLevel": "INFO",
        "dataTraceEnabled": true,
        "metricsEnabled": true,
        "throttlingRateLimit": 500.0,
        "cacheDataEncrypted": false,
        "cachingEnabled": false,
        "throttlingBurstLimit": 1000
      }
    }
  }
]
}

```

- API-Einheiten finden Sie [GetStages](#) in der AWS CLI Befehlsreferenz.

get-usage-plan-key

Das folgende Codebeispiel zeigt die Verwendung `get-usage-plan-key`.

AWS CLI

Um die Details eines API-Schlüssels abzurufen, der einem Nutzungsplan zugeordnet ist

Befehl:

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- API-Einheiten finden Sie [GetUsagePlanKey](#) unter AWS CLI Befehlsreferenz.

get-usage-plan-keys

Das folgende Codebeispiel zeigt die Verwendung `get-usage-plan-keys`.

AWS CLI

Um die Liste der API Schlüssel abzurufen, die einem Nutzungsplan zugeordnet sind

Befehl:

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- APIEinzelheiten finden Sie [GetUsagePlanKeys](#) in der AWS CLI Befehlsreferenz.

get-usage-plan

Das folgende Codebeispiel zeigt die Verwendung `get-usage-plan`.

AWS CLI

Um die Details eines Nutzungsplans abzurufen

Befehl:

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- APIEinzelheiten finden Sie [GetUsagePlan](#) in der AWS CLI Befehlsreferenz.

get-usage-plans

Das folgende Codebeispiel zeigt die Verwendung `get-usage-plans`.

AWS CLI

Um die Details aller Nutzungspläne abzurufen

Befehl:

```
aws apigateway get-usage-plans
```

- APIEinzelheiten finden Sie [GetUsagePlans](#) in der AWS CLI Befehlsreferenz.

get-usage

Das folgende Codebeispiel zeigt die Verwendung `get-usage`.

AWS CLI

Um die Nutzungsdetails für einen Nutzungsplan abzurufen

Befehl:

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-date "2016-08-17"
```

- APIEinzelheiten finden Sie [GetUsage](#) in der AWS CLI Befehlsreferenz.

import-rest-api

Das folgende Codebeispiel zeigt die Verwendung `import-rest-api`.

AWS CLI

Um eine Swagger-Vorlage zu importieren und eine zu erstellen API

Befehl:

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- APIEinzelheiten finden Sie [ImportRestApi](#) in der AWS CLI Befehlsreferenz.

put-integration-response

Das folgende Codebeispiel zeigt die Verwendung `put-integration-response`.

AWS CLI

Um eine Integrationsantwort als Standardantwort mit einer definierten Zuordnungsvorlage zu erstellen

Befehl:

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200 --selection-pattern "" --response-templates '{"application/json": "{\"json\": \"template\"}"}'
```

Um eine Integrationsantwort mit einer Regex von 400 und einem statisch definierten Header-Wert zu erstellen

Befehl:

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 400 --selection-pattern 400 --response-parameters '{"method.response.header.custom-header": "''''''custom-value''''''}'
```

- APIEinzelheiten finden Sie unter Befehlsreferenz [PutIntegrationResponse](#).AWS CLI

put-integration

Das folgende Codebeispiel zeigt die Verwendung `put-integration`.

AWS CLI

Um eine MOCK Integrationsanfrage zu erstellen

Befehl:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type MOCK --request-templates '{"application/json": "{\\"statusCode\\": 200}" }'
```

Um eine HTTP Integrationsanfrage zu erstellen

Befehl:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type HTTP --integration-http-method GET --uri 'https://domain.tld/path'
```

So erstellen Sie eine AWS Integrationsanfrage mit einem Lambda-Funktionsendpunkt

Befehl:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type AWS --integration-http-method POST --uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:function_name/invocations'
```

- APIEinzelheiten finden Sie [PutIntegration](#) in der AWS CLI Befehlsreferenz.

put-method-response

Das folgende Codebeispiel zeigt die Verwendung `put-method-response`.

AWS CLI

Um eine Methodenantwort unter dem angegebenen Statuscode mit einem benutzerdefinierten Methodenantwort-Header zu erstellen

Befehl:

```
aws apigateway put-method-response --rest-api-id 1234123412 --  
resource-id a1b2c3 --http-method GET --status-code 400 --response-  
parameters "method.response.header.custom-header=false"
```

- APIEinzelheiten finden Sie [PutMethodResponse](#) unter AWS CLI Befehlsreferenz.

put-method

Das folgende Codebeispiel zeigt die Verwendung `put-method`.

AWS CLI

Um eine Methode für eine Ressource in API einem Anforderungsheader ohne Autorisierung, ohne API Schlüssel und ohne benutzerdefinierte Methode zu erstellen

Befehl:

```
aws apigateway put-method --rest-api-id 1234123412 --resource-id a1b2c3 --  
http-method PUT --authorization-type "NONE" --no-api-key-required --request-  
parameters "method.request.header.custom-header=false"
```

- APIEinzelheiten finden Sie [PutMethod](#) unter AWS CLI Befehlsreferenz.

put-rest-api

Das folgende Codebeispiel zeigt die Verwendung `put-rest-api`.

AWS CLI

Um ein vorhandenes API mit einer Swagger-Vorlage zu überschreiben

Befehl:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body  
'fileb:///path/to/API_Swagger_template.json'
```

Um eine Swagger-Vorlage mit einer vorhandenen zusammenzuführen API

Befehl:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///  
path/to/API_Swagger_template.json'
```

- APIEinzelheiten finden Sie [PutRestApi](#) in der AWS CLI Befehlsreferenz.

test-invoke-authorizer

Das folgende Codebeispiel zeigt die Verwendung `test-invoke-authorizer`.

AWS CLI

Rufen Sie zum Testen eine Anfrage an einen Custom Authorizer auf, die den erforderlichen Header und den erforderlichen Wert enthält

Befehl:

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-  
id 5yid1t --headers Authorization='Value'
```

- APIEinzelheiten finden Sie [TestInvokeAuthorizer](#) in der AWS CLI Befehlsreferenz.

test-invoke-method

Das folgende Codebeispiel zeigt die Verwendung `test-invoke-method`.

AWS CLI

Rufen Sie zum Testen die Root-Ressource in einem auf, API indem Sie eine GET Anfrage stellen

Befehl:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id av15sg8fw8
--http-method GET --path-with-query-string '/'
```

Um zu testen, rufen Sie eine Subressource in einer auf, API indem Sie eine GET Anfrage mit einem angegebenen Pfadparameterwert stellen

Befehl:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- APIEinzelheiten finden Sie unter [TestInvokeMethod AWS CLI](#) Befehlsreferenz.

update-account

Das folgende Codebeispiel zeigt die Verwendung `update-account`.

AWS CLI

Um die IAM Rolle ARN für die Protokollierung in CloudWatch Logs zu ändern

Befehl:

```
aws apigateway update-account --patch-operations op='replace',path='/
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

Ausgabe:

```
{
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogs",
  "throttleSettings": {
    "rateLimit": 1000.0,
    "burstLimit": 2000
  }
}
```

- APIEinzelheiten finden Sie [UpdateAccount](#) in der AWS CLI Befehlsreferenz.

update-api-key

Das folgende Codebeispiel zeigt die Verwendung `update-api-key`.

AWS CLI

Um den Namen eines API Schlüssels zu ändern

Befehl:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --  
patch-operations op='replace',path='/name',value='newName'
```

Ausgabe:

```
{  
  "description": "currentDescription",  
  "enabled": true,  
  "stageKeys": [  
    "41t2j324r5/dev"  
  ],  
  "lastUpdatedDate": 1470086052,  
  "createdDate": 1445460347,  
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",  
  "name": "newName"  
}
```

Um den API Schlüssel zu deaktivieren

Befehl:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --  
patch-operations op='replace',path='/enabled',value='false'
```

Ausgabe:

```
{  
  "description": "currentDescription",  
  "enabled": false,  
  "stageKeys": [  
    "41t2j324r5/dev"  
  ]  
}
```

```

    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
  }

```

- API-Einheiten finden Sie [UpdateApiKey](#) in der AWS CLI Befehlsreferenz.

update-authorizer

Das folgende Codebeispiel zeigt die Verwendung `update-authorizer`.

AWS CLI

Um den Namen des Custom Authorizers zu ändern

Befehl:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/name',value='testAuthorizer'
```

Ausgabe:

```

{
  "authType": "custom",
  "name": "testAuthorizer",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthorizer/invocations",
  "authorizerResultTtlInSeconds": 300,
  "identitySource": "method.request.header.Authorization",
  "type": "TOKEN",
  "id": "gfi4n3"
}

```

Um die Lambda-Funktion zu ändern, die vom Custom Authorizer aufgerufen wird

Befehl:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/authorizerUri',value='arn:aws:apigateway:us-
```

```
west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-  
west-2:123412341234:function:newAuthorizer/invocations'
```

Ausgabe:

```
{  
  "authType": "custom",  
  "name": "testAuthorizer",  
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123412341234:function:newAuthorizer/invocations",  
  "authorizerResultTtlInSeconds": 300,  
  "identitySource": "method.request.header.Authorization",  
  "type": "TOKEN",  
  "id": "gfi4n3"  
}
```

- APIEinzelheiten finden Sie [UpdateAuthorizer](#) in AWS CLI der Befehlsreferenz.

update-base-path-mapping

Das folgende Codebeispiel zeigt die Verwendung `update-base-path-mapping`.

AWS CLI

Um den Basispfad für einen benutzerdefinierten Domainnamen zu ändern

Befehl:

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-  
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

Ausgabe:

```
{  
  "basePath": "v1",  
  "restApiId": "1234123412",  
  "stage": "api"  
}
```

- APIEinzelheiten finden Sie [UpdateBasePathMapping](#) in der AWS CLI Befehlsreferenz.

update-client-certificate

Das folgende Codebeispiel zeigt die Verwendung `update-client-certificate`.

AWS CLI

Um die Beschreibung eines Client-Zertifikats zu aktualisieren

Befehl:

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-operations op='replace',path='/description',value='My new description'
```

- API-Einheiten finden Sie [UpdateClientCertificate](#) in der AWS CLI Befehlsreferenz.

update-deployment

Das folgende Codebeispiel zeigt die Verwendung `update-deployment`.

AWS CLI

Um die Beschreibung einer Bereitstellung zu ändern

Befehl:

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --patch-operations op='replace',path='/description',value='newDescription'
```

Ausgabe:

```
{
  "description": "newDescription",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

- API-Einheiten finden Sie [UpdateDeployment](#) in der AWS CLI Befehlsreferenz.

update-domain-name

Das folgende Codebeispiel zeigt die Verwendung `update-domain-name`.

AWS CLI

Um den Zertifikatsnamen für einen benutzerdefinierten Domainnamen zu ändern

Im folgenden `update-domain-name` Beispiel wird der Zertifikatsname für eine benutzerdefinierte Domäne geändert.

```
aws apigateway update-domain-name \  
  --domain-name api.domain.tld \  
  --patch-operations op='replace',path='/certificateArn',value='arn:aws:acm:us-  
west-2:111122223333:certificate/CERTEXAMPLE123EXAMPLE'
```

Ausgabe:

```
{  
  "domainName": "api.domain.tld",  
  "distributionDomainName": "d123456789012.cloudfront.net",  
  "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/  
CERTEXAMPLE123EXAMPLE",  
  "certificateUploadDate": 1462565487  
}
```

Weitere Informationen finden Sie unter [Einrichten eines benutzerdefinierten Domainnamens für ein API In-Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateDomainName](#) unter AWS CLI Befehlsreferenz.

update-integration-response

Das folgende Codebeispiel zeigt die Verwendung `update-integration-response`.

AWS CLI

Um einen Antwort-Header einer Integration so zu ändern, dass er eine statische Zuordnung von '*' hat

Befehl:

```
aws apigateway update-integration-response --rest-api-id 1234123412 --  
resource-id 3gapai --http-method GET --status-code 200 --patch-operations  
  op='replace',path='/responseParameters/method.response.header.Access-Control-Allow-  
Origin',value='''*'''
```

Ausgabe:

```
{
  "statusCode": "200",
  "responseParameters": {
    "method.response.header.Access-Control-Allow-Origin": "*"
  }
}
```

Um einen Integrationsantwort-Header zu entfernen

Befehl:

```
aws apigateway update-integration-response --rest-api-id 1234123412 --resource-id 3gapai --http-method GET --status-code 200 --patch-operations op='remove',path='/responseParameters/method.response.header.Access-Control-Allow-Origin'
```

- API-Einheiten finden Sie [UpdateIntegrationResponse](#) in der AWS CLI Befehlsreferenz.

update-integration

Das folgende Codebeispiel zeigt die Verwendung `update-integration`.

AWS CLI

Um die Zuordnungsvorlage „Content-Type: application/json“ hinzuzufügen, die mit Input Passthrough konfiguriert ist

Befehl:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='add',path='/requestTemplates/application~1json'"
```

Um die mit einer benutzerdefinierten Vorlage konfigurierte Zuordnungsvorlage „Content-Type: application/json“ zu aktualisieren (zu ersetzen)

Befehl:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='replace',path='/requestTemplates/  
application~1json',value='{\"example\": \"json\"}'"
```

Um eine benutzerdefinierte Vorlage, die mit „Content-Type: application/json“ verknüpft ist, mit Input Passthrough zu aktualisieren (zu ersetzen)

Befehl:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

Um die Zuordnungsvorlage „Content-Type: application/json“ zu entfernen

Befehl:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- API-Einheiten finden Sie in der Befehlsreferenz. [UpdateIntegration](#) AWS CLI

update-method-response

Das folgende Codebeispiel zeigt die Verwendung `update-method-response`.

AWS CLI

Um einen neuen Methodenantwort-Header für die 200-Antwort in einer Methode zu erstellen und ihn als nicht erforderlich zu definieren (Standard)

Befehl:

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200 --patch-operations op="add",path="/responseParameters/method.response.header.custom-header",value="false"
```

Um ein Antwortmodell für die 200-Antwort in einer Methode zu löschen

Befehl:

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200 --patch-operations op="remove",path="/responseModels/application~1json"
```

- API-Einheiten finden Sie [UpdateMethodResponse](#) unter AWS CLI Befehlsreferenz.

update-method

Das folgende Codebeispiel zeigt die Verwendung `update-method`.

AWS CLI

Beispiel 1: Um eine Methode so zu ändern, dass sie einen API-Schlüssel erfordert

Im folgenden `update-method` Beispiel wird die Methode dahingehend geändert, dass ein API-Schlüssel erforderlich ist.

```
aws apigateway update-method \
  --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

Ausgabe:

```
{
  "httpMethod": "GET",
  "authorizationType": "NONE",
  "apiKeyRequired": true,
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {}
    }
  }
}
```

```

    }
  },
  "methodIntegration": {
    "type": "AWS",
    "httpMethod": "POST",
    "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
    "passthroughBehavior": "WHEN_NO_MATCH",
    "contentHandling": "CONVERT_TO_TEXT",
    "timeoutInMillis": 29000,
    "cacheNamespace": "h7i8j9",
    "cacheKeyParameters": [],
    "integrationResponses": {
      "200": {
        "statusCode": "200",
        "responseTemplates": {}
      }
    }
  }
}
}
}

```

Beispiel 2: Um eine Methode so zu ändern, dass eine Autorisierung erforderlich ist IAM

Im folgenden `update-method` Beispiel wird die Methode dahingehend geändert, dass eine IAM Autorisierung erforderlich ist.

```

aws apigateway update-method \
  --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"

```

Ausgabe:

```

{
  "httpMethod": "GET",
  "authorizationType": "AWS_IAM",
  "apiKeyRequired": false,
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {}
    }
  }
}

```

```

    },
    "methodIntegration": {
      "type": "AWS",
      "httpMethod": "POST",
      "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
      "passthroughBehavior": "WHEN_NO_MATCH",
      "contentHandling": "CONVERT_TO_TEXT",
      "timeoutInMillis": 29000,
      "cacheNamespace": "h7i8j9",
      "cacheKeyParameters": [],
      "integrationResponses": {
        "200": {
          "statusCode": "200",
          "responseTemplates": {}
        }
      }
    }
  }
}

```

Beispiel 3: Um eine Methode so zu ändern, dass eine Lambda-Autorisierung erforderlich ist

Im folgenden `update-method` Beispiel wird die Methode dahingehend geändert, dass eine Lambda-Autorisierung erforderlich ist.

```

aws apigateway update-method --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/authorizationType",value="CUSTOM"
  op="replace",path="/authorizerId",value="e4f5g6"

```

Ausgabe:

```

{
  "httpMethod": "GET",
  "authorizationType": "CUSTOM",
  "authorizerId" : "e4f5g6",
  "apiKeyRequired": false,
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {}
    }
  }
}

```

```

    },
    "methodIntegration": {
      "type": "AWS",
      "httpMethod": "POST",
      "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
      "passthroughBehavior": "WHEN_NO_MATCH",
      "contentHandling": "CONVERT_TO_TEXT",
      "timeoutInMillis": 29000,
      "cacheNamespace": "h7i8j9",
      "cacheKeyParameters": [],
      "integrationResponses": {
        "200": {
          "statusCode": "200",
          "responseTemplates": {}
        }
      }
    }
  }
}

```

Weitere Informationen finden [Sie unter Nutzungspläne mit dem API Gateway erstellen, konfigurieren und testen CLI REST API und den Zugriff auf ein REST API integriertes API Gateway steuern und verwalten](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateMethod](#) in der AWS CLI Befehlsreferenz.

update-model

Das folgende Codebeispiel zeigt die Verwendung `update-model`.

AWS CLI

Um die Beschreibung eines Modells in einem zu ändern API

Befehl:

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/description,value='New Description'
```

Um das Schema eines Modells in einem zu ändern API

Befehl:


```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/schema,value='{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title\": \"Empty Schema\", \"type\": \"object\" }'
```

- API-Einheiten finden Sie [UpdateModel](#) in der AWS CLI Befehlsreferenz.

update-resource

Das folgende Codebeispiel zeigt die Verwendung `update-resource`.

AWS CLI

Um eine Ressource zu verschieben und sie unter einer anderen übergeordneten Ressource in einem API

Befehl:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/parentId,value='3c2b1a'
```

Ausgabe:

```
{
  "path": "/resource",
  "pathPart": "resource",
  "id": "1a2b3c",
  "parentId": "3c2b1a"
}
```

Um eine Ressource (`pathPart`) umzubenennen in einem API

Befehl:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/pathPart,value=newresourcenam
```

Ausgabe:

```
{
  "path": "/newresourcenam",
  "pathPart": "newresourcenam",
}
```

```
"id": "1a2b3c",  
"parentId": "3c2b1a"  
}
```

- API-Einheiten finden Sie [UpdateResource](#) in der AWS CLI Befehlsreferenz.

update-rest-api

Das folgende Codebeispiel zeigt die Verwendung `update-rest-api`.

AWS CLI

Um den Namen eines zu ändern API

Befehl:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations  
op=replace,path=/name,value='New Name'
```

Um die Beschreibung eines zu ändern API

Befehl:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations  
op=replace,path=/description,value='New Description'
```

- API-Einheiten finden Sie [UpdateRestApi](#) in der AWS CLI Befehlsreferenz.

update-stage

Das folgende Codebeispiel zeigt die Verwendung `update-stage`.

AWS CLI

Beispiel 1: Um die Stufeneinstellungen für eine Ressource und Methode zu überschreiben

Das folgende `update-stage` Beispiel überschreibt die Stufeneinstellungen und deaktiviert die vollständige Protokollierung von Anforderungen/Antworten für eine bestimmte Ressource und Methode.

```
aws apigateway update-stage \  
--rest-api-id 1234123412 \  
--stage-name prod \  
--patch-operations op=replace,path=/logging/enablement,value=false
```

```
--stage-name 'dev' \  
--patch-operations op=replace,path=~1resourceName/GET/logging/  
dataTrace,value=false
```

Ausgabe:

```
{  
  "deploymentId": "5ubd17",  
  "stageName": "dev",  
  "cacheClusterEnabled": false,  
  "cacheClusterStatus": "NOT_AVAILABLE",  
  "methodSettings": {  
    "~1resourceName/GET": {  
      "metricsEnabled": false,  
      "dataTraceEnabled": false,  
      "throttlingBurstLimit": 5000,  
      "throttlingRateLimit": 10000.0,  
      "cachingEnabled": false,  
      "cacheTtlInSeconds": 300,  
      "cacheDataEncrypted": false,  
      "requireAuthorizationForCacheControl": true,  
      "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"  
    }  
  },  
  "tracingEnabled": false,  
  "createdDate": "2022-07-18T10:11:18-07:00",  
  "lastUpdatedDate": "2022-07-18T10:19:04-07:00"  
}
```

Weitere Informationen finden Sie unter [Setting up a stage for a REST API](#) im Amazon API Gateway Developer Guide.

Beispiel 2: Um die Stufeneinstellungen für alle Ressourcen und Methoden einer API Phase zu aktualisieren

Im folgenden `update-stage` Beispiel wird die vollständige Anforderungs-/Antwortprotokollierung für alle Ressourcen und Methoden einer API Phase aktiviert.

```
aws apigateway update-stage \  
--rest-api-id 1234123412 \  
--stage-name 'dev' \  
--patch-operations 'op=replace,path=/*/*/logging/dataTrace,value=true'
```

Ausgabe:

```
{
  "deploymentId": "5ubd17",
  "stageName": "dev",
  "cacheClusterEnabled": false,
  "cacheClusterStatus": "NOT_AVAILABLE",
  "methodSettings": {
    "/*/*": {
      "metricsEnabled": false,
      "dataTraceEnabled": true,
      "throttlingBurstLimit": 5000,
      "throttlingRateLimit": 10000.0,
      "cachingEnabled": false,
      "cacheTtlInSeconds": 300,
      "cacheDataEncrypted": false,
      "requireAuthorizationForCacheControl": true,
      "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"
    }
  },
  "tracingEnabled": false,
  "createdDate": "2022-07-18T10:11:18-07:00",
  "lastUpdatedDate": "2022-07-18T10:31:04-07:00"
}
```

Weitere Informationen finden Sie unter [Setting up a stage for a REST API](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateStage](#) unter AWS CLI Befehlsreferenz.

update-usage-plan

Das folgende Codebeispiel zeigt die Verwendung `update-usage-plan`.

AWS CLI

Um den in einem Nutzungsplan definierten Zeitraum zu ändern

Befehl:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
  op="replace",path="/quota/period",value="MONTH"
```

Um das in einem Nutzungsplan definierte Kontingentlimit zu ändern

Befehl:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/limit",value="500"
```

Um die in einem Nutzungsplan definierte Drosselungsratenbegrenzung zu ändern

Befehl:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/rateLimit",value="10"
```

Um den in einem Nutzungsplan definierten Grenzwert für die Drosselung der Leistungsaufnahme zu ändern

Befehl:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/burstLimit",value="20"
```

- APIEinzelheiten finden Sie [UpdateUsagePlan](#) in der AWS CLI Befehlsreferenz.

update-usage

Das folgende Codebeispiel zeigt die Verwendung `update-usage`.

AWS CLI

Um das Kontingent für einen API Schlüssel für den aktuellen Zeitraum, der im Nutzungsplan definiert ist, vorübergehend zu ändern

Befehl:

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-  
id 1NbjQzMRaAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/  
remaining",value="50"
```

- APIEinzelheiten finden Sie [UpdateUsage](#) unter AWS CLI Befehlsreferenz.

APIGateway HTTP und WebSocket API Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with API Gateway und Aktionen ausführen HTTP und allgemeine Szenarien implementieren WebSocket API.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-api-mapping

Das folgende Codebeispiel zeigt die Verwendung `create-api-mapping`.

AWS CLI

Um ein API Mapping für ein zu erstellen API

Im folgenden `create-api-mapping` Beispiel wird die `test` Phase von API an dem `/myApi` Pfad des `regional.example.com` benutzerdefinierten Domänennamens zugeordnet.

```
aws apigatewayv2 create-api-mapping \  
  --domain-name regional.example.com \  
  --api-mapping-key myApi \  
  --api-id a1b2c3d4 \  
  --stage test
```

Ausgabe:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "0qzs2sy7bh",  
  "ApiMappingKey": "myApi"
```

```
"Stage": "test"
}
```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [CreateApiMapping](#) in der AWS CLI Befehlsreferenz.

create-api

Das folgende Codebeispiel zeigt, wie man es benutzt `create-api`.

AWS CLI

Um eine zu erstellen HTTP API

Im folgenden `create-api` Beispiel wird mithilfe HTTP API von Quick Create ein erstellt. Sie können Quick Create verwenden, um eine API mit AWS Lambda oder einer HTTP Integration, eine Standard-Catch-All-Route und eine Standardphase zu erstellen, die für die automatische Bereitstellung von Änderungen konfiguriert ist. Der folgende Befehl verwendet Quick Create, um eine zu erstellen HTTPAPI, die in eine Lambda-Funktion integriert ist.

```
aws apigatewayv2 create-api \
  --name my-http-api \
  --protocol-type HTTP \
  --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

Ausgabe:

```
{
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
  "ApiId": "a1b2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreateDate": "2020-04-08T19:05:45+00:00",
  "Name": "my-http-api",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path"
}
```

Weitere Informationen finden Sie unter [Entwickeln eines HTTP API API In-Gateways](#) im Amazon API Gateway Developer Guide.

Um ein zu erstellen WebSocket API

Im folgenden `create-api` Beispiel wird eine WebSocket API mit dem angegebenen Namen erstellt.

```
aws apigatewayv2 create-api \  
  --name "myWebSocketApi" \  
  --protocol-type WEBSOCKET \  
  --route-selection-expression '$request.body.action'
```

Ausgabe:

```
{  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "Name": "myWebSocketApi",  
  "CreateDate": "2018-11-15T06:23:51Z",  
  "ProtocolType": "WEBSOCKET",  
  "RouteSelectionExpression": "$request.body.action",  
  "ApiId": "aabbccdee"  
}
```

Weitere Informationen finden Sie unter [Create a WebSocket API in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [CreateApi](#) in der AWS CLI Befehlsreferenz.

create-authorizer

Das folgende Codebeispiel zeigt, wie man es benutzt `create-authorizer`.

AWS CLI

Um einen JWT Autorisierer für einen zu erstellen HTTP API

Im folgenden `create-authorizer` Beispiel wird ein JWT Autorisierer erstellt, der Amazon Cognito als Identitätsanbieter verwendet.

```
aws apigatewayv2 create-authorizer \  
  --name my-jwt-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-type JWT \  
  --identity-provider arn:aws:iam::123456789012:role/MyRole
```



```
--identity-source '$request.header.Authorization' \  
--jwt-configuration Audience=123456abc,Issuer=https://cognito-idp.us-  
west-2.amazonaws.com/us-west-2_abc123
```

Ausgabe:

```
{  
  "AuthorizerId": "a1b2c3",  
  "AuthorizerType": "JWT",  
  "IdentitySource": [  
    "$request.header.Authorization"  
  ],  
  "JwtConfiguration": {  
    "Audience": [  
      "123456abc"  
    ],  
    "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"  
  },  
  "Name": "my-jwt-authorizer"  
}
```

Weitere Informationen finden Sie unter [Steuern des Zugriffs HTTP APIs mit JWT Autorisierern](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie unter [CreateAuthorizer AWS CLI](#) Befehlsreferenz.

create-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `create-deployment`.

AWS CLI

Um ein Deployment für ein zu erstellen API

Im folgenden `create-deployment` Beispiel wird eine Bereitstellung für eine erstellt API und diese Bereitstellung der dev Phase von zugeordnet API.

```
aws apigatewayv2 create-deployment \  
  --api-id a1b2c3d4 \  
  --stage-name dev
```

Ausgabe:

```
{
  "AutoDeployed": false,
  "CreateDate": "2020-04-06T23:38:08Z",
  "DeploymentId": "531z91",
  "DeploymentStatus": "DEPLOYED"
}
```

Weitere Informationen finden Sie unter [APIBereitstellung](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [CreateDeployment](#) in der AWS CLI Befehlsreferenz.

create-domain-name

Das folgende Codebeispiel zeigt, wie man es benutzt `create-domain-name`.

AWS CLI

Um einen benutzerdefinierten Domainnamen zu erstellen

Im folgenden `create-domain-name` Beispiel wird ein regionaler benutzerdefinierter Domainname für eine erstellt API.

```
aws apigatewayv2 create-domain-name \
  --domain-name regional.example.com \
  --domain-name-configurations CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

Ausgabe:

```
{
  "ApiMappingSelectionExpression": "$request.basepath",
  "DomainName": "regional.example.com",
  "DomainNameConfigurations": [
    {
      "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
      "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
      "EndpointType": "REGIONAL",
      "HostedZoneId": "123456789111",
      "SecurityPolicy": "TLS_1_2",
      "DomainNameStatus": "AVAILABLE"
    }
  ]
}
```

```

    }
  ]
}

```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [CreateDomainName](#) in der AWS CLI Befehlsreferenz.

create-integration

Das folgende Codebeispiel zeigt, wie man es benutzt `create-integration`.

AWS CLI

Um eine WebSocket API Integration zu erstellen

Das folgende `create-integration` Beispiel erstellt eine Scheinintegration für WebSocket API a.

```

aws apigatewayv2 create-integration \
  --api-id aabbccdde \
  --passthrough-behavior WHEN_NO_MATCH \
  --timeout-in-millis 29000 \
  --connection-type INTERNET \
  --integration-type MOCK

```

Ausgabe:

```

{
  "ConnectionType": "INTERNET",
  "IntegrationId": "0abcdef",
  "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
  "IntegrationType": "MOCK",
  "PassthroughBehavior": "WHEN_NO_MATCH",
  "PayloadFormatVersion": "1.0",
  "TimeoutInMillis": 29000
}

```

Weitere Informationen finden Sie unter [Einrichten einer WebSocket API Integrationsanfrage in API Gateway](#) im Amazon API Gateway Developer Guide.

Um eine HTTP API Integration zu erstellen

Das folgende `create-integration` Beispiel erstellt eine AWS Lambda-Integration für eine HTTPAPI.

```
aws apigatewayv2 create-integration \  
  --api-id a1b2c3d4 \  
  --integration-type AWS_PROXY \  
  --integration-uri arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --payload-format-version 2.0
```

Ausgabe:

```
{  
  "ConnectionType": "INTERNET",  
  "IntegrationId": "0abcdef",  
  "IntegrationMethod": "POST",  
  "IntegrationType": "AWS_PROXY",  
  "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "PayloadFormatVersion": "2.0",  
  "TimeoutInMillis": 30000  
}
```

Weitere Informationen finden Sie unter [Konfiguration von Integrationen für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [CreateIntegration](#) in der AWS CLI Befehlsreferenz.

create-route

Das folgende Codebeispiel zeigt, wie man es benutzt `create-route`.

AWS CLI

Um eine `$default` Route für ein WebSocket oder zu erstellen HTTP API

Das folgende `create-route` Beispiel erstellt eine `$default` Route für ein WebSocket oder HTTPAPI.

```
aws apigatewayv2 create-route \  
  --api-id aabbccdde \  
  --integration-type AWS_PROXY \  
  --integration-uri arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --payload-format-version 2.0
```

```
--route-key '$default'
```

Ausgabe:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
  "RouteKey": "$default",
  "RouteId": "1122334"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Routen für WebSocket APIs](#) im Amazon API Gateway Developer Guide

Um eine Route für ein zu erstellen HTTP API

Im folgenden `create-route` Beispiel wird eine Route mit dem Namen `signup` that accepting POST requests erstellt.

```
aws apigatewayv2 create-route \
  --api-id aabbccdee \
  --route-key 'POST /signup'
```

Ausgabe:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
  "RouteKey": "POST /signup",
  "RouteId": "1122334"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Routen für HTTP APIs](#) im Amazon API Gateway Developer Guide

- APIEinzelheiten finden Sie [CreateRoute](#) in der AWS CLI Befehlsreferenz.

create-stage

Das folgende Codebeispiel zeigt, wie man es benutzt `create-stage`.

AWS CLI

Um eine Phase zu erstellen

Im folgenden `create-stage` Beispiel wird eine Phase mit dem Namen `dev` für eine erstelltAPI.

```
aws apigatewayv2 create-stage \  
  --api-id a1b2c3d4 \  
  --stage-name dev
```

Ausgabe:

```
{  
  "CreateDate": "2020-04-06T23:23:46Z",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "LastUpdatedDate": "2020-04-06T23:23:46Z",  
  "RouteSettings": {},  
  "StageName": "dev",  
  "StageVariables": {},  
  "Tags": {}  
}
```

Weitere Informationen finden Sie unter [Working with Stages for HTTP APIs](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [CreateStage](#) in der AWS CLI Befehlsreferenz.

`create-vpc-link`

Das folgende Codebeispiel zeigt, wie man es benutzt `create-vpc-link`.

AWS CLI

Um einen VPC Link für einen zu erstellen HTTP API

Das folgende `create-vpc-link` Beispiel erstellt einen VPC Link für HTTP APIs.

```
aws apigatewayv2 create-vpc-link \  
  --name MyVpcLink \  
  --subnet-ids subnet-aaaa subnet-bbbb \  
  --
```

```
--security-group-ids sg1234 sg5678
```

Ausgabe:

```
{
  "CreateDate": "2020-04-07T00:11:46Z",
  "Name": "MyVpcLink",
  "SecurityGroupIds": [
    "sg1234",
    "sg5678"
  ],
  "SubnetIds": [
    "subnet-aaaa",
    "subnet-bbbb"
  ],
  "Tags": {},
  "VpcLinkId": "abcd123",
  "VpcLinkStatus": "PENDING",
  "VpcLinkStatusMessage": "VPC link is provisioning ENIs",
  "VpcLinkVersion": "V2"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit VPC Links für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [CreateVpcLink](#) in der AWS CLI Befehlsreferenz.

delete-access-log-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-access-log-settings`.

AWS CLI

Um die Zugriffsprotokollierung für ein zu deaktivieren API

Im folgenden `delete-access-log-settings` Beispiel werden die Zugriffsprotokolleinstellungen für die `default` Phase von gelöscht. API Um die Zugriffsprotokollierung für eine Phase zu deaktivieren, löschen Sie deren Zugriffsprotokolleinstellungen.

```
aws apigatewayv2 delete-access-log-settings \
  --api-id a1b2c3d4 \
```

```
--stage-name '$default'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfiguration der Protokollierung für an HTTP API](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [DeleteAccessLogSettings](#) in der AWS CLI Befehlsreferenz.

delete-api-mapping

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-api-mapping`.

AWS CLI

Um eine API Zuordnung zu löschen

Im folgenden `delete-api-mapping` Beispiel wird eine API Zuordnung für den `api.example.com` benutzerdefinierten Domännennamen gelöscht.

```
aws apigatewayv2 delete-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [DeleteApiMapping](#) in der AWS CLI Befehlsreferenz.

delete-api

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-api`.

AWS CLI

Um ein zu löschen API

Das folgende `delete-api` Beispiel löscht eine API.

```
aws apigatewayv2 delete-api \  
  --api-id api-id
```



```
--api-id a1b2c3d4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit HTTP APIs](#) und [Arbeiten mit WebSocket APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [DeleteApi](#) in der AWS CLI Befehlsreferenz.

delete-authorizer

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-authorizer`.

AWS CLI

Um einen Autorisierer zu löschen

Im folgenden `delete-authorizer` Beispiel wird ein Autorisierer gelöscht.

```
aws apigatewayv2 delete-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Steuern des Zugriffs HTTP APIs mit JWT Autorisierern](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [DeleteAuthorizer AWS CLI](#) Befehlsreferenz.

delete-cors-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-cors-configuration`.

AWS CLI

Um die CORS Konfiguration für ein zu löschen HTTP API

Im folgenden `delete-cors-configuration` Beispiel wird die Funktion CORS für eine deaktiviert, HTTP API indem die zugehörige CORS Konfiguration gelöscht wird.

```
aws apigatewayv2 delete-cors-configuration \  
  --api-id a1b2c3d4 \  
  --stage-name prod \  
  --cors-id a1b2c3d4
```

```
--api-id a1b2c3d4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfiguration CORS für eine HTTP API](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [DeleteCorsConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-deployment`.

AWS CLI

Um ein Deployment zu löschen

Das folgende `delete-deployment` Beispiel löscht eine Bereitstellung von. API

```
aws apigatewayv2 delete-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id a1b2c3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [APIBereitstellung](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [DeleteDeployment](#) in der AWS CLI Befehlsreferenz.

delete-domain-name

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-domain-name`.

AWS CLI

Um einen benutzerdefinierten Domainnamen zu löschen

Im folgenden `delete-domain-name` Beispiel wird ein benutzerdefinierter Domainname gelöscht.

```
aws apigatewayv2 delete-domain-name \  
  --domain-name api.example.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [DeleteDomainName](#) in der AWS CLI Befehlsreferenz.

delete-integration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-integration`.

AWS CLI

Um eine Integration zu löschen

Das folgende `delete-integration` Beispiel löscht eine API Integration.

```
aws apigatewayv2 delete-integration \  
  --api-id a1b2c3d4 \  
  --integration-id a1b2c3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Integrationen konfigurieren für HTTP APIs](#) und [Einrichten von WebSocket API Integrationen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [DeleteIntegration AWS CLI](#) Befehlsreferenz.

delete-route-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-route-settings`.

AWS CLI

Um Routeneinstellungen zu löschen

Im folgenden `delete-route-settings` Beispiel werden die Routeneinstellungen für die angegebene Route gelöscht.

```
aws apigatewayv2 delete-route-settings \  
  --api-id a1b2c3d4 \  
  --stage-name dev \  
  --route-key 'GET /pets'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Routen für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [DeleteRouteSettings](#) in der AWS CLI Befehlsreferenz.

delete-route

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-route`.

AWS CLI

Um eine Route zu löschen

Im folgenden `delete-route` Beispiel wird eine API Route gelöscht.

```
aws apigatewayv2 delete-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Routen für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [DeleteRoute](#) in der AWS CLI Befehlsreferenz.

delete-stage

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-stage`.

AWS CLI

Um eine Phase zu löschen

Im folgenden `delete-stage` Beispiel wird die `test` Phase eines API gelöscht.

```
aws apigatewayv2 delete-stage \  
  --api-id a1b2c3d4 \  
  --stage-name test
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Working with Stages for HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [DeleteStage](#) in der AWS CLI Befehlsreferenz.

delete-vpc-link

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-vpc-link`.

AWS CLI

Um einen VPC Link für einen zu löschen HTTP API

Im folgenden `delete-vpc-link` Beispiel wird ein VPC Link gelöscht.

```
aws apigatewayv2 delete-vpc-link \  
  --vpc-link-id abcd123
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit VPC Links für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [DeleteVpcLink](#) in der AWS CLI Befehlsreferenz.

export-api

Das folgende Codebeispiel zeigt, wie man es benutzt `export-api`.

AWS CLI

Um eine API Open-Definition von zu exportieren HTTP API

Im folgenden `export-api` Beispiel wird eine Open API 3.0-Definition einer API Stagingstufe mit dem Namen `prod` in eine YAML Datei mit dem Namen `exportiertstage-definition.yaml`. Die exportierte Definitionsdatei enthält standardmäßig API Gateway-Erweiterungen.

```
aws apigatewayv2 export-api \  
  --api-id a1b2c3d4 \  
  --output-type YAML \  
  --specification OAS30 \  
  --stage-name prod \  
  --output-file exportiertstage-definition.yaml
```

stage-definition.yaml

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Exportieren eines HTTP API aus API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [ExportApi](#) in der AWS CLI Befehlsreferenz.

get-api-mapping

Das folgende Codebeispiel zeigt, wie man es benutzt `get-api-mapping`.

AWS CLI

Um Informationen über eine API Zuordnung für einen benutzerdefinierten Domainnamen zu erhalten

Im folgenden `get-api-mapping` Beispiel werden Informationen zu einer API Zuordnung für den `api.example.com` benutzerdefinierten Domainnamen angezeigt.

```
aws apigatewayv2 get-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

Ausgabe:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "a1b2c3d5",  
  "ApiMappingKey": "myTestApi"  
  "Stage": "test"  
}
```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetApiMapping](#) in der AWS CLI Befehlsreferenz.

get-api-mappings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-api-mappings`.

AWS CLI

Um API Zuordnungen für einen benutzerdefinierten Domainnamen abzurufen

Im folgenden `get-api-mappings` Beispiel wird eine Liste aller API Zuordnungen für den `api.example.com` benutzerdefinierten Domänennamen angezeigt.

```
aws apigatewayv2 get-api-mappings \  
  --domain-name api.example.com
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "ApiId": "a1b2c3d4",  
      "ApiMappingId": "a1b2c3d5",  
      "ApiMappingKey": "myTestApi"  
      "Stage": "test"  
    },  
    {  
      "ApiId": "a5b6c7d8",  
      "ApiMappingId": "a1b2c3d6",  
      "ApiMappingKey": "myDevApi"  
      "Stage": "dev"  
    },  
  ],  
}
```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [GetApiMappings](#) in der AWS CLI Befehlsreferenz.

get-api

Das folgende Codebeispiel zeigt, wie man es benutzt `get-api`.

AWS CLI

Um Informationen zu einem abzurufen API

Im folgenden `get-api` Beispiel werden Informationen zu einem angezeigt API.

```
aws apigatewayv2 get-api \  
  --api-id a1b2c3d4
```

Ausgabe:

```
{  
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",  
  "ApiId": "a1b2c3d4",  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "CreateDate": "2020-03-28T00:32:37Z",  
  "Name": "my-api",  
  "ProtocolType": "HTTP",  
  "RouteSelectionExpression": "$request.method $request.path",  
  "Tags": {  
    "department": "finance"  
  }  
}
```

- APIEinzelheiten finden Sie [GetApi](#)unter AWS CLI Befehlsreferenz.

get-apis

Das folgende Codebeispiel zeigt, wie man es benutztget-apis.

AWS CLI

Um eine Liste von abzurufen APIs

Das folgende get-apis Beispiel listet alle APIs für den aktuellen Benutzer auf.

```
aws apigatewayv2 get-apis
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "ApiEndpoint": "wss://a1b2c3d4.execute-api.us-west-2.amazonaws.com",  
      "ApiId": "a1b2c3d4",  
      "ApiKeySelectionExpression": "$request.header.x-api-key",  
      "CreateDate": "2020-04-07T20:21:59Z",
```



```

        "Name": "my-websocket-api",
        "ProtocolType": "WEBSOCKET",
        "RouteSelectionExpression": "$request.body.message",
        "Tags": {}
    },
    {
        "ApiEndpoint": "https://a1b2c3d5.execute-api.us-west-2.amazonaws.com",
        "ApiId": "a1b2c3d5",
        "ApiKeySelectionExpression": "$request.header.x-api-key",
        "CreateDate": "2020-04-07T20:23:50Z",
        "Name": "my-http-api",
        "ProtocolType": "HTTP",
        "RouteSelectionExpression": "$request.method $request.path",
        "Tags": {}
    }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit HTTP APIs](#) und [Arbeiten mit WebSocket APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetApis](#) in der AWS CLI Befehlsreferenz.

get-authorizer

Das folgende Codebeispiel zeigt, wie man es benutzt `get-authorizer`.

AWS CLI

Um Informationen über einen Autorisierer abzurufen

Im folgenden `get-authorizer` Beispiel werden Informationen über einen Autorisierer angezeigt.

```

aws apigatewayv2 get-authorizer \
  --api-id a1b2c3d4 \
  --authorizer-id a1b2c3

```

Ausgabe:

```

{
  "AuthorizerId": "a1b2c3",
  "AuthorizerType": "JWT",
  "IdentitySource": [

```

```

    "$request.header.Authorization"
  ],
  "JwtConfiguration": {
    "Audience": [
      "123456abc"
    ],
    "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
  },
  "Name": "my-jwt-authorizer"
}

```

Weitere Informationen finden Sie unter [Steuern des Zugriffs HTTP APIs mit JWT Autorisierern](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [GetAuthorizer AWS CLI Befehlsreferenz](#).

get-authorizers

Das folgende Codebeispiel zeigt, wie man es benutzt `get-authorizers`.

AWS CLI

Um eine Liste von Autorisierern für ein abzurufen API

Im folgenden `get-authorizers` Beispiel wird eine Liste aller Autorisierer für eine angezeigt. API

```

aws apigatewayv2 get-authorizers \
  --api-id a1b2c3d4

```

Ausgabe:

```

{
  "Items": [
    {
      "AuthorizerId": "a1b2c3",
      "AuthorizerType": "JWT",
      "IdentitySource": [
        "$request.header.Authorization"
      ],
      "JwtConfiguration": {
        "Audience": [
          "123456abc"
        ],

```

```

        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_abc123"
      },
      "Name": "my-jwt-authorizer"
    },
    {
      "AuthorizerId": "a1b2c4",
      "AuthorizerType": "JWT",
      "IdentitySource": [
        "$request.header.Authorization"
      ],
      "JwtConfiguration": {
        "Audience": [
          "6789abcde"
        ],
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_abc234"
      },
      "Name": "new-jwt-authorizer"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Steuern des Zugriffs HTTP APIs mit JWT Autorisierern](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [GetAuthorizers AWS CLI Befehlsreferenz](#).

get-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `get-deployment`.

AWS CLI

Um Informationen über eine Bereitstellung abzurufen

Im folgenden `get-deployment` Beispiel werden Informationen zu einer Bereitstellung angezeigt.

```

aws apigatewayv2 get-deployment \
  --api-id a1b2c3d4 \
  --deployment-id abcdef

```

Ausgabe:

```
{
  "AutoDeployed": true,
  "CreateDate": "2020-04-07T23:58:40Z",
  "DeploymentId": "abcdef",
  "DeploymentStatus": "DEPLOYED",
  "Description": "Automatic deployment triggered by changes to the Api
configuration"
}
```

Weitere Informationen finden Sie unter [APIBereitstellung](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [GetDeployment](#) in der AWS CLI Befehlsreferenz.

get-deployments

Das folgende Codebeispiel zeigt, wie man es benutzt `get-deployments`.

AWS CLI

Um eine Liste von Bereitstellungen abzurufen

Im folgenden `get-deployments` Beispiel wird eine Liste aller Bereitstellungen API von an angezeigt.

```
aws apigatewayv2 get-deployments \
  --api-id a1b2c3d4
```

Ausgabe:

```
{
  "Items": [
    {
      "AutoDeployed": true,
      "CreateDate": "2020-04-07T23:58:40Z",
      "DeploymentId": "abcdef",
      "DeploymentStatus": "DEPLOYED",
      "Description": "Automatic deployment triggered by changes to the Api
configuration"
    },
    {
      "AutoDeployed": true,
```

```

        "CreatedDate": "2020-04-06T00:33:00Z",
        "DeploymentId": "bcdefg",
        "DeploymentStatus": "DEPLOYED",
        "Description": "Automatic deployment triggered by changes to the Api
configuration"
      }
    ]
  }

```

Weitere Informationen finden Sie unter [APIBereitstellung](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie [GetDeployments](#) in der AWS CLI Befehlsreferenz.

get-domain-name

Das folgende Codebeispiel zeigt, wie man es benutzt `get-domain-name`.

AWS CLI

Um Informationen zu einem benutzerdefinierten Domainnamen abzurufen

Im folgenden `get-domain-name` Beispiel werden Informationen zu einem benutzerdefinierten Domainnamen angezeigt.

```

aws apigatewayv2 get-domain-name \
  --domain-name api.example.com

```

Ausgabe:

```

{
  "ApiMappingSelectionExpression": "$request.basepath",
  "DomainName": "api.example.com",
  "DomainNameConfigurations": [
    {
      "ApiGatewayDomainName": "d-1234.execute-api.us-west-2.amazonaws.com",
      "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
      "EndpointType": "REGIONAL",
      "HostedZoneId": "123456789111",
      "SecurityPolicy": "TLS_1_2",
      "DomainNameStatus": "AVAILABLE"
    }
  ]
}

```

```

    }
  ],
  "Tags": {}
}

```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetDomainName](#) in der AWS CLI Befehlsreferenz.

get-domain-names

Das folgende Codebeispiel zeigt, wie man es benutzt `get-domain-names`.

AWS CLI

Um eine Liste mit benutzerdefinierten Domainnamen abzurufen

Im folgenden `get-domain-names` Beispiel wird eine Liste aller benutzerdefinierten Domännennamen für den aktuellen Benutzer angezeigt.

```
aws apigatewayv2 get-domain-names
```

Ausgabe:

```

{
  "Items": [
    {
      "ApiMappingSelectionExpression": "$request.basepath",
      "DomainName": "api.example.com",
      "DomainNameConfigurations": [
        {
          "ApiGatewayDomainName": "d-1234.execute-api.us-west-2.amazonaws.com",
          "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
          "EndpointType": "REGIONAL",
          "HostedZoneId": "123456789111",
          "SecurityPolicy": "TLS_1_2",
          "DomainNameStatus": "AVAILABLE"
        }
      ]
    }
  ],
}

```

```

    {
      "ApiMappingSelectionExpression": "$request.basepath",
      "DomainName": "newApi.example.com",
      "DomainNameConfigurations": [
        {
          "ApiGatewayDomainName": "d-5678.execute-api.us-
west-2.amazonaws.com",
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
          "EndpointType": "REGIONAL",
          "HostedZoneId": "123456789222",
          "SecurityPolicy": "TLS_1_2",
          "DomainNameStatus": "AVAILABLE"
        }
      ]
    }
  ]
}

```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetDomainNames](#) in der AWS CLI Befehlsreferenz.

get-integration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-integration`.

AWS CLI

Um Informationen über eine Integration abzurufen

Im folgenden `get-integration` Beispiel werden Informationen zu einer Integration angezeigt.

```

aws apigatewayv2 get-integration \
  --api-id a1b2c3d4 \
  --integration-id a1b2c3

```

Ausgabe:

```

{
  "ApiGatewayManaged": true,
  "ConnectionType": "INTERNET",

```

```
"IntegrationId": "a1b2c3",
"IntegrationMethod": "POST",
"IntegrationType": "AWS_PROXY",
"IntegrationUri": "arn:aws:lambda:us-west-2:12356789012:function:hello12",
"PayloadFormatVersion": "2.0",
"TimeoutInMillis": 30000
}
```

Weitere Informationen finden Sie unter [Integrationen konfigurieren für HTTP APIs](#) und [Einrichten von WebSocket API Integrationen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [GetIntegration AWS CLI](#) Befehlsreferenz.

get-integrations

Das folgende Codebeispiel zeigt, wie man es benutzt `get-integrations`.

AWS CLI

Um eine Liste von Integrationen abzurufen

Das folgende `get-integrations` Beispiel zeigt eine Liste aller Integrationen API von an.

```
aws apigatewayv2 get-integrations \
  --api-id a1b2c3d4
```

Ausgabe:

```
{
  "Items": [
    {
      "ApiGatewayManaged": true,
      "ConnectionType": "INTERNET",
      "IntegrationId": "a1b2c3",
      "IntegrationMethod": "POST",
      "IntegrationType": "AWS_PROXY",
      "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
      "PayloadFormatVersion": "2.0",
      "TimeoutInMillis": 30000
    },
    {
      "ConnectionType": "INTERNET",
```



```
        "IntegrationId": "a1b2c4",
        "IntegrationMethod": "ANY",
        "IntegrationType": "HTTP_PROXY",
        "IntegrationUri": "https://www.example.com",
        "PayloadFormatVersion": "1.0",
        "TimeoutInMillis": 30000
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Integrationen konfigurieren für HTTP APIs](#) und [Einrichten von WebSocket API Integrationen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [GetIntegrations AWS CLI](#) Befehlsreferenz.

get-route

Das folgende Codebeispiel zeigt, wie man es benutzt `get-route`.

AWS CLI

Um Informationen über eine Route abzurufen

Im folgenden `get-route` Beispiel werden Informationen zu einer Route angezeigt.

```
aws apigatewayv2 get-route \
  --api-id a1b2c3d4 \
  --route-id 72jz1wk
```

Ausgabe:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
  "RouteId": "72jz1wk",
  "RouteKey": "ANY /pets",
  "Target": "integrations/a1b2c3"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Routen für HTTP APIs](#) im Amazon API Gateway Developer Guide

- API-Einheiten finden Sie [GetRoute](#) in der AWS CLI Befehlsreferenz.

get-routes

Das folgende Codebeispiel zeigt, wie man es benutzt `get-routes`.

AWS CLI

Um eine Liste von Routen abzurufen

Im folgenden `get-routes` Beispiel wird eine Liste aller Routen eines API Benutzers angezeigt.

```
aws apigatewayv2 get-routes \  
  --api-id a1b2c3d4
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "ApiKeyRequired": false,  
      "AuthorizationType": "NONE",  
      "RouteId": "72jz1wk",  
      "RouteKey": "ANY /admin",  
      "Target": "integrations/a1b2c3"  
    },  
    {  
      "ApiGatewayManaged": true,  
      "ApiKeyRequired": false,  
      "AuthorizationType": "NONE",  
      "RouteId": "go65gqi",  
      "RouteKey": "$default",  
      "Target": "integrations/a1b2c4"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Routen für HTTP APIs](#) im Amazon API Gateway Developer Guide

- API-Einheiten finden Sie [GetRoutes](#) in der AWS CLI Befehlsreferenz.

get-stage

Das folgende Codebeispiel zeigt, wie man es benutzt `get-stage`.

AWS CLI

Um Informationen über eine Phase abzurufen

Im folgenden `get-stage` Beispiel werden Informationen zur `prod` Phase eines angezeigt API.

```
aws apigatewayv2 get-stage \  
  --api-id a1b2c3d4 \  
  --stage-name prod
```

Ausgabe:

```
{  
  "CreateDate": "2020-04-08T00:36:05Z",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "DeploymentId": "x1zwyv",  
  "LastUpdatedDate": "2020-04-08T00:36:13Z",  
  "RouteSettings": {},  
  "StageName": "prod",  
  "StageVariables": {  
    "function": "my-prod-function"  
  },  
  "Tags": {}  
}
```

Weitere Informationen finden Sie unter [Working with Stages for HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API Einheiten finden Sie [GetStage](#) in der AWS CLI Befehlsreferenz.

get-stages

Das folgende Codebeispiel zeigt, wie man es benutzt `get-stages`.

AWS CLI

Um eine Liste von Stufen abzurufen

Das folgende `get-stages` Beispiel listet alle Phasen API von an auf.

```
aws apigatewayv2 get-stages \  
  --api-id a1b2c3d4
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "ApiGatewayManaged": true,  
      "AutoDeploy": true,  
      "CreateDate": "2020-04-08T00:08:44Z",  
      "DefaultRouteSettings": {  
        "DetailedMetricsEnabled": false  
      },  
      "DeploymentId": "dty748",  
      "LastDeploymentStatusMessage": "Successfully deployed stage with  
deployment ID 'dty748'",  
      "LastUpdatedDate": "2020-04-08T00:09:49Z",  
      "RouteSettings": {},  
      "StageName": "$default",  
      "StageVariables": {},  
      "Tags": {}  
    },  
    {  
      "AutoDeploy": true,  
      "CreateDate": "2020-04-08T00:35:06Z",  
      "DefaultRouteSettings": {  
        "DetailedMetricsEnabled": false  
      },  
      "LastUpdatedDate": "2020-04-08T00:35:48Z",  
      "RouteSettings": {},  
      "StageName": "dev",  
      "StageVariables": {  
        "function": "my-dev-function"  
      },  
      "Tags": {}  
    },  
    {  
      "CreateDate": "2020-04-08T00:36:05Z",  
      "DefaultRouteSettings": {  
        "DetailedMetricsEnabled": false  
      },  
      "DeploymentId": "x1zwyv",
```

```

        "LastUpdatedDate": "2020-04-08T00:36:13Z",
        "RouteSettings": {},
        "StageName": "prod",
        "StageVariables": {
            "function": "my-prod-function"
        },
        "Tags": {}
    }
]
}

```

Weitere Informationen finden Sie unter [Working with Stages for HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetStages](#) in der AWS CLI Befehlsreferenz.

get-tags

Das folgende Codebeispiel zeigt, wie man es benutzt `get-tags`.

AWS CLI

Um eine Liste von Tags für eine Ressource abzurufen

Das folgende `get-tags` Beispiel listet alle Tags einer API Person auf.

```

aws apigatewayv2 get-tags \
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4

```

Ausgabe:

```

{
  "Tags": {
    "owner": "dev-team",
    "environment": "prod"
  }
}

```

Weitere Informationen finden Sie unter [Tagging your API Gateway-Ressourcen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetTags](#) in der AWS CLI Befehlsreferenz.

get-vpc-link

Das folgende Codebeispiel zeigt, wie man es benutzt `get-vpc-link`.

AWS CLI

Um Informationen über einen VPC Link abzurufen

Im folgenden `get-vpc-link` Beispiel werden Informationen zu einem VPC Link angezeigt.

```
aws apigatewayv2 get-vpc-link \
  --vpc-link-id abcd123
```

Ausgabe:

```
{
  "CreateDate": "2020-04-07T00:27:47Z",
  "Name": "MyVpcLink",
  "SecurityGroupIds": [
    "sg1234",
    "sg5678"
  ],
  "SubnetIds": [
    "subnet-aaaa",
    "subnet-bbbb"
  ],
  "Tags": {},
  "VpcLinkId": "abcd123",
  "VpcLinkStatus": "AVAILABLE",
  "VpcLinkStatusMessage": "VPC link is ready to route traffic",
  "VpcLinkVersion": "V2"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit VPC Links für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetVpcLink](#) in der AWS CLI Befehlsreferenz.

get-vpc-links

Das folgende Codebeispiel zeigt, wie man es benutzt `get-vpc-links`.

AWS CLI

Um eine Liste von VPC Links abzurufen

Im folgenden `get-vpc-links` Beispiel wird eine Liste aller VPC Links für den aktuellen Benutzer angezeigt.

```
aws apigatewayv2 get-vpc-links
```

Ausgabe:

```
{
  "Items": [
    {
      "CreateDate": "2020-04-07T00:27:47Z",
      "Name": "MyVpcLink",
      "SecurityGroupIds": [
        "sg1234",
        "sg5678"
      ],
      "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
      ],
      "Tags": {},
      "VpcLinkId": "abcd123",
      "VpcLinkStatus": "AVAILABLE",
      "VpcLinkStatusMessage": "VPC link is ready to route traffic",
      "VpcLinkVersion": "V2"
    }
  ],
  {
    "CreateDate": "2020-04-07T00:27:47Z",
    "Name": "MyOtherVpcLink",
    "SecurityGroupIds": [
      "sg1234",
      "sg5678"
    ],
    "SubnetIds": [
      "subnet-aaaa",
      "subnet-bbbb"
    ],
    "Tags": {},
    "VpcLinkId": "abcd456",
```

```

        "VpcLinkStatus": "AVAILABLE",
        "VpcLinkStatusMessage": "VPC link is ready to route traffic",
        "VpcLinkVersion": "V2"
    }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit VPC Links für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [GetVpcLinks](#) in der AWS CLI Befehlsreferenz.

import-api

Das folgende Codebeispiel zeigt, wie man es benutzt `import-api`.

AWS CLI

Um ein zu importieren HTTP API

Im folgenden `import-api` Beispiel wird HTTP API aus einer Open API 3.0-Definitionsdatei eine Datei mit dem Namen `erstelltapi-definition.yaml`.

```

aws apigatewayv2 import-api \
  --body file://api-definition.yaml

```

Inhalt von `api-definition.yaml`:

```

openapi: 3.0.1
info:
  title: My Lambda API
  version: v1.0
paths:
  /hello:
    x-amazon-apigateway-any-method:
      x-amazon-apigateway-integration:
        payloadFormatVersion: 2.0
        type: aws_proxy
        httpMethod: POST
        uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123456789012:function:hello/invocations
        connectionType: INTERNET

```


Ausgabe:

```
{
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
  "ApiId": "a1b2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreateDate": "2020-04-08T17:19:38+00:00",
  "Name": "My Lambda API",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path",
  "Tags": {},
  "Version": "v1.0"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit offenen API Definitionen für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [ImportApi](#) in der AWS CLI Befehlsreferenz.

reimport-api

Das folgende Codebeispiel zeigt, wie man es benutzt `reimport-api`.

AWS CLI

Um eine erneut zu importieren HTTP API

Im folgenden `reimport-api` Beispiel wird ein vorhandenes aktualisiert HTTPAPI, sodass es die in `api-definition.yaml` angegebene Open API 3.0-Definition verwendet.

```
aws apigatewayv2 reimport-api \
  --body file://api-definition.yaml \
  --api-id a1b2c3d4
```

Inhalt von `api-definition.yaml`:

```
openapi: 3.0.1
info:
  title: My Lambda API
  version: v1.0
paths:
  /hello:
```

```
x-amazon-apigateway-any-method:
  x-amazon-apigateway-integration:
    payloadFormatVersion: 2.0
    type: aws_proxy
    httpMethod: POST
    uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:12356789012:function:hello/invocations
    connectionType: INTERNET
```

Ausgabe:

```
{
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
  "ApiId": "a1b2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreateDate": "2020-04-08T17:19:38+00:00",
  "Name": "My Lambda API",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path",
  "Tags": {},
  "Version": "v1.0"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit offenen API Definitionen für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [ReimportApi](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird ein Tag mit dem Schlüsselnamen `Department` und dem angegebenen Wert hinzugefügt `API.Accounting`

```
aws apigatewayv2 tag-resource \
  --resource-arn arn:aws:apigateway:us-west-2:apis/a1b2c3d4 \
  --tags Department=Accounting
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging your API Gateway-Ressourcen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden Tags mit den Schlüsselnamen `Project` und `Owner` aus den angegebenen Schlüsselnamen entfernt API.

```
aws apigatewayv2 untag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tag-keys Project Owner
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging your API Gateway-Ressourcen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-api-mapping

Das folgende Codebeispiel zeigt, wie man es benutzt `update-api-mapping`.

AWS CLI

Um ein API Mapping zu aktualisieren

Im folgenden `update-api-mapping` Beispiel wird eine API Zuordnung für einen benutzerdefinierten Domainnamen geändert. Dadurch wird die Basisstufe, die den URL benutzerdefinierten Domainnamen für die angegebene API Endstufe verwendet, zu `https://api.example.com/dev`.

```
aws apigatewayv2 update-api-mapping \  
  --api-id a1b2c3d4 \  
  --stage dev \  
  --domain-name api.example.com \  
  --api-mapping-id 0qzs2sy7bh \  
  --api-mapping-key dev
```

Ausgabe:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "0qzs2sy7bh",  
  "ApiMappingKey": "dev"  
  "Stage": "dev"  
}
```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateApiMapping](#) in der AWS CLI Befehlsreferenz.

update-api

Das folgende Codebeispiel zeigt, wie man es benutzt `update-api`.

AWS CLI

Zur Aktivierung CORS für ein HTTP API

Im folgenden `update-api` Beispiel wird die angegebene API CORS Konfiguration aktualisiert, sodass Anfragen von möglich sind `https://www.example.com`.

```
aws apigatewayv2 update-api \  
  --api-id a1b2c3d4 \  
  --cors-configuration AllowOrigins=https://www.example.com
```

Ausgabe:

```
{  
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
```

```

"ApiId": "a1b2c3d4",
"ApiKeySelectionExpression": "$request.header.x-api-key",
"CorsConfiguration": {
  "AllowCredentials": false,
  "AllowHeaders": [
    "header1",
    "header2"
  ],
  "AllowMethods": [
    "GET",
    "OPTIONS"
  ],
  "AllowOrigins": [
    "https://www.example.com"
  ]
},
"CreateDate": "2020-04-08T18:39:37+00:00",
"Name": "my-http-api",
"ProtocolType": "HTTP",
"RouteSelectionExpression": "$request.method $request.path",
"Tags": {},
"Version": "v1.0"
}

```

Weitere Informationen finden Sie unter [Konfiguration CORS für eine HTTP API](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateApi](#) in der AWS CLI Befehlsreferenz.

update-authorizer

Das folgende Codebeispiel zeigt, wie man es benutzt `update-authorizer`.

AWS CLI

Um einen Autorisierer zu aktualisieren

Im folgenden `update-authorizer` Beispiel wird die Identitätsquelle eines JWT Autorisierers in einen Header mit dem Namen geändert. `Authorization`

```

aws apigatewayv2 update-authorizer \
  --api-id a1b2c3d4 \
  --authorizer-id a1b2c3 \

```

```
--identity-source '$request.header.Authorization'
```

Ausgabe:

```
{
  "AuthorizerId": "a1b2c3",
  "AuthorizerType": "JWT",
  "IdentitySource": [
    "$request.header.Authorization"
  ],
  "JwtConfiguration": {
    "Audience": [
      "123456abc"
    ],
    "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"
  },
  "Name": "my-jwt-authorizer"
}
```

Weitere Informationen finden Sie unter [Steuern des Zugriffs HTTP APIs mit JWT Autorisierern](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [UpdateAuthorizer AWS CLI Befehlsreferenz](#).

update-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `update-deployment`.

AWS CLI

Um die Beschreibung einer Bereitstellung zu ändern

Im folgenden `update-deployment` Beispiel wird die Beschreibung einer Bereitstellung aktualisiert.

```
aws apigatewayv2 update-deployment \
  --api-id a1b2c3d4 \
  --deployment-id abcdef \
  --description 'Manual deployment to fix integration test failures.'
```

Ausgabe:

```
{
  "AutoDeployed": false,
  "CreateDate": "2020-02-05T16:21:48+00:00",
  "DeploymentId": "abcdef",
  "DeploymentStatus": "DEPLOYED",
  "Description": "Manual deployment to fix integration test failures."
}
```

Weitere Informationen finden Sie unter [Entwickeln eines HTTP API API In-Gateways](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateDeployment](#) in der AWS CLI Befehlsreferenz.

update-domain-name

Das folgende Codebeispiel zeigt, wie man es benutzt `update-domain-name`.

AWS CLI

Um einen benutzerdefinierten Domainnamen zu aktualisieren

Das folgende `update-domain-name` Beispiel spezifiziert ein neues ACM Zertifikat für den `api.example.com` benutzerdefinierten Domainnamen.

```
aws apigatewayv2 update-domain-name \
  --domain-name api.example.com \
  --domain-name-configurations CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

Ausgabe:

```
{
  "ApiMappingSelectionExpression": "$request.basepath",
  "DomainName": "regional.example.com",
  "DomainNameConfigurations": [
    {
      "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
      "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
      "EndpointType": "REGIONAL",
      "HostedZoneId": "123456789111",
    }
  ]
}
```

```
        "SecurityPolicy": "TLS_1_2",
        "DomainNameStatus": "AVAILABLE"
    }
]
}
```

Weitere Informationen finden Sie unter [Einrichtung eines regionalen benutzerdefinierten Domainnamens in API Gateway](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateDomainName](#) in der AWS CLI Befehlsreferenz.

update-integration

Das folgende Codebeispiel zeigt, wie man es benutzt `update-integration`.

AWS CLI

Um eine Lambda-Integration zu aktualisieren

Im folgenden `update-integration` Beispiel wird eine bestehende AWS Lambda-Integration aktualisiert, sodass sie die angegebene Lambda-Funktion verwendet.

```
aws apigatewayv2 update-integration \
  --api-id a1b2c3d4 \
  --integration-id a1b2c3 \
  --integration-uri arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123456789012:function:my-new-function/invocations
```

Ausgabe:

```
{
  "ConnectionType": "INTERNET",
  "IntegrationId": "a1b2c3",
  "IntegrationMethod": "POST",
  "IntegrationType": "AWS_PROXY",
  "IntegrationUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123456789012:function:my-new-function/
invocations",
  "PayloadFormatVersion": "2.0",
  "TimeoutInMillis": 5000
}
```


Weitere Informationen finden Sie unter [Integrationen konfigurieren für HTTP APIs](#) und [Einrichten von WebSocket API Integrationen](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [UpdateIntegration AWS CLI](#) Befehlsreferenz.

update-route

Das folgende Codebeispiel zeigt, wie man es benutzt `update-route`.

AWS CLI

Beispiel 1: Um die Integration einer Route zu aktualisieren

Das folgende `update-route` Beispiel aktualisiert die Integration einer angegebenen Route.

```
aws apigatewayv2 update-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3 \  
  --target integrations/a1b2c6
```

Ausgabe:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteId": "a1b2c3",  
  "RouteKey": "ANY /pets",  
  "Target": "integrations/a1b2c6"  
}
```

Beispiel 2: Um einer Route einen Autorisierer hinzuzufügen

Im folgenden `update-route` Beispiel wird die angegebene Route aktualisiert, sodass sie einen JWT Autorisierer verwendet.

```
aws apigatewayv2 update-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3 \  
  --authorization-type JWT \  
  --authorizer-id a1b2c5 \  
  --authorization-scopes user.id user.email
```

Ausgabe:

```
{
  "ApiKeyRequired": false,
  "AuthorizationScopes": [
    "user.id",
    "user.email"
  ],
  "AuthorizationType": "JWT",
  "AuthorizerId": "a1b2c5",
  "OperationName": "GET HTTP",
  "RequestParameters": {},
  "RouteId": "a1b2c3",
  "RouteKey": "GET /pets",
  "Target": "integrations/a1b2c6"
}
```

Weitere Informationen finden Sie unter [Steuern des Zugriffs HTTP APIs mit JWT Autorisierern](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [UpdateRoute AWS CLI](#) Befehlsreferenz.

update-stage

Das folgende Codebeispiel zeigt, wie man es benutzt `update-stage`.

AWS CLI

Um eine benutzerdefinierte Drosselung zu konfigurieren

Im folgenden `update-stage` Beispiel wird die benutzerdefinierte Drosselung für die angegebene Phase und Route eines konfiguriert. API

```
aws apigatewayv2 update-stage \
  --api-id a1b2c3d4 \
  --stage-name dev \
  --route-settings '{"GET /pets":
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

Ausgabe:

```
{
```

```

    "CreateDate": "2020-04-05T16:21:16+00:00",
    "DefaultRouteSettings": {
      "DetailedMetricsEnabled": false
    },
    "DeploymentId": "shktxb",
    "LastUpdatedDate": "2020-04-08T22:23:17+00:00",
    "RouteSettings": {
      "GET /pets": {
        "ThrottlingBurstLimit": 100,
        "ThrottlingRateLimit": 2000.0
      }
    },
    "StageName": "dev",
    "StageVariables": {},
    "Tags": {}
  }
}

```

Weitere Informationen finden Sie unter [Protecting your HTTP API](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateStage](#) in der AWS CLI Befehlsreferenz.

update-vpc-link

Das folgende Codebeispiel zeigt, wie man es benutzt `update-vpc-link`.

AWS CLI

Um einen VPC Link zu aktualisieren

Im folgenden `update-vpc-link` Beispiel wird der Name eines VPC Links aktualisiert. Nachdem Sie einen VPC Link erstellt haben, können Sie seine Sicherheitsgruppen oder Subnetze nicht mehr ändern.

```

aws apigatewayv2 update-vpc-link \
  --vpc-link-id abcd123 \
  --name MyUpdatedVpcLink

```

Ausgabe:

```

{
  "CreateDate": "2020-04-07T00:27:47Z",

```

```
"Name": "MyUpdatedVpcLink",
"SecurityGroupIds": [
  "sg1234",
  "sg5678"
],
"SubnetIds": [
  "subnet-aaaa",
  "subnet-bbbb"
],
"Tags": {},
"VpcLinkId": "abcd123",
"VpcLinkStatus": "AVAILABLE",
"VpcLinkStatusMessage": "VPC link is ready to route traffic",
"VpcLinkVersion": "V2"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit VPC Links für HTTP APIs](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie [UpdateVpcLink](#) in der AWS CLI Befehlsreferenz.

API Gateway-Management-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit API Gateway Management Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-connection`.

AWS CLI

Um eine WebSocket Verbindung zu löschen

Im folgenden `delete-connection` Beispiel wird die Verbindung eines Clients mit der angegebenen WebSocket API Verbindung getrennt.

```
aws apigatewaymanagementapi delete-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von @connections -Befehlen in Ihrem Backend-Service](#) im Amazon API Gateway Developer Guide.

- APIEinzelheiten finden Sie unter [DeleteConnection AWS CLI](#) Befehlsreferenz.

get-connection

Das folgende Codebeispiel zeigt die Verwendung `get-connection`.

AWS CLI

Um Informationen über eine WebSocket Verbindung zu erhalten

Das folgende `get-connection` Beispiel beschreibt eine Verbindung zu der angegebenen WebSocket API.

```
aws apigatewaymanagementapi get-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Ausgabe:

```
{  
  "ConnectedAt": "2020-04-30T20:10:33.236Z",  
  "Identity": {  
    "SourceIp": "192.0.2.1"  
  },  
  "LastActiveAt": "2020-04-30T20:10:42.997Z"  
}
```

Weitere Informationen finden Sie unter [Verwenden von @connections -Befehlen in Ihrem Backend-Service](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [GetConnection AWS CLI](#) Befehlsreferenz.

post-to-connection

Das folgende Codebeispiel zeigt die Verwendung `post-to-connection`.

AWS CLI

Um Daten an eine WebSocket-Verbindung zu senden

Im folgenden `post-to-connection` Beispiel wird eine Nachricht an einen Client gesendet, der mit der angegebenen Verbindung verbunden ist. WebSocket API.

```
aws apigatewaymanagementapi post-to-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --data "Hello from API Gateway!" \  
  --endpoint-url https://aabbccdde.execute-api.us-west-2.amazonaws.com/prod
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von @connections -Befehlen in Ihrem Backend-Service](#) im Amazon API Gateway Developer Guide.

- API-Einheiten finden Sie unter [PostToConnection AWS CLI](#) Befehlsreferenz.

App Mesh Mesh-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit App Mesh Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-mesh

Das folgende Codebeispiel zeigt die Verwendung `create-mesh`.

AWS CLI

Beispiel 1: Um ein neues Service Mesh zu erstellen

Das folgende `create-mesh` Beispiel erstellt ein Service Mesh.

```
aws appmesh create-mesh \  
  --mesh-name app1
```

Ausgabe:

```
{  
  "mesh": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",  
      "createdAt": 1563809909.282,  
      "lastUpdatedAt": 1563809909.282,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "spec": {},  
    "status": {  
      "status": "ACTIVE"  
    }  
  }  
}
```

Beispiel 2: Um ein neues Service Mesh mit mehreren Tags zu erstellen

Das folgende `create-mesh` Beispiel erstellt ein Service Mesh mit mehreren Tags.

```
aws appmesh create-mesh \  
  --mesh-name app1
```

```
--mesh-name app2 \  
--tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Ausgabe:

```
{  
  "mesh":{  
    "meshName":"app2",  
    "metadata":{  
      "arn":"arn:aws:appmesh:us-east-1:123456789012:mesh/app2",  
      "createdAt":1563822121.877,  
      "lastUpdatedAt":1563822121.877,  
      "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version":1  
    },  
    "spec":{ },  
    "status":{  
      "status":"ACTIVE"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Service Meshes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateMesh](#) in der AWS CLI Befehlsreferenz.

create-route

Das folgende Codebeispiel zeigt die Verwendung `create-route`.

AWS CLI

Um eine neue RPC G-Route zu erstellen

Das folgende `create-route` Beispiel verwendet eine JSON Eingabedatei, um eine RPC G-Route zu erstellen. GRPC-Datenverkehr mit Metadaten, die mit 123 beginnen, wird an einen virtuellen Knoten mit dem Namen `serviceBgrpc` weitergeleitet. Wenn beim Versuch RPC-HTTP, mit dem Ziel der Route zu kommunizieren, bestimmte G-, oder TCP Fehler auftreten, wird die Route dreimal wiederholt. Zwischen jedem Wiederholungsversuch liegt eine Verzögerung von 15 Sekunden.


```
aws appmesh create-route \  
--cli-input-json file://create-route-grpc.json
```

Inhalt von `create-route-grpc.json`:

```
{  
  "meshName" : "apps",  
  "routeName" : "grpcRoute",  
  "spec" : {  
    "grpcRoute" : {  
      "action" : {  
        "weightedTargets" : [  
          {  
            "virtualNode" : "serviceBgrpc",  
            "weight" : 100  
          }  
        ]  
      },  
      "match" : {  
        "metadata" : [  
          {  
            "invert" : false,  
            "match" : {  
              "prefix" : "123"  
            },  
            "name" : "myMetadata"  
          }  
        ],  
        "methodName" : "GetColor",  
        "serviceName" : "com.amazonaws.services.ColorService"  
      },  
      "retryPolicy" : {  
        "grpcRetryEvents" : [ "deadline-exceeded" ],  
        "httpRetryEvents" : [ "server-error", "gateway-error" ],  
        "maxRetries" : 3,  
        "perRetryTimeout" : {  
          "unit" : "s",  
          "value" : 15  
        },  
        "tcpRetryEvents" : [ "connection-error" ]  
      }  
    },  
    "priority" : 100  
  }  
}
```

```

    },
    "virtualRouterName" : "serviceBgrpc"
  }

```

Ausgabe:

```

{
  "route": {
    "meshName": "apps",
    "metadata": {
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBgrpc/route/grpcRoute",
      "createdAt": 1572010806.008,
      "lastUpdatedAt": 1572010806.008,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "routeName": "grpcRoute",
    "spec": {
      "grpcRoute": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "serviceBgrpc",
              "weight": 100
            }
          ]
        },
        "match": {
          "metadata": [
            {
              "invert": false,
              "match": {
                "prefix": "123"
              },
              "name": "mymetadata"
            }
          ],
          "methodName": "GetColor",
          "serviceName": "com.amazonaws.services.ColorService"
        },
        "retryPolicy": {
          "grpcRetryEvents": [

```

```

        "deadline-exceeded"
    ],
    "httpRetryEvents": [
        "server-error",
        "gateway-error"
    ],
    "maxRetries": 3,
    "perRetryTimeout": {
        "unit": "s",
        "value": 15
    },
    "tcpRetryEvents": [
        "connection-error"
    ]
    }
    },
    "priority": 100
},
"status": {
    "status": "ACTIVE"
},
"virtualRouterName": "serviceBgrpc"
}
}

```

Um eine neue HTTP oder HTTP /2 Route zu erstellen

Das folgende `create-route` Beispiel verwendet eine JSON Eingabedatei, um eine HTTP /2-Route zu erstellen. Um eine HTTP Route zu erstellen, ersetzen Sie `Http2Route` durch `httpRoute` under `spec`. Der gesamte HTTP /2-Verkehr, der an ein URL Präfix adressiert ist, dessen Header-Wert mit 123 beginnt, wird an einen virtuellen Knoten mit dem Namen 2 weitergeleitet. `serviceBhttp` Wenn beim Versuch, mit dem Ziel der Route zu kommunizieren, bestimmte HTTP oder TCP Fehler auftreten, wird die Route dreimal wiederholt. Zwischen jedem erneuten Versuch liegt eine Verzögerung von 15 Sekunden.

```

aws appmesh create-route \
  --cli-input-json file://create-route-http2.json

```

Inhalt von `create-route-http2.json`:

```

{
  "meshName": "apps",

```

```
"routeName": "http2Route",
"spec": {
  "http2Route": {
    "action": {
      "weightedTargets": [
        {
          "virtualNode": "serviceBhttp2",
          "weight": 100
        }
      ]
    },
    "match": {
      "headers": [
        {
          "invert": false,
          "match": {
            "prefix": "123"
          },
          "name": "clientRequestId"
        }
      ],
      "method": "POST",
      "prefix": "/",
      "scheme": "http"
    },
    "retryPolicy": {
      "httpRetryEvents": [
        "server-error",
        "gateway-error"
      ],
      "maxRetries": 3,
      "perRetryTimeout": {
        "unit": "s",
        "value": 15
      },
      "tcpRetryEvents": [
        "connection-error"
      ]
    }
  },
  "priority": 200
},
"virtualRouterName": "serviceBhttp2"
```

```
}
```

Ausgabe:

```
{
  "route": {
    "meshName": "apps",
    "metadata": {
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBhttp2/route/http2Route",
      "createdAt": 1572011008.352,
      "lastUpdatedAt": 1572011008.352,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "routeName": "http2Route",
    "spec": {
      "http2Route": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "serviceBhttp2",
              "weight": 100
            }
          ]
        },
        "match": {
          "headers": [
            {
              "invert": false,
              "match": {
                "prefix": "123"
              },
              "name": "clientRequestId"
            }
          ],
          "method": "POST",
          "prefix": "/",
          "scheme": "http"
        },
        "retryPolicy": {
          "httpRetryEvents": [
            "server-error",

```

```

        "gateway-error"
      ],
      "maxRetries": 3,
      "perRetryTimeout": {
        "unit": "s",
        "value": 15
      },
      "tcpRetryEvents": [
        "connection-error"
      ]
    }
  },
  "priority": 200
},
"status": {
  "status": "ACTIVE"
},
"virtualRouterName": "serviceBhttp2"
}
}

```

Um eine neue TCP Route zu erstellen

Im folgenden `create-route` Beispiel wird eine JSON Eingabedatei verwendet, um eine TCP Route zu erstellen. 75 Prozent des Datenverkehrs werden an einen virtuellen Knoten mit dem Namen `weitergeleitet-serviceBtcp`, und 25 Prozent des Datenverkehrs werden an einen virtuellen Knoten namens `serviceBv 2tcp` weitergeleitet. Die Angabe unterschiedlicher Gewichtungen für verschiedene Ziele ist eine effektive Methode zur Bereitstellung einer neuen Version einer Anwendung. Sie können die Gewichtungen so anpassen, dass letztendlich 100 Prozent des gesamten Datenverkehrs an ein Ziel weitergeleitet werden, auf dem die neue Version einer Anwendung installiert ist.

```

aws appmesh create-route \
  --cli-input-json file://create-route-tcp.json

```

Inhalt `create-route-tcp von.json`:

```

{
  "meshName": "apps",
  "routeName": "tcpRoute",
  "spec": {

```

```

    "priority": 300,
    "tcpRoute": {
      "action": {
        "weightedTargets": [
          {
            "virtualNode": "serviceBtcp",
            "weight": 75
          },
          {
            "virtualNode": "serviceBv2tcp",
            "weight": 25
          }
        ]
      }
    }
  },
  "virtualRouterName": "serviceBtcp"
}

```

Ausgabe:

```

{
  "route": {
    "meshName": "apps",
    "metadata": {
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBtcp/route/tcpRoute",
      "createdAt": 1572011436.26,
      "lastUpdatedAt": 1572011436.26,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "routeName": "tcpRoute",
    "spec": {
      "priority": 300,
      "tcpRoute": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "serviceBtcp",
              "weight": 75
            },
            {

```

```

        "virtualNode": "serviceBv2tcp",
        "weight": 25
      }
    ]
  }
},
"status": {
  "status": "ACTIVE"
},
"virtualRouterName": "serviceBtcp"
}
}

```

Weitere Informationen finden Sie unter [Routes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateRoute](#) in der AWS CLI Befehlsreferenz.

create-virtual-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-virtual-gateway`.

AWS CLI

Um ein neues virtuelles Gateway zu erstellen

Das folgende `create-virtual-gateway` Beispiel verwendet eine JSON Eingabedatei, um ein virtuelles Gateway mit einem Listener für die HTTP Verwendung von Port 9080 zu erstellen.

```

aws appmesh create-virtual-gateway \
  --mesh-name meshName \
  --virtual-gateway-name virtualGatewayName \
  --cli-input-json file://create-virtual-gateway.json

```

Inhalt von `create-virtual-gateway.json`:

```

{
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 9080,

```



```
        "protocol": "http"
      }
    ]
  }
}
```

Ausgabe:

```
{
  "virtualGateway": {
    "meshName": "meshName",
    "metadata": {
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/meshName/
virtualGateway/virtualGatewayName",
      "createdAt": "2022-04-06T10:42:42.015000-05:00",
      "lastUpdatedAt": "2022-04-06T10:42:42.015000-05:00",
      "meshOwner": "123456789012",
      "resourceOwner": "123456789012",
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 9080,
            "protocol": "http"
          }
        }
      ]
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualGatewayName": "virtualGatewayName"
  }
}
```

Weitere Informationen finden Sie unter [Virtual Gateways](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVirtualGateway](#) in der AWS CLI Befehlsreferenz.

create-virtual-node

Das folgende Codebeispiel zeigt die Verwendung `create-virtual-node`.

AWS CLI

Beispiel 1: Um einen neuen virtuellen Knoten zu erstellen, der DNS für die Erkennung verwendet wird

Im folgenden `create-virtual-node` Beispiel wird eine JSON Eingabedatei verwendet, um einen virtuellen Knoten zu erstellen, der DNS für die Diensterkennung verwendet wird.

```
aws appmesh create-virtual-node \  
  --cli-input-json file://create-virtual-node-dns.json
```

Inhalt von `create-virtual-node-dns.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ],  
    "serviceDiscovery": {  
      "dns": {  
        "hostname": "serviceBv1.svc.cluster.local"  
      }  
    }  
  },  
  "virtualNodeName": "vnServiceBv1"  
}
```

Ausgabe:

```
{  
  "virtualNode": {  
    "meshName": "app1",
```

```

    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
      "createdAt": 1563810019.874,
      "lastUpdatedAt": 1563810019.874,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ],
      "serviceDiscovery": {
        "dns": {
          "hostname": "serviceBv1.svc.cluster.local"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualNodeName": "vnServiceBv1"
  }
}

```

Beispiel 2: So erstellen Sie einen neuen virtuellen Knoten, der AWS Cloud Map für die Erkennung verwendet

Das folgende `create-virtual-node` Beispiel verwendet eine JSON Eingabedatei, um einen virtuellen Knoten zu erstellen, der AWS Cloud Map für die Diensterkennung verwendet.

```

aws appmesh create-virtual-node \
  --cli-input-json file://create-virtual-node-cloud-map.json

```

Inhalt von `create-virtual-node-cloud-map.json`:

```
{
```

```

"meshName": "app1",
"spec": {
  "backends": [
    {
      "virtualService": {
        "virtualServiceName": "serviceA.svc.cluster.local"
      }
    }
  ],
  "listeners": [
    {
      "portMapping": {
        "port": 80,
        "protocol": "http"
      }
    }
  ],
  "serviceDiscovery": {
    "awsCloudMap": {
      "attributes": [
        {
          "key": "Environment",
          "value": "Testing"
        }
      ],
      "namespaceName": "namespace1",
      "serviceName": "serviceA"
    }
  }
},
"virtualNodeName": "vnServiceA"
}

```

Ausgabe:

```

{
  "virtualNode": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceA",
      "createdAt": 1563810859.465,
      "lastUpdatedAt": 1563810859.465,

```

```
    "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "version": 1
  },
  "spec": {
    "backends": [
      {
        "virtualService": {
          "virtualServiceName": "serviceA.svc.cluster.local"
        }
      }
    ],
    "listeners": [
      {
        "portMapping": {
          "port": 80,
          "protocol": "http"
        }
      }
    ],
    "serviceDiscovery": {
      "awsCloudMap": {
        "attributes": [
          {
            "key": "Environment",
            "value": "Testing"
          }
        ],
        "namespaceName": "namespace1",
        "serviceName": "serviceA"
      }
    }
  },
  "status": {
    "status": "ACTIVE"
  },
  "virtualNodeName": "vnServiceA"
}
}
```

Weitere Informationen finden Sie unter [Virtuelle Knoten](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVirtualNode](#) in der AWS CLI Befehlsreferenz.

create-virtual-router

Das folgende Codebeispiel zeigt die Verwendung `create-virtual-router`.

AWS CLI

Um einen neuen virtuellen Router zu erstellen

Das folgende `create-virtual-router` Beispiel verwendet eine JSON Eingabedatei, um einen virtuellen Router mit einem Listener für die HTTP Verwendung von Port 80 zu erstellen.

```
aws appmesh create-virtual-router \  
  --cli-input-json file://create-virtual-router.json
```

Inhalt von `create-virtual-router.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ]  
  },  
  "virtualRouterName": "vrServiceB"  
}
```

Ausgabe:

```
{  
  "virtualRouter": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB",  
      "createdAt": 1563810546.59,  
      "lastUpdatedAt": 1563810546.59,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    }  
  }  
}
```

```

    },
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ]
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualRouterName": "vrServiceB"
  }
}

```

Weitere Informationen finden Sie unter [Virtuelle Router](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVirtualRouter](#) in der AWS CLI Befehlsreferenz.

create-virtual-service

Das folgende Codebeispiel zeigt die Verwendung `create-virtual-service`.

AWS CLI

Beispiel 1: Um einen neuen virtuellen Dienst mit einem Anbieter für virtuelle Knoten zu erstellen

Im folgenden `create-virtual-service` Beispiel wird eine JSON Eingabedatei verwendet, um einen virtuellen Dienst mit einem Anbieter für virtuelle Knoten zu erstellen.

```

aws appmesh create-virtual-service \
  --cli-input-json file://create-virtual-service-virtual-node.json

```

Inhalt von `create-virtual-service-virtual-node.json`:

```

{
  "meshName": "app1",
  "spec": {
    "provider": {

```

```

        "virtualNode": {
            "virtualNodeName": "vnServiceA"
        }
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}

```

Ausgabe:

```

{
  "virtualService": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/serviceA.svc.cluster.local",
      "createdAt": 1563810859.474,
      "lastUpdatedAt": 1563810967.179,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 2
    },
    "spec": {
      "provider": {
        "virtualNode": {
          "virtualNodeName": "vnServiceA"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
  }
}

```

Weitere Informationen finden Sie unter [Virtual Node](#) im AWS App Mesh Mesh-Benutzerhandbuch.

Beispiel 2: Um einen neuen virtuellen Dienst mit einem virtuellen Router-Anbieter zu erstellen

Das folgende `create-virtual-service` Beispiel verwendet eine JSON Eingabedatei, um einen virtuellen Dienst mit einem virtuellen Router-Anbieter zu erstellen.

```
aws appmesh create-virtual-service \
```



```
--cli-input-json file://create-virtual-service-virtual-router.json
```

Inhalt von `create-virtual-service-virtual-router.json`:

```
{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualRouter": {
        "virtualRouterName": "vrServiceB"
      }
    }
  },
  "virtualServiceName": "serviceB.svc.cluster.local"
}
```

Ausgabe:

```
{
  "virtualService": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
      "createdAt": 1563908363.999,
      "lastUpdatedAt": 1563908363.999,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "provider": {
        "virtualRouter": {
          "virtualRouterName": "vrServiceB"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
  }
}
```

Weitere Informationen finden Sie unter [Virtual Services](https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html) < https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html > im App Mesh Mesh-Benutzerhandbuch AWS

- API-Einheiten finden Sie in der Befehlsreferenz. [CreateVirtualService](#) AWS CLI

delete-mesh

Das folgende Codebeispiel zeigt die Verwendung `delete-mesh`.

AWS CLI

Um ein Service Mesh zu löschen

Im folgenden `delete-mesh` Beispiel wird das angegebene Service Mesh gelöscht.

```
aws appmesh delete-mesh \  
  --mesh-name app1
```

Ausgabe:

```
{  
  "mesh": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",  
      "createdAt": 1563809909.282,  
      "lastUpdatedAt": 1563824981.248,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "spec": {  
      "egressFilter": {  
        "type": "ALLOW_ALL"  
      }  
    },  
    "status": {  
      "status": "DELETED"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Service Meshes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteMesh](#) in der AWS CLI Befehlsreferenz.

delete-route

Das folgende Codebeispiel zeigt die Verwendung `delete-route`.

AWS CLI

Um eine Route zu löschen

Im folgenden `delete-route` Beispiel wird die angegebene Route gelöscht.

```
aws appmesh delete-route \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB \  
  --route-name toVnServiceB-weighted
```

Ausgabe:

```
{  
  "route": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB/route/toVnServiceB-weighted",  
      "createdAt": 1563811384.015,  
      "lastUpdatedAt": 1563823915.936,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 3  
    },  
    "routeName": "toVnServiceB-weighted",  
    "spec": {  
      "httpRoute": {  
        "action": {  
          "weightedTargets": [  
            {  
              "virtualNode": "vnServiceBv1",  
              "weight": 80  
            },  
            {  
              "virtualNode": "vnServiceBv2",  
              "weight": 20  
            }  
          ]  
        }  
      }  
    }  
  }  
}
```

```

        ]
      },
      "match": {
        "prefix": "/"
      }
    }
  },
  "status": {
    "status": "DELETED"
  },
  "virtualRouterName": "vrServiceB"
}
}

```

Weitere Informationen finden Sie unter [Routes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteRoute](#) in der AWS CLI Befehlsreferenz.

delete-virtual-node

Das folgende Codebeispiel zeigt die Verwendung `delete-virtual-node`.

AWS CLI

Um einen virtuellen Knoten zu löschen

Im folgenden `delete-virtual-node` Beispiel wird der angegebene virtuelle Knoten gelöscht.

```

aws appmesh delete-virtual-node \
  --mesh-name app1 \
  --virtual-node-name vnServiceBv2

```

Ausgabe:

```

{
  "virtualNode": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceBv2",
      "createdAt": 1563810117.297,
      "lastUpdatedAt": 1563824700.678,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    }
  }
}

```

```

    "version": 2
  },
  "spec": {
    "backends": [],
    "listeners": [
      {
        "portMapping": {
          "port": 80,
          "protocol": "http"
        }
      }
    ],
    "serviceDiscovery": {
      "dns": {
        "hostname": "serviceBv2.svc.cluster.local"
      }
    }
  },
  "status": {
    "status": "DELETED"
  },
  "virtualNodeName": "vnServiceBv2"
}
}

```

Weitere Informationen finden Sie unter [Virtuelle Knoten](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVirtualNode](#) in der AWS CLI Befehlsreferenz.

delete-virtual-router

Das folgende Codebeispiel zeigt die Verwendung `delete-virtual-router`.

AWS CLI

Um einen virtuellen Router zu löschen

Im folgenden `delete-virtual-router` Beispiel wird der angegebene virtuelle Router gelöscht.

```

aws appmesh delete-virtual-router \
  --mesh-name app1 \
  --virtual-router-name vrServiceB

```

Ausgabe:

```
{
  "virtualRouter": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/vrServiceB",
      "createdAt": 1563810546.59,
      "lastUpdatedAt": 1563824253.467,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 3
    },
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ]
    },
    "status": {
      "status": "DELETED"
    },
    "virtualRouterName": "vrServiceB"
  }
}
```

Weitere Informationen finden Sie unter [Virtuelle Router](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVirtualRouter](#) in der AWS CLI Befehlsreferenz.

delete-virtual-service

Das folgende Codebeispiel zeigt die Verwendung `delete-virtual-service`.

AWS CLI

Um einen virtuellen Dienst zu löschen

Im folgenden `delete-virtual-service` Beispiel wird der angegebene virtuelle Dienst gelöscht.

```
aws appmesh delete-virtual-service \  
  --mesh-name app1 \  
  --virtual-service-name serviceB.svc.cluster.local
```

Ausgabe:

```
{  
  "virtualService": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/  
serviceB.svc.cluster.local",  
      "createdAt": 1563908363.999,  
      "lastUpdatedAt": 1563913940.866,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 3  
    },  
    "spec": {},  
    "status": {  
      "status": "DELETED"  
    },  
    "virtualServiceName": "serviceB.svc.cluster.local"  
  }  
}
```

Weitere Informationen finden Sie unter [Virtual Service](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVirtualService](#) in der AWS CLI Befehlsreferenz.

describe-mesh

Das folgende Codebeispiel zeigt die Verwendung `describe-mesh`.

AWS CLI

Um ein Service Mesh zu beschreiben

Das folgende `describe-mesh` Beispiel gibt Details zum angegebenen Service Mesh zurück.

```
aws appmesh describe-mesh \  
  --mesh-name app1
```

Ausgabe:

```
{  
  "mesh": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",  
      "createdAt": 1563809909.282,  
      "lastUpdatedAt": 1563809909.282,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "spec": {},  
    "status": {  
      "status": "ACTIVE"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Service Meshes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMesh](#) in der AWS CLI Befehlsreferenz.

describe-route

Das folgende Codebeispiel zeigt die Verwendung `describe-route`.

AWS CLI

Um eine Route zu beschreiben

Das folgende `describe-route` Beispiel gibt Details zur angegebenen Route zurück.

```
aws appmesh describe-route \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB \  
  --route-name toVnServiceB-weighted
```


Ausgabe:

```
{
  "route": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/vrServiceB/route/toVnServiceB-weighted",
      "createdAt": 1563811384.015,
      "lastUpdatedAt": 1563811384.015,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "routeName": "toVnServiceB-weighted",
    "spec": {
      "httpRoute": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "vnServiceBv1",
              "weight": 90
            },
            {
              "virtualNode": "vnServiceBv2",
              "weight": 10
            }
          ]
        },
        "match": {
          "prefix": "/"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualRouterName": "vrServiceB"
  }
}
```

Weitere Informationen finden Sie unter [Routes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeRoute](#) in der AWS CLI Befehlsreferenz.

describe-virtual-node

Das folgende Codebeispiel zeigt die Verwendung `describe-virtual-node`.

AWS CLI

Um einen virtuellen Knoten zu beschreiben

Das folgende `describe-virtual-node` Beispiel gibt Details über den angegebenen virtuellen Knoten zurück.

```
aws appmesh describe-virtual-node \  
  --mesh-name app1 \  
  --virtual-node-name vnServiceBv1
```

Ausgabe:

```
{  
  "virtualNode": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/  
vnServiceBv1",  
      "createdAt": 1563810019.874,  
      "lastUpdatedAt": 1563810019.874,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "spec": {  
      "backends": [],  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 80,  
            "protocol": "http"  
          }  
        }  
      ],  
      "serviceDiscovery": {  
        "dns": {  
          "hostname": "serviceBv1.svc.cluster.local"  
        }  
      }  
    }  
  }  
}
```

```

    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualNodeName": "vnServiceBv1"
  }
}

```

Weitere Informationen finden Sie unter [Virtuelle Knoten](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVirtualNode](#) in der AWS CLI Befehlsreferenz.

describe-virtual-router

Das folgende Codebeispiel zeigt die Verwendung `describe-virtual-router`.

AWS CLI

Um einen virtuellen Router zu beschreiben

Das folgende `describe-virtual-router` Beispiel gibt Details zum angegebenen virtuellen Router zurück.

```

aws appmesh describe-virtual-router \
  --mesh-name app1 \
  --virtual-router-name vrServiceB

```

Ausgabe:

```

{
  "virtualRouter": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/vrServiceB",
      "createdAt": 1563810546.59,
      "lastUpdatedAt": 1563810546.59,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {

```

```

    "listeners": [
      {
        "portMapping": {
          "port": 80,
          "protocol": "http"
        }
      }
    ],
    "status": {
      "status": "ACTIVE"
    },
    "virtualRouterName": "vrServiceB"
  }
}

```

Weitere Informationen finden Sie unter [Virtuelle Router](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVirtualRouter](#) in der AWS CLI Befehlsreferenz.

describe-virtual-service

Das folgende Codebeispiel zeigt die Verwendung `describe-virtual-service`.

AWS CLI

Um einen virtuellen Dienst zu beschreiben

Das folgende `describe-virtual-service` Beispiel gibt Details zum angegebenen virtuellen Dienst zurück.

```

aws appmesh describe-virtual-service \
  --mesh-name app1 \
  --virtual-service-name serviceB.svc.cluster.local

```

Ausgabe:

```

{
  "virtualService": {
    "meshName": "app1",
    "metadata": {

```

```

        "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
        "createdAt": 1563908363.999,
        "lastUpdatedAt": 1563908363.999,
        "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "version": 1
    },
    "spec": {
        "provider": {
            "virtualRouter": {
                "virtualRouterName": "vrServiceB"
            }
        }
    },
    "status": {
        "status": "ACTIVE"
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
}
}

```

Weitere Informationen finden Sie unter [Virtual Services](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVirtualService](#) in der AWS CLI Befehlsreferenz.

list-meshes

Das folgende Codebeispiel zeigt die Verwendung `list-meshes`.

AWS CLI

Um Service Meshes aufzulisten

Das folgende `list-meshes` Beispiel listet alle Service Meshes in der aktuellen AWS Region auf.

```
aws appmesh list-meshes
```

Ausgabe:

```
{
  "meshes": [
```

```
{
  "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
  "meshName": "app1"
}
]
```

Weitere Informationen finden Sie unter [Service Meshes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [ListMeshes](#) in der AWS CLI Befehlsreferenz.

list-routes

Das folgende Codebeispiel zeigt die Verwendung `list-routes`.

AWS CLI

Um Routen aufzulisten

Das folgende `list-routes` Beispiel listet alle Routen für den angegebenen virtuellen Router auf.

```
aws appmesh list-routes \
  --mesh-name app1 \
  --virtual-router-name vrServiceB
```

Ausgabe:

```
{
  "routes": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB",
      "meshName": "app1",
      "routeName": "toVnServiceB-weighted",
      "virtualRouterName": "vrServiceB"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Routes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRoutes](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags auf, die der angegebenen Ressource zugewiesen sind.

```
aws appmesh list-tags-for-resource \
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

Ausgabe:

```
{
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value2"
    },
    {
      "key": "key3",
      "value": "value3"
    }
  ]
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

list-virtual-nodes

Das folgende Codebeispiel zeigt die Verwendung `list-virtual-nodes`.

AWS CLI

Um virtuelle Knoten aufzulisten

Das folgende `list-virtual-nodes` Beispiel listet alle virtuellen Knoten im angegebenen Service Mesh auf.

```
aws appmesh list-virtual-nodes \  
  --mesh-name app1
```

Ausgabe:

```
{  
  "virtualNodes": [  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/  
vnServiceBv1",  
      "meshName": "app1",  
      "virtualNodeName": "vnServiceBv1"  
    },  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/  
vnServiceBv2",  
      "meshName": "app1",  
      "virtualNodeName": "vnServiceBv2"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Virtuelle Knoten](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [ListVirtualNodes](#) in der AWS CLI Befehlsreferenz.

list-virtual-routers

Das folgende Codebeispiel zeigt die Verwendung `list-virtual-routers`.

AWS CLI

Um virtuelle Router aufzulisten

Das folgende `list-virtual-routers` Beispiel listet alle virtuellen Router im angegebenen Service Mesh auf.

```
aws appmesh list-virtual-routers \  
  --mesh-name app1
```

Ausgabe:

```
{  
  "virtualRouters": [  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB",  
      "meshName": "app1",  
      "virtualRouterName": "vrServiceB"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Virtuelle Router](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [ListVirtualRouters](#) in der AWS CLI Befehlsreferenz.

list-virtual-services

Das folgende Codebeispiel zeigt die Verwendung `list-virtual-services`.

AWS CLI

Um virtuelle Dienste aufzulisten

Das folgende `list-virtual-services` Beispiel listet alle virtuellen Dienste im angegebenen Service Mesh auf.

```
aws appmesh list-virtual-services \  
  --mesh-name app1
```

Ausgabe:

```
{
```

```

"virtualServices": [
  {
    "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
    "meshName": "app1",
    "virtualServiceName": "serviceA.svc.cluster.local"
  },
  {
    "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
    "meshName": "app1",
    "virtualServiceName": "serviceB.svc.cluster.local"
  }
]
}

```

Weitere Informationen finden Sie unter [Virtual Services](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [ListVirtualServices](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird der angegebenen Ressource das Tag `key1` mit `value1` dem Wert hinzugefügt.

```

aws appmesh tag-resource \
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
  --tags key=key1,value=value1

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung einer Ressource aufzuheben

Im folgenden `untag-resource` Beispiel wird ein Tag mit dem Schlüssel `key1` aus der angegebenen Ressource entfernt.

```
aws appmesh untag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tag-keys key1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-mesh

Das folgende Codebeispiel zeigt die Verwendung `update-mesh`.

AWS CLI

Um ein Service Mesh zu aktualisieren

Im folgenden `update-mesh` Beispiel wird eine JSON Eingabedatei verwendet, um ein Service Mesh zu aktualisieren, sodass der gesamte externe Ausgangsverkehr unverändert über den Envoy-Proxy weitergeleitet werden kann.

```
aws appmesh update-mesh \  
  --cli-input-json file://update-mesh.json
```

Inhalt von `update-mesh.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "egressFilter": {  
      "type": "ALLOW_ALL"  
    }  
  }  
}
```

Ausgabe:

```
{
  "mesh": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
      "createdAt": 1563809909.282,
      "lastUpdatedAt": 1563812829.687,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 2
    },
    "spec": {
      "egressFilter": {
        "type": "ALLOW_ALL"
      }
    },
    "status": {
      "status": "ACTIVE"
    }
  }
}
```

Weitere Informationen finden Sie unter [Service Meshes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateMesh](#) in der AWS CLI Befehlsreferenz.

update-route

Das folgende Codebeispiel zeigt die Verwendung `update-route`.

AWS CLI

Um eine Route zu aktualisieren

Das folgende `update-route` Beispiel verwendet eine JSON Eingabedatei, um die Gewichtungen für eine Route zu aktualisieren.

```
aws appmesh update-route \
  --cli-input-json file://update-route-weighted.json
```

Inhalt von update-route-weighted.json:

```
{
  "meshName": "app1",
  "routeName": "toVnServiceB-weighted",
  "spec": {
    "httpRoute": {
      "action": {
        "weightedTargets": [
          {
            "virtualNode": "vnServiceBv1",
            "weight": 80
          },
          {
            "virtualNode": "vnServiceBv2",
            "weight": 20
          }
        ]
      },
      "match": {
        "prefix": "/"
      }
    }
  },
  "virtualRouterName": "vrServiceB"
}
```

Ausgabe:

```
{
  "route": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/vrServiceB/route/toVnServiceB-weighted",
      "createdAt": 1563811384.015,
      "lastUpdatedAt": 1563819600.022,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 2
    },
    "routeName": "toVnServiceB-weighted",
    "spec": {
      "httpRoute": {
```

```

    "action": {
      "weightedTargets": [
        {
          "virtualNode": "vnServiceBv1",
          "weight": 80
        },
        {
          "virtualNode": "vnServiceBv2",
          "weight": 20
        }
      ]
    },
    "match": {
      "prefix": "/"
    }
  }
},
"status": {
  "status": "ACTIVE"
},
"virtualRouterName": "vrServiceB"
}
}

```

Weitere Informationen finden Sie unter [Routes](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRoute](#) in der AWS CLI Befehlsreferenz.

update-virtual-node

Das folgende Codebeispiel zeigt die Verwendung `update-virtual-node`.

AWS CLI

Um einen virtuellen Knoten zu aktualisieren

Im folgenden `update-virtual-node` Beispiel wird eine JSON Eingabedatei verwendet, um einem virtuellen Knoten eine Integritätsprüfung hinzuzufügen.

```

aws appmesh update-virtual-node \
  --cli-input-json file://update-virtual-node.json

```

Inhalt von `update-virtual-node.json`:

```
{
  "clientToken": "500",
  "meshName": "app1",
  "spec": {
    "listeners": [
      {
        "healthCheck": {
          "healthyThreshold": 5,
          "intervalMillis": 10000,
          "path": "/",
          "port": 80,
          "protocol": "http",
          "timeoutMillis": 3000,
          "unhealthyThreshold": 3
        },
        "portMapping": {
          "port": 80,
          "protocol": "http"
        }
      }
    ],
    "serviceDiscovery": {
      "dns": {
        "hostname": "serviceBv1.svc.cluster.local"
      }
    }
  },
  "virtualNodeName": "vnServiceBv1"
}
```

Ausgabe:

```
{
  "virtualNode": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceBv1",
      "createdAt": 1563810019.874,
      "lastUpdatedAt": 1563819234.825,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 2
    }
  },
}
```

```
"spec": {
  "listeners": [
    {
      "healthCheck": {
        "healthyThreshold": 5,
        "intervalMillis": 10000,
        "path": "/",
        "port": 80,
        "protocol": "http",
        "timeoutMillis": 3000,
        "unhealthyThreshold": 3
      },
      "portMapping": {
        "port": 80,
        "protocol": "http"
      }
    }
  ],
  "serviceDiscovery": {
    "dns": {
      "hostname": "serviceBv1.svc.cluster.local"
    }
  }
},
"status": {
  "status": "ACTIVE"
},
"virtualNodeName": "vnServiceBv1"
}
```

Weitere Informationen finden Sie unter [Virtuelle Knoten](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateVirtualNode](#) in der AWS CLI Befehlsreferenz.

update-virtual-router

Das folgende Codebeispiel zeigt die Verwendung `update-virtual-router`.

AWS CLI

Um einen virtuellen Router zu aktualisieren

Im folgenden `update-virtual-router` Beispiel wird eine JSON Eingabedatei verwendet, um den Listener-Port eines virtuellen Routers zu aktualisieren.

```
aws appmesh update-virtual-router \  
  --cli-input-json file://update-virtual-router.json
```

Inhalt von `update-virtual-router.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 8080,  
          "protocol": "http"  
        }  
      }  
    ]  
  },  
  "virtualRouterName": "vrServiceB"  
}
```

Ausgabe:

```
{  
  "virtualRouter": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB",  
      "createdAt": 1563810546.59,  
      "lastUpdatedAt": 1563819431.352,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "spec": {  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 8080,  
            "protocol": "http"  
          }  
        }  
      ]  
    }  
  }  
}
```

```

        }
      }
    ]
  },
  "status": {
    "status": "ACTIVE"
  },
  "virtualRouterName": "vrServiceB"
}
}

```

Weitere Informationen finden Sie unter [Virtuelle Router](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateVirtualRouter](#) in der AWS CLI Befehlsreferenz.

update-virtual-service

Das folgende Codebeispiel zeigt die Verwendung `update-virtual-service`.

AWS CLI

Um einen virtuellen Dienst zu aktualisieren

Im folgenden `update-virtual-service` Beispiel wird eine JSON Eingabedatei verwendet, um einen virtuellen Dienst so zu aktualisieren, dass er einen virtuellen Router-Anbieter verwendet.

```
aws appmesh update-virtual-service \
  --cli-input-json file://update-virtual-service.json
```

Inhalt von `update-virtual-service.json`:

```

{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualRouter": {
        "virtualRouterName": "vrServiceA"
      }
    }
  },
  "virtualServiceName": "serviceA.svc.cluster.local"
}

```

```
}
```

Ausgabe:

```
{
  "virtualService": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceA.svc.cluster.local",
      "createdAt": 1563810859.474,
      "lastUpdatedAt": 1563820257.411,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 3
    },
    "spec": {
      "provider": {
        "virtualRouter": {
          "virtualRouterName": "vrServiceA"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
  }
}
```

Weitere Informationen finden Sie unter [Virtual Services](#) im AWS App Mesh Mesh-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateVirtualService](#) in der AWS CLI Befehlsreferenz.

App Runner-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von App Runner Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-custom-domain

Das folgende Codebeispiel zeigt die Verwendung `associate-custom-domain`.

AWS CLI

Um einen Domainnamen und die WWW-Subdomain einem Dienst zuzuordnen

Im folgenden `associate-custom-domain` Beispiel wird ein benutzerdefinierter Domainname, den Sie steuern, einem App Runner-Dienst zugeordnet. Der Domainname ist die Stammdomain `example.com`, einschließlich der Subdomain für Sonderfälle. `www.example.com`

```
aws apprunner associate-custom-domain \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "DomainName": "example.com",  
  "EnableWWWSubdomain": true  
}
```

Ausgabe:

```
{  
  "CustomDomain": {  
    "CertificateValidationRecords": [  
      {  
        "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
      }  
    ]  
  }  
}
```

```

        "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
      },
      {
        "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
        "Status": "PENDING_VALIDATION",
        "Type": "CNAME",
        "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
      }
    ],
    "DomainName": "example.com",
    "EnableWWWSubdomain": true,
    "Status": "CREATING"
  },
  "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

- API-Einheiten finden Sie [AssociateCustomDomain](#) in AWS CLI der Befehlsreferenz.

create-auto-scaling-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-auto-scaling-configuration`.

AWS CLI

So erstellen Sie eine Auto-Scaling-Konfiguration mit hoher Verfügbarkeit

Im folgenden `create-auto-scaling-configuration` Beispiel wird eine für Hochverfügbarkeit optimierte Auto-Scaling-Konfiguration erstellt, indem der Wert `MinSize` auf 5 gesetzt wird. Mit dieser Konfiguration versucht App Runner, Ihre Serviceinstanzen auf möglichst viele Availability Zones zu verteilen, je nach AWS Region bis zu fünf.

Der Aufruf gibt ein `AutoScalingConfiguration` Objekt zurück, bei dem die anderen Einstellungen auf die Standardwerte gesetzt sind. In diesem Beispiel ist dies der erste Aufruf zur Erstellung einer Konfiguration mit dem Namen `high-availability`. Die Revision ist auf 1 gesetzt und es ist die neueste Revision.

```

aws apprunner create-auto-scaling-configuration \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```
{
  "AutoScalingConfigurationName": "high-availability",
  "MinSize": 5
}
```

Ausgabe:

```
{
  "AutoScalingConfiguration": {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-east-1:123456789012:autoscalingconfiguration/high-availability/1/2f50e7656d7819fead0f59672e68042e",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 1,
    "CreatedAt": "2020-11-03T00:29:17Z",
    "Latest": true,
    "Status": "ACTIVE",
    "MaxConcurrency": 100,
    "MaxSize": 50,
    "MinSize": 5
  }
}
```

- API-Einheiten finden Sie [CreateAutoScalingConfiguration](#) in der AWS CLI Befehlsreferenz.

create-connection

Das folgende Codebeispiel zeigt die Verwendung `create-connection`.

AWS CLI

Um eine GitHub-Verbindung herzustellen

Das folgende `create-connection`-Beispiel stellt eine Verbindung zu einem privaten GitHub Code-Repository her. Der Verbindungsstatus nach einem erfolgreichen Aufruf lautet `PENDING_HANDSHAKE`. Dies liegt daran, dass ein Authentifizierungs-Handshake mit dem Anbieter immer noch nicht stattgefunden hat. Schließen Sie den Handshake mit der App Runner-Konsole ab.

```
aws apprunner create-connection \
```

```
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ConnectionName": "my-github-connection",
  "ProviderType": "GITHUB"
}
```

Ausgabe:

```
{
  "Connection": {
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-connection",
    "ConnectionName": "my-github-connection",
    "Status": "PENDING_HANDSHAKE",
    "CreatedAt": "2020-11-03T00:32:51Z",
    "ProviderType": "GITHUB"
  }
}
```

Weitere Informationen finden Sie unter [App Runner-Verbindungen verwalten](#) im AWS App Runner-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateConnection](#) in der AWS CLI Befehlsreferenz.

create-service

Das folgende Codebeispiel zeigt die Verwendung `create-service`.

AWS CLI

Beispiel 1: Um einen Quellcode-Repository-Service zu erstellen

Im folgenden `create-service` Beispiel wird ein App Runner-Dienst erstellt, der auf einem Python-Quellcode-Repository basiert.

```
aws apprunner create-service \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ServiceName": "python-app",
  "SourceConfiguration": {
    "AuthenticationConfiguration": {
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection/e7656250f67242d7819feade6800f59e"
    },
    "AutoDeploymentsEnabled": true,
    "CodeRepository": {
      "RepositoryUrl": "https://github.com/my-account/python-hello",
      "SourceCodeVersion": {
        "Type": "BRANCH",
        "Value": "main"
      },
    },
    "CodeConfiguration": {
      "ConfigurationSource": "API",
      "CodeConfigurationValues": {
        "Runtime": "PYTHON_3",
        "BuildCommand": "pip install -r requirements.txt",
        "StartCommand": "python server.py",
        "Port": "8080",
        "RuntimeEnvironmentVariables": [
          {
            "NAME": "Jane"
          }
        ]
      }
    }
  },
  "InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "3 GB"
  }
}
```

Ausgabe:

```
{
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-20T19:05:25Z",
  }
}
```



```
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
      },
      "AutoDeploymentsEnabled": true,
      "CodeRepository": {
        "CodeConfiguration": {
          "CodeConfigurationValues": {
            "BuildCommand": "pip install -r requirements.txt",
            "Port": "8080",
            "Runtime": "PYTHON_3",
            "RuntimeEnvironmentVariables": [
              {
                "NAME": "Jane"
              }
            ],
            "StartCommand": "python server.py"
          },
          "ConfigurationSource": "Api"
        },
        "RepositoryUrl": "https://github.com/my-account/python-hello",
        "SourceCodeVersion": {
          "Type": "BRANCH",
          "Value": "main"
        }
      }
    },
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}
```

Beispiel 2: So erstellen Sie einen Quellcode-Repository-Service

Im folgenden `create-service` Beispiel wird ein App Runner-Dienst erstellt, der auf einem Python-Quellcode-Repository basiert.

```
aws apprunner create-service \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceName": "python-app",  
  "SourceConfiguration": {  
    "AuthenticationConfiguration": {  
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/  
my-github-connection/e7656250f67242d7819feade6800f59e"  
    },  
    "AutoDeploymentsEnabled": true,  
    "CodeRepository": {  
      "RepositoryUrl": "https://github.com/my-account/python-hello",  
      "SourceCodeVersion": {  
        "Type": "BRANCH",  
        "Value": "main"  
      },  
    },  
    "CodeConfiguration": {  
      "ConfigurationSource": "API",  
      "CodeConfigurationValues": {  
        "Runtime": "PYTHON_3",  
        "BuildCommand": "pip install -r requirements.txt",  
        "StartCommand": "python server.py",  
        "Port": "8080",  
        "RuntimeEnvironmentVariables": [  
          {  
            "NAME": "Jane"  
          }  
        ]  
      }  
    }  
  },  
  "InstanceConfiguration": {  
    "CPU": "1 vCPU",  
    "Memory": "3 GB"  
  }  
}
```

```
}

```

Ausgabe:

```
{
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-20T19:05:25Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
      },
      "AutoDeploymentsEnabled": true,
      "CodeRepository": {
        "CodeConfiguration": {
          "CodeConfigurationValues": {
            "BuildCommand": "pip install -r requirements.txt",
            "Port": "8080",
            "Runtime": "PYTHON_3",
            "RuntimeEnvironmentVariables": [
              {
                "NAME": "Jane"
              }
            ]
          },
          "StartCommand": "python server.py"
        },
        "ConfigurationSource": "Api"
      },
      "RepositoryUrl": "https://github.com/my-account/python-hello",
      "SourceCodeVersion": {
        "Type": "BRANCH",
        "Value": "main"
      }
    }
  },
}
```

```
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}
```

Beispiel 3: So erstellen Sie einen Quell-Image-Repository-Service

Im folgenden `create-service` Beispiel wird ein App Runner-Dienst auf der Grundlage eines in Elastic Container Registry (ECR) gespeicherten Images erstellt.

```
aws apprunner create-service \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ServiceName": "golang-container-app",
  "SourceConfiguration": {
    "AuthenticationConfiguration": {
      "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
    },
    "AutoDeploymentsEnabled": true,
    "ImageRepository": {
      "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/golang-
app:latest",
      "ImageConfiguration": {
        "Port": "8080",
        "RuntimeEnvironmentVariables": [
          {
            "NAME": "Jane"
          }
        ]
      },
      "ImageRepositoryType": "ECR"
    }
  },
  "InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "3 GB"
  }
}
```

```
}
}
```

Ausgabe:

```
{
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
  "Service": {
    "CreatedAt": "2020-11-06T23:15:30Z",
    "UpdatedAt": "2020-11-06T23:15:30Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/51728f8a20ce46d39b25398a6c8e9d1a",
    "ServiceId": "51728f8a20ce46d39b25398a6c8e9d1a",
    "ServiceName": "golang-container-app",
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
      },
      "AutoDeploymentsEnabled": true,
      "ImageRepository": {
        "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/
golang-app:latest",
        "ImageConfiguration": {
          "Port": "8080",
          "RuntimeEnvironmentVariables": [
            {
              "NAME": "Jane"
            }
          ]
        },
        "ImageRepositoryType": "ECR"
      }
    },
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}
```

- API-Einheiten finden Sie [CreateService](#) unter AWS CLI Befehlsreferenz.

delete-auto-scaling-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-auto-scaling-configuration`.

AWS CLI

Beispiel 1: Um die letzte aktive Revision einer Auto Scaling-Konfiguration zu löschen

Im folgenden `delete-auto-scaling-configuration` Beispiel wird die letzte aktive Version einer App Runner Auto Scaling-Konfiguration gelöscht. Um die letzte aktive Revision zu löschen, geben Sie einen Amazon-Ressourcennamen (ARN) an, der mit dem Konfigurationsnamen endet, ohne die Revisionskomponente.

In diesem Beispiel sind vor dieser Aktion zwei Revisionen vorhanden. Daher wird Revision 2 (die neueste) gelöscht. Es wird jetzt jedoch angezeigt `"Latest": false`, da es sich nach dem Löschen nicht mehr um die letzte aktive Revision handelt.

```
aws apprunner delete-auto-scaling-configuration \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability"  
}
```

Ausgabe:

```
{  
  "AutoScalingConfiguration": {  
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability/2/  
e76562f50d78042e819fead0f59672e6",  
    "AutoScalingConfigurationName": "high-availability",  
    "AutoScalingConfigurationRevision": 2,  
    "CreatedAt": "2021-02-25T17:42:59Z",  
    "DeletedAt": "2021-03-02T08:07:06Z",  
    "Latest": false,  
    "Status": "INACTIVE",  
    "MaxConcurrency": 30,  
    "MaxSize": 90,  
  }  
}
```

```
    "MinSize": 5
  }
}
```

Beispiel 2: Um eine bestimmte Revision einer Auto Scaling-Konfiguration zu löschen

Im folgenden `delete-auto-scaling-configuration` Beispiel wird eine bestimmte Version einer App Runner Auto Scaling-Konfiguration gelöscht. Um eine bestimmte Revision zu löschen, geben Sie eine an, ARN die die Revisionsnummer enthält.

In diesem Beispiel gab es vor dieser Aktion mehrere Revisionen. Die Aktion löscht die Revision. 1

```
aws apprunner delete-auto-scaling-configuration \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Ausgabe:

```
{
  "AutoScalingConfiguration": {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 1,
    "CreatedAt": "2020-11-03T00:29:17Z",
    "DeletedAt": "2021-03-02T08:07:06Z",
    "Latest": false,
    "Status": "INACTIVE",
    "MaxConcurrency": 100,
    "MaxSize": 50,
    "MinSize": 5
  }
}
```

- API-Einheiten finden Sie [DeleteAutoScalingConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-connection`.

AWS CLI

Um eine Verbindung zu löschen

Im folgenden `delete-connection` Beispiel wird eine App Runner-Verbindung gelöscht. Der Verbindungsstatus nach einem erfolgreichen Anruf lautet `DELETED`. Dies liegt daran, dass die Verbindung nicht mehr verfügbar ist.

```
aws apprunner delete-connection \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-  
connection"  
}
```

Ausgabe:

```
{  
  "Connection": {  
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-  
github-connection",  
    "ConnectionName": "my-github-connection",  
    "Status": "DELETED",  
    "CreatedAt": "2020-11-03T00:32:51Z",  
    "ProviderType": "GITHUB"  
  }  
}
```

- API-Einheiten finden Sie [DeleteConnection](#) in der AWS CLI Befehlsreferenz.

delete-service

Das folgende Codebeispiel zeigt die Verwendung `delete-service`.

AWS CLI

Um einen Dienst zu löschen

Im folgenden `delete-service` Beispiel wird ein App Runner-Dienst gelöscht.

```
aws apprunner delete-service \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

Ausgabe:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-20T19:05:25Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceName": "python-app",  
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",  
    "SourceConfiguration": {  
      "AuthenticationConfiguration": {  
        "ConnectionArn": "arn:aws:apprunner:us-  
east-1:123456789012:connection/my-github-connection/  
e7656250f67242d7819feade6800f59e"  
      },  
      "AutoDeploymentsEnabled": true,  
      "CodeRepository": {  
        "CodeConfiguration": {  
          "CodeConfigurationValues": {  
            "BuildCommand": "pip install -r requirements.txt",  
            "Port": "8080",  
            "Runtime": "PYTHON_3",  
            "RuntimeEnvironmentVariables": [  

```

```
        {
            "NAME": "Jane"
        }
    ],
    "StartCommand": "python server.py"
},
"ConfigurationSource": "Api"
},
"RepositoryUrl": "https://github.com/my-account/python-hello",
"SourceCodeVersion": {
    "Type": "BRANCH",
    "Value": "main"
}
}
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "3 GB"
}
}
}
```

- API-Einheiten finden Sie [DeleteService](#) in der AWS CLI Befehlsreferenz.

describe-auto-scaling-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-auto-scaling-configuration`.

AWS CLI

Beispiel 1: Um die letzte aktive Revision einer Auto Scaling-Konfiguration zu beschreiben

Im folgenden `describe-auto-scaling-configuration` Beispiel wird die letzte aktive Version einer App Runner Auto Scaling-Konfiguration beschrieben. Um die letzte aktive Revision zu beschreiben, geben Sie eine ARN, die mit dem Konfigurationsnamen endet, ohne die Revisionskomponente.

In dem Beispiel sind zwei Revisionen vorhanden. Daher wird die Revision 2 (die neueste) beschrieben. Das resultierende Objekt wird angezeigt `"Latest": true`.

```
aws apprunner describe-auto-scaling-configuration \
```

```
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

Ausgabe:

```
{
  "AutoScalingConfiguration": {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 2,
    "CreatedAt": "2021-02-25T17:42:59Z",
    "Latest": true,
    "Status": "ACTIVE",
    "MaxConcurrency": 30,
    "MaxSize": 90,
    "MinSize": 5
  }
}
```

Beispiel 2: Um eine bestimmte Version einer Auto Scaling-Konfiguration zu beschreiben

Das folgende `describe-auto-scaling-configuration` Beispiel enthält eine Beschreibung einer bestimmten Version einer App Runner Auto Scaling-Konfiguration. Um eine bestimmte Revision zu beschreiben, geben Sie eine anARN, die die Revisionsnummer enthält.

In dem Beispiel sind mehrere Revisionen vorhanden und die Revision 1 wird abgefragt. Das resultierende Objekt wird angezeigt. `"Latest": false`

```
aws apprunner describe-auto-scaling-configuration \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Ausgabe:

```
{
  "AutoScalingConfiguration": {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 1,
    "CreatedAt": "2020-11-03T00:29:17Z",
    "Latest": false,
    "Status": "ACTIVE",
    "MaxConcurrency": 100,
    "MaxSize": 50,
    "MinSize": 5
  }
}
```

- API-Einheiten finden Sie [DescribeAutoScalingConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-custom-domains

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-domains`.

AWS CLI

Um Beschreibungen von benutzerdefinierten Domainnamen abzurufen, die mit einem Dienst verknüpft sind

Im folgenden `describe-custom-domains` Beispiel werden Beschreibungen und Status der benutzerdefinierten Domainnamen abgerufen, die einem App Runner-Dienst zugeordnet sind.

```
aws apprunner describe-custom-domains \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "DomainName": "example.com",
  "EnableWWWSubdomain": true
}
```

Ausgabe:

```
{
  "CustomDomains": [
    {
      "CertificateValidationRecords": [
        {
          "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
          "Status": "PENDING_VALIDATION",
          "Type": "CNAME",
          "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
        },
        {
          "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
          "Status": "PENDING_VALIDATION",
          "Type": "CNAME",
          "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
        }
      ],
      "DomainName": "example.com",
      "EnableWWWSubdomain": true,
      "Status": "PENDING_CERTIFICATE_DNS_VALIDATION"
    },
    {
      "CertificateValidationRecords": [
        {
          "Name": "_a94f784c70d3f507c72dc28f55db2f6b.deals.example.com",
          "Status": "SUCCESS",
          "Type": "CNAME",
          "Value": "_2db02504c1270c137383c6307b6834b0.bsgbmzkfwj.acm-
validations.aws."
        }
      ],
      "DomainName": "deals.example.com",

```

```

        "EnableWWWSubdomain": false,
        "Status": "ACTIVE"
    }
],
"DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
"ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

- API-Einheiten finden Sie [DescribeCustomDomains](#) unter AWS CLI Befehlsreferenz.

describe-service

Das folgende Codebeispiel zeigt die Verwendung `describe-service`.

AWS CLI

Um einen Dienst zu beschreiben

Im folgenden `describe-service` Beispiel wird eine Beschreibung eines App Runner-Dienstes abgerufen.

```
aws apprunner describe-service \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Ausgabe:

```
{
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-20T19:05:25Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",

```

```

    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
      },
      "AutoDeploymentsEnabled": true,
      "CodeRepository": {
        "CodeConfiguration": {
          "CodeConfigurationValues": {
            "BuildCommand": "pip install -r requirements.txt",
            "Port": "8080",
            "Runtime": "PYTHON_3",
            "RuntimeEnvironmentVariables": [
              {
                "NAME": "Jane"
              }
            ],
            "StartCommand": "python server.py"
          },
          "ConfigurationSource": "Api"
        },
        "RepositoryUrl": "https://github.com/my-account/python-hello",
        "SourceCodeVersion": {
          "Type": "BRANCH",
          "Value": "main"
        }
      }
    },
    "Status": "RUNNING",
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}

```

- API-Einheiten finden Sie [DescribeService](#) in der AWS CLI Befehlsreferenz.

disassociate-custom-domain

Das folgende Codebeispiel zeigt die Verwendung `disassociate-custom-domain`.

AWS CLI

Um einen Domainnamen von einem Dienst zu trennen

Im folgenden `disassociate-custom-domain` Beispiel wird die Domain `example.com` von einem App Runner-Dienst getrennt. Durch den Aufruf wird auch die Subdomain `getrenntwww.example.com`, die zusammen mit der Stammdomain verknüpft war.

```
aws apprunner disassociate-custom-domain \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "DomainName": "example.com"  
}
```

Ausgabe:

```
{  
  "CustomDomain": {  
    "CertificateValidationRecords": [  
      {  
        "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
        "Value": "_1270c137383c6307b6832db02504c4b0.bsghbmzkfwj.acm-  
validations.aws."  
      },  
      {  
        "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
        "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-  
validations.aws."  
      }  
    ],  
    "DomainName": "example.com",  
    "EnableWWWSubdomain": true,  
  }  
}
```



```
    "Status": "DELETING"
  },
  "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- API-Einheiten finden Sie [DisassociateCustomDomain](#) in der AWS CLI Befehlsreferenz.

list-auto-scaling-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-auto-scaling-configurations`.

AWS CLI

Um eine paginierte Liste der Auto Scaling-Konfigurationen von App Runner zu erhalten

Das folgende `list-auto-scaling-configurations` Beispiel listet alle App Runner Auto Scaling-Konfigurationen in Ihrem AWS Konto auf. In jeder Antwort sind bis zu fünf Auto Scaling-Konfigurationen aufgeführt. `AutoScalingConfigurationName` und `LatestOnly` sind nicht spezifiziert. Ihre Standardeinstellungen führen dazu, dass die neueste Version aller aktiven Konfigurationen aufgelistet wird.

In diesem Beispiel enthält die Antwort zwei Ergebnisse und es gibt keine weiteren, sodass kein Ergebnis zurückgegeben `NextToken` wird.

```
aws apprunner list-auto-scaling-configurations \
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "MaxResults": 5
}
```

Ausgabe:

```
{
  "AutoScalingConfigurationSummaryList": [
```

```

    {
      "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
      "AutoScalingConfigurationName": "high-availability",
      "AutoScalingConfigurationRevision": 2
    },
    {
      "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/low-
cost/1/50d7804e7656fead0f59672e62f2e819",
      "AutoScalingConfigurationName": "low-cost",
      "AutoScalingConfigurationRevision": 1
    }
  ]
}

```

- API-Einheiten finden Sie [ListAutoScalingConfigurations](#) unter AWS CLI Befehlsreferenz.

list-connections

Das folgende Codebeispiel zeigt die Verwendung `list-connections`.

AWS CLI

Beispiel 1: Um alle Verbindungen aufzulisten

Das folgende `list-connections` Beispiel listet alle App Runner-Verbindungen im AWS Konto auf.

```
aws apprunner list-connections
```

Ausgabe:

```

{
  "ConnectionSummaryList": [
    {
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection",
      "ConnectionName": "my-github-connection",
      "Status": "AVAILABLE",

```

```

        "CreatedAt": "2020-11-03T00:32:51Z",
        "ProviderType": "GITHUB"
    },
    {
        "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
        "ConnectionName": "my-github-org-connection",
        "Status": "AVAILABLE",
        "CreatedAt": "2020-11-03T02:54:17Z",
        "ProviderType": "GITHUB"
    }
]
}

```

Beispiel 2: Um eine Verbindung nach Namen aufzulisten

Das folgende `list-connections` Beispiel listet eine Verbindung nach ihrem Namen auf.

```

aws apprunner list-connections \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "ConnectionName": "my-github-org-connection"
}

```

Ausgabe:

```

{
  "ConnectionSummaryList": [
    {
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
      "ConnectionName": "my-github-org-connection",
      "Status": "AVAILABLE",
      "CreatedAt": "2020-11-03T02:54:17Z",
      "ProviderType": "GITHUB"
    }
  ]
}

```

- API-Einheiten finden Sie [ListConnections](#) unter AWS CLI Befehlsreferenz.

list-operations

Das folgende Codebeispiel zeigt die Verwendung `list-operations`.

AWS CLI

Um Operationen aufzulisten, die bei einem Dienst aufgetreten sind

Das folgende `list-operations` Beispiel listet alle Operationen auf, die bisher in einem App Runner-Dienst aufgetreten sind. In diesem Beispiel ist der Dienst neu und es `CREATE_SERVICE` wurde nur ein einziger Vorgang des Typs ausgeführt.

```
aws apprunner list-operations \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

Ausgabe:

```
{  
  "OperationSummaryList": [  
    {  
      "EndedAt": 1606156217,  
      "Id": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
      "StartedAt": 1606156014,  
      "Status": "SUCCEEDED",  
      "TargetArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
      "Type": "CREATE_SERVICE",  
      "UpdatedAt": 1606156217  
    }  
  ]  
}
```



```

        "CreatedAt": "2020-11-06T23:15:30Z",
        "UpdatedAt": "2020-11-23T13:21:22Z",
        "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/ab8f94cfe29a460fb8760afd2ee87555",
        "ServiceId": "ab8f94cfe29a460fb8760afd2ee87555",
        "ServiceName": "golang-container-app",
        "ServiceUrl": "e2m8rrrx33.us-east-1.awsapprunner.com",
        "Status": "RUNNING"
    }
]
}

```

- API-Einheiten finden Sie [ListServices](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags aufzulisten, die einem App Runner-Dienst zugeordnet sind

Das folgende `list-tags-for-resource` Beispiel listet alle Tags auf, die einem App Runner-Dienst zugeordnet sind.

```

aws apprunner list-tags-for-resource \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

Ausgabe:

```

{
  "Tags": [
    {
      "Key": "Department",

```

```

        "Value": "Retail"
      },
      {
        "Key": "CustomerId",
        "Value": "56439872357912"
      }
    ]
  }
}

```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

pause-service

Das folgende Codebeispiel zeigt die Verwendung `pause-service`.

AWS CLI

Um einen Dienst anzuhalten

Im folgenden `pause-service` Beispiel wird ein App Runner-Dienst angehalten.

```

aws apprunner pause-service \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

Ausgabe:

```

{
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-23T12:41:37Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
  }
}

```

```

    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
      },
      "AutoDeploymentsEnabled": true,
      "CodeRepository": {
        "CodeConfiguration": {
          "CodeConfigurationValues": {
            "BuildCommand": "pip install -r requirements.txt",
            "Port": "8080",
            "Runtime": "PYTHON_3",
            "RuntimeEnvironmentVariables": [
              {
                "NAME": "Jane"
              }
            ],
            "StartCommand": "python server.py"
          },
          "ConfigurationSource": "Api"
        },
        "RepositoryUrl": "https://github.com/my-account/python-hello",
        "SourceCodeVersion": {
          "Type": "BRANCH",
          "Value": "main"
        }
      }
    },
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}

```

- API-Einheiten finden Sie [PauseService](#) in der AWS CLI Befehlsreferenz.

resume-service

Das folgende Codebeispiel zeigt die Verwendung `resume-service`.

AWS CLI

Um einen Dienst wieder aufzunehmen

Im folgenden `resume-service` Beispiel wird ein App Runner-Dienst wieder aufgenommen.

```
aws apprunner resume-service \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

Ausgabe:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-23T12:41:37Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceName": "python-app",  
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",  
    "SourceConfiguration": {  
      "AuthenticationConfiguration": {  
        "ConnectionArn": "arn:aws:apprunner:us-  
east-1:123456789012:connection/my-github-connection/  
e7656250f67242d7819feade6800f59e"  
      },  
      "AutoDeploymentsEnabled": true,  
      "CodeRepository": {  
        "CodeConfiguration": {  
          "CodeConfigurationValues": {  
            "BuildCommand": "pip install -r requirements.txt",  
            "Port": "8080",  
            "Runtime": "PYTHON_3",  
            "RuntimeEnvironmentVariables": [  

```

```
        {
            "NAME": "Jane"
        }
    ],
    "StartCommand": "python server.py"
},
"ConfigurationSource": "Api"
},
"RepositoryUrl": "https://github.com/my-account/python-hello",
"SourceCodeVersion": {
    "Type": "BRANCH",
    "Value": "main"
}
}
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "3 GB"
}
}
}
```

- API-Einheiten finden Sie [ResumeService](#) in der AWS CLI Befehlsreferenz.

start-deployment

Das folgende Codebeispiel zeigt die Verwendung `start-deployment`.

AWS CLI

Um eine manuelle Bereitstellung zu initiieren

Im folgenden `start-deployment` Beispiel wird eine manuelle Bereitstellung für einen App Runner-Dienst durchgeführt.

```
aws apprunner start-deployment \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
```

```
"ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Ausgabe:

```
{
  "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"
}
```

- API-Einheiten finden Sie [StartDeployment](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einem App Runner-Dienst Tags hinzuzufügen

Das folgende `tag-resource` Beispiel fügt einem App Runner-Dienst zwei Tags hinzu.

```
aws apprunner tag-resource \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "Tags": [
    {
      "Key": "Department",
      "Value": "Retail"
    },
    {
      "Key": "CustomerId",
      "Value": "56439872357912"
    }
  ]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einem App Runner-Dienst zu entfernen

Im folgenden `untag-resource` Beispiel werden zwei Tags aus einem App Runner-Dienst entfernt.

```
aws apprunner untag-resource \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "TagKeys": [  
    "Department",  
    "CustomerId"  
  ]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-service

Das folgende Codebeispiel zeigt die Verwendung `update-service`.

AWS CLI

Um die Speichergröße zu aktualisieren

Im folgenden `update-service` Beispiel wird die Speichergröße von Instanzen (Skalierungseinheiten) eines App Runner-Dienstes auf 2048 MiB aktualisiert.

Wenn der Aufruf erfolgreich ist, startet App Runner einen asynchronen Aktualisierungsprozess. Die Service Struktur, die durch den Aufruf zurückgegeben wird, spiegelt den neuen Speicherwert wider, der durch diesen Aufruf angewendet wird.

```
aws apprunner update-service \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "InstanceConfiguration": {  
    "Memory": "4 GB"  
  }  
}
```

Ausgabe:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-23T12:41:37Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceName": "python-app",  
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",  
    "SourceConfiguration": {  
      "AuthenticationConfiguration": {  
        "ConnectionArn": "arn:aws:apprunner:us-  
east-1:123456789012:connection/my-github-connection/  
e7656250f67242d7819feade6800f59e"  
      },  
      "AutoDeploymentsEnabled": true,  
      "CodeRepository": {  
        "CodeConfiguration": {  
          "CodeConfigurationValues": {  
            "BuildCommand": "pip install -r requirements.txt",  
            "Port": "8080",  
            "Runtime": "PYTHON_3",  

```

```
        "RuntimeEnvironmentVariables": [
            {
                "NAME": "Jane"
            }
        ],
        "StartCommand": "python server.py"
    },
    "ConfigurationSource": "Api"
},
"RepositoryUrl": "https://github.com/my-account/python-hello",
"SourceCodeVersion": {
    "Type": "BRANCH",
    "Value": "main"
}
}
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "4 GB"
}
}
}
```

- API-Einheiten finden Sie [UpdateService](#) unter AWS CLI Befehlsreferenz.

AWS AppConfig Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS AppConfig.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-application

Das folgende Codebeispiel zeigt die Verwendung `create-application`.

AWS CLI

Um eine Anwendung zu erstellen

Das folgende `create-application` Beispiel erstellt eine Anwendung in AWS AppConfig.

```
aws appconfig create-application \  
  --name "example-application" \  
  --description "An application used for creating an example."
```

Ausgabe:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [CreateApplication](#) in der AWS CLI Befehlsreferenz.

create-configuration-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `create-configuration-profile`.

AWS CLI

Um ein Konfigurationsprofil zu erstellen

Im folgenden `create-configuration-profile` Beispiel wird ein Konfigurationsprofil mithilfe einer Konfiguration erstellt, die in Parameter Store, einer Funktion von Systems Manager, gespeichert ist.

```
aws appconfig create-configuration-profile \  
  --application-id "339ohji" \  
  --description "An application used for creating an example."
```

```
--name "Example-Configuration-Profile" \  
--location-uri "ssm-parameter://Example-Parameter" \  
--retrieval-role-arn "arn:aws:iam::111122223333:role/Example-App-Config-Role"
```

Ausgabe:

```
{  
  "ApplicationId": "339ohji",  
  "Description": null,  
  "Id": "ur8hx2f",  
  "LocationUri": "ssm-parameter://Example-Parameter",  
  "Name": "Example-Configuration-Profile",  
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role",  
  "Type": null,  
  "Validators": null  
}
```

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [CreateConfigurationProfile](#) unter AWS CLI Befehlsreferenz.

create-environment

Das folgende Codebeispiel zeigt, wie man es benutzt `create-environment`.

AWS CLI

Um eine Umgebung zu erstellen

Im folgenden `create-environment` Beispiel wird mithilfe der Anwendung, die Sie mit `create-application` erstellt haben, eine AWS AppConfig Umgebung mit dem Namen `Example-Environment` erstellt.

```
aws appconfig create-environment \  
  --application-id "339ohji" \  
  --name "Example-Environment"
```

Ausgabe:

```
{  
  "ApplicationId": "339ohji",
```



```
"Description": null,
"Id": "54j1r29",
"Monitors": null,
"Name": "Example-Environment",
"State": "ReadyForDeployment"
}
```

Weitere Informationen finden Sie unter [Schritt 2: Erstellen einer Umgebung](#) im Benutzerhandbuch.AWS AppConfig

- APIEinzelheiten finden Sie [CreateEnvironment](#) in der AWS CLI Befehlsreferenz.

create-extension-association

Das folgende Codebeispiel zeigt, wie man es benutzt `create-extension-association`.

AWS CLI

Um eine Erweiterungszuordnung zu erstellen

Im folgenden `create-extension-association` Beispiel wird eine neue Erweiterungsassoziatio in erstellt AWS AppConfig.

```
aws appconfig create-extension-association \
  --region us-west-2 \
  --extension-identifier S3-backup-extension \
  --resource-identifier "arn:aws:appconfig:us-west-2:123456789012:application/Finance" \
  --parameters S3bucket=FinanceConfigurationBackup
```

Ausgabe:

```
{
  "Id": "a1b2c3d4",
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-extension/1",
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
  "Parameters": {
    "S3bucket": "FinanceConfigurationBackup"
  },
  "ExtensionVersionNumber": 1
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [CreateExtensionAssociation](#) unter AWS CLI Befehlsreferenz.

create-extension

Das folgende Codebeispiel zeigt, wie man es benutzt `create-extension`.

AWS CLI

Um eine Erweiterung zu erstellen

Das folgende `create-extension` Beispiel erstellt eine neue Erweiterung in AWS AppConfig.

```
aws appconfig create-extension \
  --region us-west-2 \
  --name S3-backup-extension \
  --
  actions PRE_CREATE_HOSTED_CONFIGURATION_VERSION=[{Name=S3backup,Uri=arn:aws:lambda:us-west-2:123456789012:function:s3backupfunction,RoleArn=arn:aws:iam::123456789012:role/appconfigextensionrole}] \
  --parameters S3bucket={Required=true}
```

Ausgabe:

```
{
  "Id": "1A2B3C4D",
  "Name": "S3-backup-extension",
  "VersionNumber": 1,
  "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",
  "Actions": {
    "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
      {
        "Name": "S3backup",
        "Uri": "arn:aws:lambda:us-west-2:123456789012:function:s3backupfunction",
        "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
      }
    ]
  },
  "Parameters": {
    "S3bucket": {
```

```

        "Required": true
      }
    }
  }
}

```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [CreateExtension](#) unter AWS CLI Befehlsreferenz.

create-hosted-configuration-version

Das folgende Codebeispiel zeigt, wie man es benutzt `create-hosted-configuration-version`.

AWS CLI

Um eine gehostete Konfigurationsversion zu erstellen

Im folgenden `create-hosted-configuration-version` Beispiel wird eine neue Konfiguration im AWS AppConfig gehosteten Konfigurationsspeicher erstellt. Der Konfigurationsinhalt muss zuerst in Base64 konvertiert werden.

```

aws appconfig create-hosted-configuration-version \
  --application-id "339ohji" \
  --configuration-profile-id "ur8hx2f" \
  --
content eyAiTmFtZSI6ICJFeGFtcGx1QXBwbGljYXRpb24iLCAiSWQ1OiBFcGFtcGx1SUQsICJSYW5rIjogNyB9
\
  --content-type "application/json" \
  configuration_version_output_file

```

Inhalt von `configuration_version_output_file`:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Ausgabe:

```
{
  "ApplicationId": "339ohji",
  "ConfigurationProfileId": "ur8hx2f",
  "VersionNumber": "1",
}
```

```
"ContentType": "application/json"  
}
```

Weitere Informationen finden Sie unter [Über den AWS AppConfig gehosteten Konfigurationsspeicher](#) im AWS AppConfig-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateHostedConfigurationVersion AWS CLI Befehlsreferenz](#).

delete-application

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-application`.

AWS CLI

So löschen Sie eine Anwendung

Im folgenden `delete-application` Beispiel wird die angegebene Anwendung gelöscht.

```
aws AppConfig delete-application \  
--application-id 339ohji
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteApplication](#) in der AWS CLI Befehlsreferenz.

delete-configuration-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-configuration-profile`.

AWS CLI

Um ein Konfigurationsprofil zu löschen

Im folgenden `delete-configuration-profile` Beispiel wird das angegebene Konfigurationsprofil gelöscht.

```
aws AppConfig delete-configuration-profile \  
--application-id 339ohji \  
--configuration-profile-id ur8hx2f
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteConfigurationProfile](#) unter AWS CLI Befehlsreferenz.

delete-deployment-strategy

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-deployment-strategy`.

AWS CLI

Um eine Bereitstellungsstrategie zu löschen

Im folgenden `delete-deployment-strategy` Beispiel wird die angegebene Bereitstellungsstrategie gelöscht.

```
aws appconfig delete-deployment-strategy \  
  --deployment-strategy-id 1225qzk
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 4: Erstellen einer Bereitstellungsstrategie](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDeploymentStrategy](#) in der AWS CLI Befehlsreferenz.

delete-environment

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-environment`.

AWS CLI

Um eine Umgebung zu löschen

Im folgenden `delete-environment` Beispiel wird die angegebene Anwendungsumgebung gelöscht.

```
aws appconfig delete-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 2: Erstellen einer Umgebung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteEnvironment](#) in der AWS CLI Befehlsreferenz.

delete-extension-association

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-extension-association`.

AWS CLI

Um eine Erweiterungsverknüpfung zu löschen

Im folgenden `delete-extension-association` Beispiel wird eine Erweiterungsassoziation von AWS AppConfig gelöscht.

```
aws appconfig delete-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [DeleteExtensionAssociation](#) unter AWS CLI Befehlsreferenz.

delete-extension

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-extension`.

AWS CLI

Um eine Erweiterung zu löschen

Im folgenden `delete-extension` Beispiel wird eine Erweiterung von AWS AppConfig gelöscht.

```
aws appconfig delete-extension \  
  --region us-west-2 \  
  --extension-identifizier S3-backup-extension
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [DeleteExtension](#) unter AWS CLI Befehlsreferenz.

delete-hosted-configuration-version

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-hosted-configuration-version`.

AWS CLI

Um eine gehostete Konfigurationsversion zu löschen

Im folgenden `delete-hosted-configuration-version` Beispiel wird eine im gehosteten Konfigurationsspeicher gehostete Konfigurationsversion gelöscht. AWS AppConfig

```
aws appconfig delete-hosted-configuration-version \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f \  
  --version-number 1
```

Ausgabe:: Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteHostedConfigurationVersion](#) unter AWS CLI Befehlsreferenz.

get-application

Das folgende Codebeispiel zeigt, wie man es benutzt `get-application`.

AWS CLI

Um Details einer Anwendung aufzulisten

Das folgende `get-application` Beispiel listet die Details der angegebenen Anwendung auf.

```
aws appconfig get-application \  
  --application-id 339ohji
```

Ausgabe:

```
{
  "Description": "An application used for creating an example.",
  "Id": "339ohji",
  "Name": "example-application"
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [So AWS AppConfig funktioniert](#) es.

- API-Einheiten finden Sie [GetApplication](#) in der AWS CLI Befehlsreferenz.

get-configuration-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `get-configuration-profile`.

AWS CLI

Um Details zum Konfigurationsprofil abzurufen

Im folgenden `get-configuration-profile` Beispiel werden die Details des angegebenen Konfigurationsprofils zurückgegeben.

```
aws appconfig get-configuration-profile \
  --application-id 339ohji \
  --configuration-profile-id ur8hx2f
```

Ausgabe:

```
{
  "ApplicationId": "339ohji",
  "Id": "ur8hx2f",
  "Name": "Example-Configuration-Profile",
  "LocationUri": "ssm-parameter://Example-Parameter",
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [GetConfigurationProfile](#) unter AWS CLI Befehlsreferenz.

get-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-configuration`.

AWS CLI

Um Konfigurationsdetails abzurufen

Das folgende `get-configuration` Beispiel gibt die Konfigurationsdetails der Beispielanwendung zurück. Verwenden Sie bei nachfolgenden Aufrufen von `get-configuration` den `client-configuration-version` Parameter, um die Konfiguration Ihrer Anwendung nur zu aktualisieren, wenn sich die Version geändert hat. Wenn Sie die Konfiguration nur aktualisieren, wenn sich die Version geändert hat, werden zusätzliche Kosten vermieden, die durch den Aufruf von `get-configuration` entstehen.

```
aws appconfig get-configuration \  
  --application "example-application" \  
  --environment "Example-Environment" \  
  --configuration "Example-Configuration-Profile" \  
  --client-id "test-id" \  
  configuration-output-file
```

Inhalt von `configuration-output-file`:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Ausgabe:

```
{  
  "ConfigurationVersion": "1",  
  "ContentType": "application/json"  
}
```

Weitere Informationen finden Sie unter [Schritt 6: Empfangen der Konfiguration](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [GetConfiguration](#) in der AWS CLI Befehlsreferenz.

get-deployment-strategy

Das folgende Codebeispiel zeigt, wie man es benutzt `get-deployment-strategy`.

AWS CLI

Um Details einer Bereitstellungsstrategie abzurufen

Das folgende `get-deployment-strategy` Beispiel listet die Details der angegebenen Bereitstellungsstrategie auf.

```
aws appconfig get-deployment-strategy \  
  --deployment-strategy-id 1225qzk
```

Ausgabe:

```
{  
  "Id": "1225qzk",  
  "Name": "Example-Deployment",  
  "DeploymentDurationInMinutes": 15,  
  "GrowthType": "LINEAR",  
  "GrowthFactor": 25.0,  
  "FinalBakeTimeInMinutes": 0,  
  "ReplicateTo": "SSM_DOCUMENT"  
}
```

Weitere Informationen finden Sie unter [Schritt 4: Erstellen einer Bereitstellungsstrategie](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [GetDeploymentStrategy](#) in der AWS CLI Befehlsreferenz.

get-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `get-deployment`.

AWS CLI

Um Bereitstellungsdetails abzurufen

Im folgenden `get-deployment` Beispiel werden Details zur Bereitstellung für die Anwendung in der angegebenen Umgebung und Bereitstellung aufgeführt.

```
aws appconfig get-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-number 1
```

Ausgabe:

```
{
  "ApplicationId": "339ohji",
  "EnvironmentId": "54j1r29",
  "DeploymentStrategyId": "1225qzk",
  "ConfigurationProfileId": "ur8hx2f",
  "DeploymentNumber": 1,
  "ConfigurationName": "Example-Configuration-Profile",
  "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",
  "ConfigurationVersion": "1",
  "DeploymentDurationInMinutes": 15,
  "GrowthType": "LINEAR",
  "GrowthFactor": 25.0,
  "FinalBakeTimeInMinutes": 0,
  "State": "COMPLETE",
  "EventLog": [
    {
      "EventType": "DEPLOYMENT_COMPLETED",
      "TriggeredBy": "APPCONFIG",
      "Description": "Deployment completed",
      "OccurredAt": "2021-09-17T21:59:03.888000+00:00"
    },
    {
      "EventType": "BAKE_TIME_STARTED",
      "TriggeredBy": "APPCONFIG",
      "Description": "Deployment bake time started",
      "OccurredAt": "2021-09-17T21:58:57.722000+00:00"
    },
    {
      "EventType": "PERCENTAGE_UPDATED",
      "TriggeredBy": "APPCONFIG",
      "Description": "Configuration available to 100.00% of clients",
      "OccurredAt": "2021-09-17T21:55:56.816000+00:00"
    },
    {
      "EventType": "PERCENTAGE_UPDATED",
      "TriggeredBy": "APPCONFIG",
      "Description": "Configuration available to 75.00% of clients",
      "OccurredAt": "2021-09-17T21:52:56.567000+00:00"
    },
    {
      "EventType": "PERCENTAGE_UPDATED",
      "TriggeredBy": "APPCONFIG",

```

```

    "Description": "Configuration available to 50.00% of clients",
    "OccurredAt": "2021-09-17T21:49:55.737000+00:00"
  },
  {
    "EventType": "PERCENTAGE_UPDATED",
    "TriggeredBy": "APPCONFIG",
    "Description": "Configuration available to 25.00% of clients",
    "OccurredAt": "2021-09-17T21:46:55.187000+00:00"
  },
  {
    "EventType": "DEPLOYMENT_STARTED",
    "TriggeredBy": "USER",
    "Description": "Deployment started",
    "OccurredAt": "2021-09-17T21:43:54.205000+00:00"
  }
],
"PercentageComplete": 100.0,
"StartedAt": "2021-09-17T21:43:54.205000+00:00",
"CompletedAt": "2021-09-17T21:59:03.888000+00:00"
}

```

Weitere Informationen finden Sie unter [Schritt 5: Bereitstellen einer Konfiguration](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [GetDeployment](#) in der AWS CLI Befehlsreferenz.

get-environment

Das folgende Codebeispiel zeigt, wie man es benutzt `get-environment`.

AWS CLI

Um Umgebungsdetails abzurufen

Das folgende `get-environment` Beispiel gibt die Details und den Status der angegebenen Umgebung zurück.

```

aws appconfig get-environment \
  --application-id 339ohji \
  --environment-id 54j1r29

```

Ausgabe:

```
{
  "ApplicationId": "339ohji",
  "Id": "54j1r29",
  "Name": "Example-Environment",
  "State": "ReadyForDeployment"
}
```

Weitere Informationen finden Sie unter [Schritt 2: Erstellen einer Umgebung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [GetEnvironment](#) in der AWS CLI Befehlsreferenz.

get-extension-association

Das folgende Codebeispiel zeigt, wie man es benutzt `get-extension-association`.

AWS CLI

Um Details zur Erweiterungsverknüpfung abzurufen

Im folgenden `get-extension-association` Beispiel werden Informationen zu einer Erweiterungszuordnung angezeigt.

```
aws appconfig get-extension-association \
  --region us-west-2 \
  --extension-association-id a1b2c3d4
```

Ausgabe:

```
{
  "Id": "a1b2c3d4",
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-extension/1",
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
  "Parameters": {
    "S3bucket": "FinanceConfigurationBackup"
  },
  "ExtensionVersionNumber": 1
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [GetExtensionAssociation](#) unter AWS CLI Befehlsreferenz.

get-extension

Das folgende Codebeispiel zeigt, wie man es benutzt `get-extension`.

AWS CLI

Um Details zur Erweiterung zu erhalten

Im folgenden `get-extension` Beispiel werden Informationen zu einer Erweiterung angezeigt.

```
aws appconfig get-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension
```

Ausgabe:

```
{  
  "Id": "1A2B3C4D",  
  "Name": "S3-backup-extension",  
  "VersionNumber": 1,  
  "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-  
extension/1",  
  "Actions": {  
    "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [  
      {  
        "Name": "S3backup",  
        "Uri": "arn:aws:lambda:us-  
west-2:123456789012:function:S3backupfunction",  
        "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"  
      }  
    ]  
  },  
  "Parameters": {  
    "S3bucket": {  
      "Required": true  
    }  
  }  
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [GetExtension](#) unter AWS CLI Befehlsreferenz.

get-hosted-configuration-version

Das folgende Codebeispiel zeigt, wie man es benutzt `get-hosted-configuration-version`.

AWS CLI

Um Details zur gehosteten Konfiguration abzurufen

Im folgenden `get-hosted-configuration-version` Beispiel werden die Konfigurationsdetails der gehosteten Konfiguration abgerufen. AWS AppConfig

```
aws appconfig get-hosted-configuration-version \
  --application-id 339ohji \
  --configuration-profile-id ur8hx2f \
  --version-number 1 \
  hosted-configuration-version-output
```

Inhalt von `hosted-configuration-version-output`:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Ausgabe:

```
{
  "ApplicationId": "339ohji",
  "ConfigurationProfileId": "ur8hx2f",
  "VersionNumber": "1",
  "ContentType": "application/json"
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Über den AWS AppConfig gehosteten Konfigurationsspeicher](#).

- API-Einheiten finden Sie [GetHostedConfigurationVersion](#) in der AWS CLI Befehlsreferenz.

list-applications

Das folgende Codebeispiel zeigt, wie man es benutzt `list-applications`.

AWS CLI

Um die verfügbaren Anwendungen aufzulisten

Das folgende `list-applications` Beispiel listet die verfügbaren Anwendungen in Ihrem AWS Konto auf.

```
aws appconfig list-applications
```

Ausgabe:

```
{
  "Items": [
    {
      "Id": "339ohji",
      "Name": "test-application",
      "Description": "An application used for creating an example."
    },
    {
      "Id": "rwalwu7",
      "Name": "Test-Application"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [ListApplications](#) in der AWS CLI Befehlsreferenz.

list-configuration-profiles

Das folgende Codebeispiel zeigt, wie man es benutzt `list-configuration-profiles`.

AWS CLI

Um die verfügbaren Konfigurationsprofile aufzulisten

Das folgende `list-configuration-profiles` Beispiel listet die verfügbaren Konfigurationsprofile für die angegebene Anwendung auf.

```
aws appconfig list-configuration-profiles \  
  --application-id 339ohji
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "ApplicationId": "339ohji",  
      "Id": "ur8hx2f",  
      "Name": "Example-Configuration-Profile",  
      "LocationUri": "ssm-parameter://Example-Parameter"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [ListConfigurationProfiles](#) unter AWS CLI Befehlsreferenz.

list-deployment-strategies

Das folgende Codebeispiel zeigt, wie man es benutzt `list-deployment-strategies`.

AWS CLI

Um die verfügbaren Bereitstellungsstrategien aufzulisten

Im folgenden `list-deployment-strategies` Beispiel werden die verfügbaren Bereitstellungsstrategien in Ihrem AWS Konto aufgeführt.

```
aws appconfig list-deployment-strategies
```

Ausgabe:

```
{  
  "Items": [  
    {
```

```
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "ReplicateTo": "SSM_DOCUMENT"
  },
  {
    "Id": "AppConfig.AllAtOnce",
    "Name": "AppConfig.AllAtOnce",
    "Description": "Quick",
    "DeploymentDurationInMinutes": 0,
    "GrowthType": "LINEAR",
    "GrowthFactor": 100.0,
    "FinalBakeTimeInMinutes": 10,
    "ReplicateTo": "NONE"
  },
  {
    "Id": "AppConfig.Linear50PercentEvery30Seconds",
    "Name": "AppConfig.Linear50PercentEvery30Seconds",
    "Description": "Test/Demo",
    "DeploymentDurationInMinutes": 1,
    "GrowthType": "LINEAR",
    "GrowthFactor": 50.0,
    "FinalBakeTimeInMinutes": 1,
    "ReplicateTo": "NONE"
  },
  {
    "Id": "AppConfig.Canary10Percent20Minutes",
    "Name": "AppConfig.Canary10Percent20Minutes",
    "Description": "AWS Recommended",
    "DeploymentDurationInMinutes": 20,
    "GrowthType": "EXPONENTIAL",
    "GrowthFactor": 10.0,
    "FinalBakeTimeInMinutes": 10,
    "ReplicateTo": "NONE"
  }
]
```

Weitere Informationen finden Sie unter [Schritt 4: Erstellen einer Bereitstellungsstrategie](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [ListDeploymentStrategies](#) in der AWS CLI Befehlsreferenz.

list-deployments

Das folgende Codebeispiel zeigt, wie man es benutzt `list-deployments`.

AWS CLI

Um die verfügbaren Bereitstellungen aufzulisten

Das folgende `list-deployments` Beispiel listet die verfügbaren Bereitstellungen in Ihrem AWS Konto für die angegebene Anwendung und Umgebung auf.

```
aws appconfig list-deployments \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "DeploymentNumber": 1,  
      "ConfigurationName": "Example-Configuration-Profile",  
      "ConfigurationVersion": "1",  
      "DeploymentDurationInMinutes": 15,  
      "GrowthType": "LINEAR",  
      "GrowthFactor": 25.0,  
      "FinalBakeTimeInMinutes": 0,  
      "State": "COMPLETE",  
      "PercentageComplete": 100.0,  
      "StartedAt": "2021-09-17T21:43:54.205000+00:00",  
      "CompletedAt": "2021-09-17T21:59:03.888000+00:00"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Schritt 5: Bereitstellen einer Konfiguration](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [ListDeployments](#) in der AWS CLI Befehlsreferenz.

list-environments

Das folgende Codebeispiel zeigt, wie man es benutzt `list-environments`.

AWS CLI

Um die verfügbaren Umgebungen aufzulisten

Das folgende `list-environments` Beispiel listet die verfügbaren Umgebungen in Ihrem AWS Konto für die angegebene Anwendung auf.

```
aws appconfig list-environments \  
  --application-id 339ohji
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "ApplicationId": "339ohji",  
      "Id": "54j1r29",  
      "Name": "Example-Environment",  
      "State": "ReadyForDeployment"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Schritt 2: Erstellen einer Umgebung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [ListEnvironments](#) in der AWS CLI Befehlsreferenz.

list-extension-associations

Das folgende Codebeispiel zeigt, wie man es benutzt `list-extension-associations`.

AWS CLI

Um alle AWS AppConfig Erweiterungszuordnungen in Ihrem AWS Konto für eine AWS Region aufzulisten

Im folgenden `list-extension-associations` Beispiel werden alle AWS AppConfig Erweiterungszuordnungen für das aktuelle AWS Konto in einer bestimmten AWS Region aufgeführt.

```
aws appconfig list-extension-associations \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "Id": "a1b2c3d4",  
      "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-  
backup-extension/1",  
      "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/  
Finance"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [ListExtensionAssociations](#) unter AWS CLI Befehlsreferenz.

list-extensions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-extensions`.

AWS CLI

Um alle AWS AppConfig Erweiterungen in Ihrem AWS Konto für eine AWS Region aufzulisten

Das folgende `list-extensions` Beispiel listet alle AWS AppConfig Erweiterungen für das AWS Girokonto in einer bestimmten AWS Region auf. Der Befehl gibt benutzerdefinierte und AWS erstellte Erweiterungen zurück.

```
aws appconfig list-extensions \  
  --region us-west-2
```

Ausgabe:

```
{
  "Items": [
    {
      "Id": "1A2B3C4D",
      "Name": "S3-backup-extension",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1"
    },
    {
      "Id": "AWS.AppConfig.FeatureFlags",
      "Name": "AppConfig Feature Flags Helper",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.FeatureFlags/1",
      "Description": "Validates AppConfig feature flag data automatically
against a JSON schema that includes structure and constraints. Also transforms
feature flag data prior to sending to the client. This extension is automatically
associated to configuration profiles with type \"AWS.AppConfig.FeatureFlags\"."
    },
    {
      "Id": "AWS.AppConfig.JiraIntegration",
      "Name": "AppConfig integration with Atlassian Jira",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.JiraIntegration/1",
      "Description": "Exports feature flag data from AWS AppConfig into
Jira. The lifecycle of each feature flag in AppConfig is tracked in Jira as an
individual issue. Customers can see in Jira when flags are updated, turned on or
off. Works in conjunction with the AppConfig app in the Atlassian Marketplace and
is automatically associated to configuration profiles configured within that app."
    },
    {
      "Id": "AWS.AppConfig.DeploymentNotificationsToEventBridge",
      "Name": "AppConfig deployment events to Amazon EventBridge",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToEventBridge/1",
      "Description": "Sends events to Amazon EventBridge when a deployment
of configuration data in AppConfig is started, completed, or rolled back. Can
be associated to the following resources in AppConfig: Application, Environment,
Configuration Profile."
    },
  ],
}
```

```

    {
      "Id": "AWS.AppConfig.DeploymentNotificationsToSqs",
      "Name": "AppConfig deployment events to Amazon SQS",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToSqs/1",
      "Description": "Sends messages to the configured Amazon SQS queue when
a deployment of configuration data in AppConfig is started, completed, or rolled
back. Can be associated to the following resources in AppConfig: Application,
Environment, Configuration Profile."
    },
    {
      "Id": "AWS.AppConfig.DeploymentNotificationsToSns",
      "Name": "AppConfig deployment events to Amazon SNS",
      "VersionNumber": 1,
      "Description": "Sends events to the configured Amazon SNS topic when
a deployment of configuration data in AppConfig is started, completed, or rolled
back. Can be associated to the following resources in AppConfig: Application,
Environment, Configuration Profile."
    }
  ]
}

```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [ListExtensions](#) unter AWS CLI Befehlsreferenz.

list-hosted-configuration-versions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-hosted-configuration-versions`.

AWS CLI

Um die verfügbaren Versionen der gehosteten Konfiguration aufzulisten

Im folgenden `list-hosted-configuration-versions` Beispiel werden die Konfigurationsversionen aufgeführt, die im AWS AppConfig gehosteten Konfigurationsspeicher für die angegebene Anwendung und das angegebene Konfigurationsprofil gehostet werden.

```

aws appconfig list-hosted-configuration-versions \
  --application-id 339ohji \

```

```
--configuration-profile-id ur8hx2f
```

Ausgabe:

```
{
  "Items": [
    {
      "ApplicationId": "339ohji",
      "ConfigurationProfileId": "ur8hx2f",
      "VersionNumber": 1,
      "ContentType": "application/json"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Über den AWS AppConfig gehosteten Konfigurationsspeicher](#).

- API-Einheiten finden Sie [ListHostedConfigurationVersions](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um die Tags einer Anwendung aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags einer angegebenen Anwendung auf.

```
aws appconfig list-tags-for-resource \
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

Ausgabe:

```
{
  "Tags": {
    "group1": "1"
  }
}
```



```
}
```

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `start-deployment`.

AWS CLI

Um eine Konfigurationsbereitstellung zu starten

Im folgenden `start-deployment` Beispiel wird eine Bereitstellung für die Anwendung unter Verwendung der angegebenen Umgebung, Bereitstellungsstrategie und des angegebenen Konfigurationsprofils gestartet.

```
aws appconfig start-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-strategy-id 1225qzk \  
  --configuration-profile-id ur8hx2f \  
  --configuration-version 1
```

Ausgabe:

```
{  
  "ApplicationId": "339ohji",  
  "EnvironmentId": "54j1r29",  
  "DeploymentStrategyId": "1225qzk",  
  "ConfigurationProfileId": "ur8hx2f",  
  "DeploymentNumber": 1,  
  "ConfigurationName": "Example-Configuration-Profile",  
  "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",  
  "ConfigurationVersion": "1",  
  "DeploymentDurationInMinutes": 15,  
  "GrowthType": "LINEAR",  
  "GrowthFactor": 25.0,  
  "FinalBakeTimeInMinutes": 0,
```

```

    "State": "DEPLOYING",
    "EventLog": [
      {
        "EventType": "DEPLOYMENT_STARTED",
        "TriggeredBy": "USER",
        "Description": "Deployment started",
        "OccurredAt": "2021-09-17T21:43:54.205000+00:00"
      }
    ],
    "PercentageComplete": 0.0,
    "StartedAt": "2021-09-17T21:43:54.205000+00:00"
  }

```

Weitere Informationen finden Sie unter [Schritt 5: Bereitstellen einer Konfiguration](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [StartDeployment](#) in der AWS CLI Befehlsreferenz.

stop-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `stop-deployment`.

AWS CLI

Um die Bereitstellung der Konfiguration zu beenden

Im folgenden `stop-deployment` Beispiel wird die Bereitstellung einer Anwendungskonfiguration in der angegebenen Umgebung gestoppt.

```

aws appconfig stop-deployment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --deployment-number 2

```

Ausgabe:

```

{
  "DeploymentNumber": 0,
  "DeploymentDurationInMinutes": 0,
  "GrowthFactor": 0.0,
  "FinalBakeTimeInMinutes": 0,
  "PercentageComplete": 0.0
}

```

```
}
```

Weitere Informationen finden Sie unter [Schritt 5: Bereitstellen einer Konfiguration](#) im AWS AppConfig Benutzerhandbuch.

- APIEinzelheiten finden Sie [StopDeployment](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um eine Anwendung zu taggen

Im folgenden `tag-resource` Beispiel wird eine Anwendungsressource markiert.

```
aws appconfig tag-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \  
  --tags '{"group1" : "1"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- APIEinzelheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um ein Tag aus einer Anwendung zu entfernen

Im folgenden `untag-resource` Beispiel wird das `group1`-Tag aus der angegebenen Anwendung entfernt.

```
aws appconfig untag-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \  
  --tag-key group1
```

```
--tag-keys ["group1"]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- APIEinzelheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-application

Das folgende Codebeispiel zeigt, wie man es benutzt `update-application`.

AWS CLI

Um eine Anwendung zu aktualisieren

Im folgenden `update-application` Beispiel wird der Name der angegebenen Anwendung aktualisiert.

```
aws appconfig update-application \
  --application-id 339ohji \
  --name "Example-Application"
```

Ausgabe:

```
{
  "Id": "339ohji",
  "Name": "Example-Application",
  "Description": "An application used for creating an example."
}
```

Weitere Informationen finden Sie unter [Schritt 1: Erstellen einer AWS AppConfig Anwendung](#) im AWS AppConfig Benutzerhandbuch.

- APIEinzelheiten finden Sie [UpdateApplication](#) in der AWS CLI Befehlsreferenz.

update-configuration-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `update-configuration-profile`.

AWS CLI

Um ein Konfigurationsprofil zu aktualisieren

Im folgenden `update-configuration-profile` Beispiel wird die Beschreibung des angegebenen Konfigurationsprofils aktualisiert.

```
aws appconfig update-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f \  
  --description "Configuration profile used for examples."
```

Ausgabe:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "ur8hx2f",  
  "Name": "Example-Configuration-Profile",  
  "Description": "Configuration profile used for examples.",  
  "LocationUri": "ssm-parameter://Example-Parameter",  
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"  
}
```

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateConfigurationProfile](#) unter AWS CLI Befehlsreferenz.

update-deployment-strategy

Das folgende Codebeispiel zeigt, wie man es benutzt `update-deployment-strategy`.

AWS CLI

Um eine Bereitstellungsstrategie zu aktualisieren

Im folgenden `update-deployment-strategy` Beispiel wird die endgültige Backzeit in der angegebenen Bereitstellungsstrategie auf 20 Minuten aktualisiert.

```
aws appconfig update-deployment-strategy \  
  --deployment-strategy-id 1225qzk \  
  --backlog-size 20
```

```
--final-bake-time-in-minutes 20
```

Ausgabe:

```
{
  "Id": "1225qzk",
  "Name": "Example-Deployment",
  "DeploymentDurationInMinutes": 15,
  "GrowthType": "LINEAR",
  "GrowthFactor": 25.0,
  "FinalBakeTimeInMinutes": 20,
  "ReplicateTo": "SSM_DOCUMENT"
}
```

Weitere Informationen finden Sie unter [Schritt 4: Erstellen einer Bereitstellungsstrategie](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDeploymentStrategy](#) in der AWS CLI Befehlsreferenz.

update-environment

Das folgende Codebeispiel zeigt, wie man es benutzt `update-environment`.

AWS CLI

Um eine Umgebung zu aktualisieren

Das folgende `update-environment` Beispiel aktualisiert die Beschreibung einer Umgebung.

```
aws appconfig update-environment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --description "An environment for examples."
```

Ausgabe:

```
{
  "ApplicationId": "339ohji",
  "Id": "54j1r29",
  "Name": "Example-Environment",
  "Description": "An environment for examples.",
}
```

```
"State": "RolledBack"
}
```

Weitere Informationen finden Sie unter [Schritt 2: Erstellen einer Umgebung](#) im AWS AppConfig Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateEnvironment](#) in der AWS CLI Befehlsreferenz.

update-extension-association

Das folgende Codebeispiel zeigt, wie man es benutzt `update-extension-association`.

AWS CLI

Um eine AWS AppConfig Erweiterungsverknüpfung zu aktualisieren

Im folgenden `update-extension-association` Beispiel wird einer Erweiterungsassoziation ein neuer Parameterwert hinzugefügt AWS AppConfig.

```
aws appconfig update-extension-association \
  --region us-west-2 \
  --extension-association-id a1b2c3d4 \
  --parameters S3bucket=FinanceMobileApp
```

Ausgabe:

```
{
  "Id": "a1b2c3d4",
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-extension/1",
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
  "Parameters": {
    "S3bucket": "FinanceMobileApp"
  },
  "ExtensionVersionNumber": 1
}
```

Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- API-Einheiten finden Sie [UpdateExtensionAssociation](#) unter AWS CLI Befehlsreferenz.

update-extension

Das folgende Codebeispiel zeigt, wie man es benutzt `update-extension`.

AWS CLI

Um eine AWS AppConfig Erweiterung zu aktualisieren

Das folgende `update-extension` Beispiel fügt einer Erweiterung in einen zusätzlichen Parameter Key hinzu AWS AppConfig.

```
aws appconfig update-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension \  
  --parameters S3bucket={Required=true}, CampaignID={Required=false}
```

Ausgabe:

```
{  
  "Id": "1A2B3C4D",  
  "Name": "S3-backup-extension",  
  "VersionNumber": 1,  
  "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",  
  "Actions": {  
    "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [  
      {  
        "Name": "S3backup",  
        "Uri": "arn:aws:lambda:us-west-2:123456789012:function:S3backupfunction",  
        "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"  
      }  
    ]  
  },  
  "Parameters": {  
    "CampaignID": {  
      "Required": false  
    },  
    "S3bucket": {  
      "Required": true  
    }  
  }  
}
```


Weitere Informationen finden Sie im AWS AppConfig Benutzerhandbuch unter [Arbeiten mit AWS AppConfig Erweiterungen](#).

- APIEinzelheiten finden Sie [UpdateExtension](#) unter AWS CLI Befehlsreferenz.

validate-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `validate-configuration`.

AWS CLI

Um eine Konfiguration zu validieren

Im folgenden `validate-configuration` Beispiel werden die Validatoren in einem Konfigurationsprofil verwendet, um eine Konfiguration zu validieren.

```
aws appconfig validate-configuration \  
  --application-id abc1234 \  
  --configuration-profile-id ur8hx2f \  
  --configuration-version 1
```

Der Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Schritt 3: Konfiguration und Konfigurationsprofil erstellen](#) im AWS AppConfig Benutzerhandbuch.

- APIEinzelheiten finden Sie [ValidateConfiguration](#) unter AWS CLI Befehlsreferenz.

Beispiele für Application Auto Scaling mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Application Auto Scaling Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-scaling-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-scaling-policy`.

AWS CLI

Um eine Skalierungsrichtlinie zu löschen

In diesem Beispiel wird eine Skalierungsrichtlinie für die Amazon ECS Service-Web-App gelöscht, die im Standard-Cluster ausgeführt wird.

Befehl:

```
aws application-autoscaling delete-scaling-policy --policy-name web-app-cpu-lt-25 --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app --service-namespace ecs
```

- API-Einheiten finden Sie unter [DeleteScalingPolicy AWS CLI](#) Befehlsreferenz.

delete-scheduled-action

Das folgende Codebeispiel zeigt die Verwendung `delete-scheduled-action`.

AWS CLI

Löschen einer geplanten Aktion

Im folgenden `delete-scheduled-action` Beispiel wird die angegebene geplante Aktion aus der angegebenen Amazon AppStream 2.0-Flotte gelöscht:

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace appstream \
  --scalable-dimension appstream:fleet:DesiredCapacity \
  --resource-id fleet/sample-fleet \
  --scheduled-action-name my-recurring-action
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Benutzerhandbuch für Application Auto Scaling.

- API-Einheiten finden Sie unter [DeleteScheduledAction AWS CLI Befehlsreferenz](#).

deregister-scalable-target

Das folgende Codebeispiel zeigt die Verwendung `deregister-scalable-target`.

AWS CLI

Um die Registrierung eines skalierbaren Ziels aufzuheben

In diesem Beispiel wird die Registrierung eines skalierbaren Ziels für einen ECS Amazon-Service namens `Web-App` aufgehoben, der im Standard-Cluster ausgeführt wird.

Befehl:

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

In diesem Beispiel wird die Registrierung eines skalierbaren Ziels für eine benutzerdefinierte Ressource aufgehoben. Die `custom-resource-id.txt`-Datei enthält eine Zeichenfolge, die die Ressourcen-ID identifiziert, die für eine benutzerdefinierte Ressource der Pfad zu der benutzerdefinierten Ressource über Ihren Amazon API Gateway-Endpunkt ist.

Befehl:

```
aws application-autoscaling deregister-scalable-target --service-namespace custom-resource --scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

Inhalt der `custom-resource-id.txt`-Datei:

```
https://example.execute-api.us-west-2.amazonaws.com/prod/scalableTargetDimensions/1-23456789
```

- API-Einheiten finden Sie [DeregisterScalableTarget](#) in der AWS CLI Befehlsreferenz.

describe-scalable-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-scalable-targets`.

AWS CLI

Um skalierbare Ziele zu beschreiben

Das folgende `describe-scalable-targets` Beispiel beschreibt die skalierbaren Ziele für den `ecs` Service-Namespace.

```
aws application-autoscaling describe-scalable-targets \  
  --service-namespace ecs
```

Ausgabe:

```
{  
  "ScalableTargets": [  
    {  
      "ServiceNamespace": "ecs",  
      "ScalableDimension": "ecs:service:DesiredCount",  
      "ResourceId": "service/default/web-app",  
      "MinCapacity": 1,  
      "MaxCapacity": 10,  
      "RoleARN": "arn:aws:iam::123456789012:role/  
aws-service-role/ecs.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_ECSService",  
      "CreationTime": 1462558906.199,  
      "SuspendedState": {  
        "DynamicScalingOutSuspended": false,  
        "ScheduledScalingSuspended": false,  
        "DynamicScalingInSuspended": false  
      },  
      "ScalableTargetARN": "arn:aws:application-autoscaling:us-  
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"  
    }  
  ]  
}
```

Weitere Informationen [finden Sie im Application Auto Scaling-Benutzerhandbuch unter AWS Dienste, die Sie mit Application Auto Scaling verwenden können.](#)

- API-Einheiten finden Sie [DescribeScalableTargets](#) in der AWS CLI Befehlsreferenz.

describe-scaling-activities

Das folgende Codebeispiel zeigt die Verwendung `describe-scaling-activities`.

AWS CLI

Beispiel 1: Um Skalierungsaktivitäten für den angegebenen ECS Amazon-Service zu beschreiben

Das folgende `describe-scaling-activities` Beispiel beschreibt die Skalierungsaktivitäten für einen ECS Amazon-Service namens `web-app`, der im `default` Cluster ausgeführt wird. Die Ausgabe zeigt eine Skalierungsaktivität, die durch eine Skalierungsrichtlinie initiiert wurde.

```
aws application-autoscaling describe-scaling-activities \
  --service-namespace ecs \
  --resource-id service/default/web-app
```

Ausgabe:

```
{
  "ScalingActivities": [
    {
      "ScalableDimension": "ecs:service:DesiredCount",
      "Description": "Setting desired count to 1.",
      "ResourceId": "service/default/web-app",
      "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",
      "StartTime": 1462575838.171,
      "ServiceNamespace": "ecs",
      "EndTime": 1462575872.111,
      "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered
policy web-app-cpu-lt-25",
      "StatusMessage": "Successfully set desired count to 1. Change
successfully fulfilled by ecs.",
      "StatusCode": "Successful"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Skalierungsaktivitäten für Application Auto Scaling](#) im Application Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um Skalierungsaktivitäten für die angegebene DynamoDB-Tabelle zu beschreiben

Das folgende `describe-scaling-activities` Beispiel beschreibt die Skalierungsaktivitäten für eine DynamoDB-Tabelle namens `TestTable`. Die Ausgabe zeigt Skalierungsaktivitäten, die durch zwei verschiedene geplante Aktionen ausgelöst wurden.

```
aws application-autoscaling describe-scaling-activities \  
  --service-namespace dynamodb \  
  --resource-id table/TestTable
```

Ausgabe:

```
{  
  "ScalingActivities": [  
    {  
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
      "Description": "Setting write capacity units to 10.",  
      "ResourceId": "table/my-table",  
      "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",  
      "StartTime": 1561574415.086,  
      "ServiceNamespace": "dynamodb",  
      "EndTime": 1561574449.51,  
      "Cause": "maximum capacity was set to 10",  
      "StatusMessage": "Successfully set write capacity units to 10. Change  
successfully fulfilled by dynamodb.",  
      "StatusCode": "Successful"  
    },  
    {  
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
      "Description": "Setting min capacity to 5 and max capacity to 10",  
      "ResourceId": "table/my-table",  
      "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",  
      "StartTime": 1561574414.644,  
      "ServiceNamespace": "dynamodb",  
      "Cause": "scheduled action name my-second-scheduled-action was  
triggered",  
      "StatusMessage": "Successfully set min capacity to 5 and max capacity to  
10",  
      "StatusCode": "Successful"  
    },  
    {  
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
      "Description": "Setting write capacity units to 15.",  
      "ResourceId": "table/my-table",  
      "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
```

```

        "StartTime": 1561574108.904,
        "ServiceNamespace": "dynamodb",
        "EndTime": 1561574140.255,
        "Cause": "minimum capacity was set to 15",
        "StatusMessage": "Successfully set write capacity units to 15. Change
successfully fulfilled by dynamodb.",
        "StatusCode": "Successful"
    },
    {
        "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
        "Description": "Setting min capacity to 15 and max capacity to 20",
        "ResourceId": "table/my-table",
        "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
        "StartTime": 1561574108.512,
        "ServiceNamespace": "dynamodb",
        "Cause": "scheduled action name my-first-scheduled-action was
triggered",
        "StatusMessage": "Successfully set min capacity to 15 and max capacity
to 20",
        "StatusCode": "Successful"
    }
]
}

```

Weitere Informationen finden Sie unter [Skalierungsaktivitäten für Application Auto Scaling](#) im Application Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeScalingActivities](#) unter AWS CLI Befehlsreferenz.

describe-scaling-policies

Das folgende Codebeispiel zeigt die Verwendung `describe-scaling-policies`.

AWS CLI

Um Skalierungsrichtlinien zu beschreiben

Dieser Beispielbefehl beschreibt die Skalierungsrichtlinien für den ECS-Service-Namespace.

Befehl:

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

Ausgabe:

```
{
  "ScalingPolicies": [
    {
      "PolicyName": "web-app-cpu-gt-75",
      "ScalableDimension": "ecs:service:DesiredCount",
      "ResourceId": "service/default/web-app",
      "CreationTime": 1462561899.23,
      "StepScalingPolicyConfiguration": {
        "Cooldown": 60,
        "StepAdjustments": [
          {
            "ScalingAdjustment": 200,
            "MetricIntervalLowerBound": 0.0
          }
        ],
        "AdjustmentType": "PercentChangeInCapacity"
      },
      "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-gt-75",
      "PolicyType": "StepScaling",
      "Alarms": [
        {
          "AlarmName": "web-app-cpu-gt-75",
          "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-gt-75"
        }
      ],
      "ServiceNamespace": "ecs"
    },
    {
      "PolicyName": "web-app-cpu-lt-25",
      "ScalableDimension": "ecs:service:DesiredCount",
      "ResourceId": "service/default/web-app",
      "CreationTime": 1462562575.099,
      "StepScalingPolicyConfiguration": {
        "Cooldown": 1,
        "StepAdjustments": [
          {
            "ScalingAdjustment": -50,
            "MetricIntervalUpperBound": 0.0
          }
        ]
      }
    }
  ]
}
```



```

        ],
        "AdjustmentType": "PercentChangeInCapacity"
    },
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-1t-25",
    "PolicyType": "StepScaling",
    "Alarms": [
        {
            "AlarmName": "web-app-cpu-1t-25",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-1t-25"
        }
    ],
    "ServiceNamespace": "ecs"
}
]
}

```

- API-Einheiten finden Sie [DescribeScalingPolicies](#) in der AWS CLI Befehlsreferenz.

describe-scheduled-actions

Das folgende Codebeispiel zeigt die Verwendung `describe-scheduled-actions`.

AWS CLI

Um geplante Aktionen zu beschreiben

Im folgenden `describe-scheduled-actions` Beispiel werden Details zu den geplanten Aktionen für den angegebenen Dienst-Namespaces angezeigt:

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace dynamodb
```

Ausgabe:

```
{
  "ScheduledActions": [
    {
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
```

```

    "Schedule": "at(2019-05-20T18:35:00)",
    "ResourceId": "table/my-table",
    "CreationTime": 1561571888.361,
    "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-first-scheduled-action",
    "ScalableTargetAction": {
      "MinCapacity": 15,
      "MaxCapacity": 20
    },
    "ScheduledActionName": "my-first-scheduled-action",
    "ServiceNamespace": "dynamodb"
  },
  {
    "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
    "Schedule": "at(2019-05-20T18:40:00)",
    "ResourceId": "table/my-table",
    "CreationTime": 1561571946.021,
    "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-second-scheduled-action",
    "ScalableTargetAction": {
      "MinCapacity": 5,
      "MaxCapacity": 10
    },
    "ScheduledActionName": "my-second-scheduled-action",
    "ServiceNamespace": "dynamodb"
  }
]
}

```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Benutzerhandbuch für Application Auto Scaling.

- API-Einheiten finden Sie unter [DescribeScheduledActions AWS CLI](#) Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für ein skalierbares Ziel aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Namen und Werte der Tag-Schlüssel auf, die an das durch `its` angegebene skalierbare Ziel angehängt sindARN.

```
aws application-autoscaling list-tags-for-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123
```

Ausgabe:

```
{  
  "Tags": {  
    "environment": "production"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging-Unterstützung für Application Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-scaling-policy

Das folgende Codebeispiel zeigt die Verwendung `put-scaling-policy`.

AWS CLI

Beispiel 1: So wenden Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung mit einer vordefinierten Metrikspezifikation an

Im folgenden `put-scaling-policy` Beispiel wird eine Skalierungsrichtlinie für die Zielverfolgung mit einer vordefinierten Metrikspezifikation auf einen ECS Amazon-Service namens Web-App im Standard-Cluster angewendet. Die Richtlinie hält die durchschnittliche CPU Auslastung des Service bei 75 Prozent, wobei die Abklingzeiten für Scale-Out und Scale-In 60 Sekunden betragen. Die Ausgabe enthält die Namen ARNs und der beiden CloudWatch Alarmer, die in Ihrem Namen erstellt wurden.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \  
  --scalable-dimension ecs:service:DesiredCount \  
  --resource-id service/default/web-app \  
  --policy-name cpu75-target-tracking-scaling-policy --policy-  
type TargetTrackingScaling \  

```

```
--target-tracking-scaling-policy-configuration file://config.json
```

In diesem Beispiel wird davon ausgegangen, dass Sie im aktuellen Verzeichnis eine config.json-Datei mit dem folgenden Inhalt haben:

```
{
  "TargetValue": 75.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
  },
  "ScaleOutCooldown": 60,
  "ScaleInCooldown": 60
}
```

Ausgabe:

```
{
  "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/cpu75-target-tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca",
      "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca"
    },
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d",
      "AlarmName": "TargetTracking-service/default/web-app-AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
    }
  ]
}
```

Beispiel 2: So wenden Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung mit einer benutzerdefinierten Metrikspezifikation an

Im folgenden `put-scaling-policy` Beispiel wird eine Skalierungsrichtlinie für die Zielverfolgung mit einer benutzerdefinierten Metrikspezifikation auf einen ECS Amazon-Service namens `Web-App` im Standard-Cluster angewendet. Die Richtlinie hält die durchschnittliche Auslastung des Service bei 75 Prozent, wobei die Abklingzeiten für `Scale-Out` und `Scale-In` 60 Sekunden betragen. Die Ausgabe enthält die Namen ARNs und der beiden CloudWatch Alarmer, die in Ihrem Namen erstellt wurden.

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \  
--scalable-dimension ecs:service:DesiredCount \  
--resource-id service/default/web-app \  
--policy-name cms75-target-tracking-scaling-policy \  
--policy-type TargetTrackingScaling \  
--target-tracking-scaling-policy-configuration file://config.json
```

In diesem Beispiel wird davon ausgegangen, dass Sie im aktuellen Verzeichnis eine `config.json`-Datei mit dem folgenden Inhalt haben:

```
{  
  "TargetValue":75.0,  
  "CustomizedMetricSpecification":{  
    "MetricName":"MyUtilizationMetric",  
    "Namespace":"MyNamespace",  
    "Dimensions": [  
      {  
        "Name":"MyOptionalMetricDimensionName",  
        "Value":"MyOptionalMetricDimensionValue"  
      }  
    ],  
    "Statistic":"Average",  
    "Unit":"Percent"  
  },  
  "ScaleOutCooldown": 60,  
  "ScaleInCooldown": 60  
}
```

Ausgabe:

```
{  
  "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:  
8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/default/web-  
app:policyName/cms75-target-tracking-scaling-policy",
```

```

    "Alarms": [
      {
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
        "AlarmName": "TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
      },
      {
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
        "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
      }
    ]
  }
}

```

Beispiel 3: So wenden Sie eine Skalierungsrichtlinie für die Ziel-Nachverfolgung nur für die horizontale Skalierung nach oben an

Im folgenden `put-scaling-policy` Beispiel wird eine Skalierungsrichtlinie für die Zielverfolgung auf einen ECS Amazon-Service angewendet, der `web-app` im Standard-Cluster aufgerufen wird. Die Richtlinie wird verwendet, um den ECS Service zu skalieren, wenn die `RequestCountPerTarget` Metrik aus dem Application Load Balancer den Schwellenwert überschreitet. Die Ausgabe enthält den ARN und den Namen des CloudWatch Alarms, der in Ihrem Namen erstellt wurde.

```

aws application-autoscaling put-scaling-policy \
  --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/default/web-app \
  --policy-name alb-scale-out-target-tracking-scaling-policy \
  --policy-type TargetTrackingScaling \
  --target-tracking-scaling-policy-configuration file://config.json

```

Inhalt von `config.json`:

```

{
  "TargetValue": 1000.0,
  "PredefinedMetricSpecification": {

```

```

    "PredefinedMetricType": "ALBRequestCountPerTarget",
    "ResourceLabel": "app/EC2Co-EcsE1-1TKLTMITMM0E0/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
  },
  "ScaleOutCooldown": 60,
  "ScaleInCooldown": 60,
  "DisableScaleIn": true
}

```

Ausgabe:

```

{
  "PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/alb-
scale-out-target-tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
      "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Target Tracking Scaling-Richtlinien für Application Auto Scaling](#) im AWS Application Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutScalingPolicy](#) unter AWS CLI Befehlsreferenz.

put-scheduled-action

Das folgende Codebeispiel zeigt die Verwendung `put-scheduled-action`.

AWS CLI

So fügen Sie einer DynamoDB-Tabelle eine geplante Aktion hinzu

In diesem Beispiel wird eine geplante Aktion zu einer DynamoDB-Tabelle hinzugefügt, die aufgerufen wird `TestTable`, um nach einem wiederkehrenden Zeitplan zu skalieren. Gemäß dem angegebenen Zeitplan (täglich um 12:15 Uhr UTC) wird Application Auto Scaling auf den von

angegebenen Wert skaliert MinCapacity, wenn die aktuelle Kapazität unter dem für angegebenen Wert liegt. MinCapacity

Befehl:

```
aws application-autoscaling put-scheduled-action --service-namespace dynamodb
--scheduled-action-name my-recurring-action --schedule "cron(15 12 * * ? *)" --
resource-id table/TestTable --scalable-dimension dynamodb:table:WriteCapacityUnits
--scalable-target-action MinCapacity=6
```

Weitere Informationen finden Sie unter Scheduled Scaling im Application Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutScheduledAction](#) in der AWS CLI Befehlsreferenz.

register-scalable-target

Das folgende Codebeispiel zeigt die Verwendung `register-scalable-target`.

AWS CLI

Beispiel 1: Um einen ECS Dienst als skalierbares Ziel zu registrieren

Im folgenden `register-scalable-target` Beispiel wird ein ECS Amazon-Service bei Application Auto Scaling registriert. Außerdem wird dem skalierbaren Ziel ein Tag mit dem Schlüsselnamen `environment` und `production` dem Wert hinzugefügt.

```
aws application-autoscaling register-scalable-target \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--min-capacity 1 --max-capacity 10 \
--tags environment=production
```

Ausgabe:

```
{
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```


Beispiele für andere AWS Dienste und benutzerdefinierte Ressourcen finden Sie in den Themen unter [AWS Dienste, die Sie mit Application Auto Scaling verwenden können im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

Beispiel 2: So setzen Sie die Skalierungsaktivitäten für ein skalierbares Ziel aus

Im folgenden `register-scalable-target` Beispiel werden die Skalierungsaktivitäten für ein vorhandenes skalierbares Ziel ausgesetzt.

```
aws application-autoscaling register-scalable-target \  
  --service-namespace dynamodb \  
  --scalable-dimension dynamodb:table:ReadCapacityUnits \  
  --resource-id table/my-table \  
  --suspended-  
state DynamicScalingInSuspended=true,DynamicScalingOutSuspended=true,ScheduledScalingSuspen
```

Ausgabe:

```
{  
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-  
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"  
}
```

Weitere Informationen finden Sie unter [Aussetzen und Wiederaufnehmen der Skalierung für Application Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

Beispiel 3: Um die Skalierungsaktivitäten für ein skalierbares Ziel wieder aufzunehmen

Im folgenden `register-scalable-target` Beispiel werden die Skalierungsaktivitäten für ein vorhandenes skalierbares Ziel wieder aufgenommen.

```
aws application-autoscaling register-scalable-target \  
  --service-namespace dynamodb \  
  --scalable-dimension dynamodb:table:ReadCapacityUnits \  
  --resource-id table/my-table \  
  --suspended-  
state DynamicScalingInSuspended=false,DynamicScalingOutSuspended=false,ScheduledScalingSuspe
```

Ausgabe:

```
{
```

```
"ScalableTargetARN": "arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Weitere Informationen finden Sie unter [Aussetzen und Wiederaufnehmen der Skalierung für Application Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterScalableTarget](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einem skalierbaren Ziel ein Tag hinzuzufügen

Im folgenden `tag-resource` Beispiel wird dem durch `its` angegebenen skalierbaren Ziel ein Tag mit `production` dem Schlüsselnamen `environment` und dem Wert `ARN` hinzugefügt.

```
aws application-autoscaling tag-resource \
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123 \
  --tags environment=production
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging-Unterstützung für Application Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einem skalierbaren Ziel zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag-Paar mit dem Schlüsselnamen `environment` aus dem durch `its` angegebenen skalierbaren Ziel entfernt.

```
aws application-autoscaling untag-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123 \  
  --tag-keys "environment"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging-Unterstützung für Application Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

Beispiele für den Application Discovery Service mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie den AWS Command Line Interface with Application Discovery Service verwenden.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-agents

Das folgende Codebeispiel zeigt die Verwendung `describe-agents`.

AWS CLI

Beschreiben Sie Agenten mit bestimmten `collectionStatus` Zuständen

Dieser Beispielbefehl beschreibt Inkasso-Agenten mit dem Sammlungsstatus `STARTED` oder `STOPPED`.

Befehl:

```
aws discovery describe-agents --filters
name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
results 3
```

Ausgabe:

```
{
  "Snapshots": [
    {
      "version": "1.0.40.0",
      "agentType": "EC2",
      "hostName": "ip-172-31-40-234",
      "collectionStatus": "STOPPED",
      "agentNetworkInfoList": [
        {
          "macAddress": "06:b5:97:14:fc:0d",
          "ipAddress": "172.31.40.234"
        }
      ],
      "health": "UNKNOWN",
      "agentId": "i-003305c02a776e883",
      "registeredTime": "2016-12-09T19:05:06Z",
      "lastHealthPingTime": "2016-12-09T19:05:10Z"
    },
    {
      "version": "1.0.40.0",
      "agentType": "EC2",
      "hostName": "ip-172-31-39-64",
      "collectionStatus": "STARTED",
      "agentNetworkInfoList": [
        {
          "macAddress": "06:a1:0e:c7:b2:73",
          "ipAddress": "172.31.39.64"
        }
      ],
      "health": "SHUTDOWN",
      "agentId": "i-003a5e5e2b36cf8bd",
      "registeredTime": "2016-11-16T16:36:25Z",
      "lastHealthPingTime": "2016-11-16T16:47:37Z"
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie [DescribeAgents](#) in der AWS CLI Befehlsreferenz.

describe-configurations

Das folgende Codebeispiel zeigt die Verwendung `describe-configurations`.

AWS CLI

Beschreiben Sie ausgewählte Asset-Konfigurationen

Dieser Beispielbefehl beschreibt die Konfigurationen von zwei angegebenen Servern. Die Aktion erkennt den Asset-Typ anhand der Konfigurations-ID. Pro Befehl ist nur ein Asset-Typ zulässig.

Befehl:

```
aws discovery describe-configurations --configuration-ids "d-server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

Ausgabe:

```
{
  "configurations": [
    {
      "server.performance.maxCpuUsagePct": "0.0",
      "server.performance.maxDiskReadIOPS": "0.0",
      "server.performance.avgCpuUsagePct": "0.0",
      "server.type": "EC2",
      "server.performance.maxNetworkReadsPerSecondInKB": "0.19140625",
      "server.hostName": "ip-172-31-35-152",
      "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
      "server.tags.hasMoreValues": "false",
      "server.performance.minFreeRAMInKB": "1543496.0",
      "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
      "server.performance.maxDiskReadsPerSecondInKB": "0.0",
      "server.applications": "[]",
      "server.performance.numDisks": "1",
      "server.performance.numCpus": "1",
      "server.performance.numCores": "1",
      "server.performance.maxDiskWriteIOPS": "0.0",
      "server.performance.maxNetworkWritesPerSecondInKB": "0.82421875",
      "server.performance.avgDiskWritesPerSecondInKB": "0.0",
```

```

        "server.networkInterfaceInfo": "[{\"name\":\"eth0\",
\\\"macAddress\\\":\\\"06:A7:7D:3F:54:57\\\",\\\"ipAddress\\\":\\\"172.31.35.152\\\",\\\"netMask\\\":
\\\"255.255.240.0\\\"},{\"name\":\"lo\",\\\"macAddress\\\":\\\"00:00:00:00:00:00\\\",\\\"ipAddress
\\\":\\\"127.0.0.1\\\",\\\"netMask\\\":\\\"255.0.0.0\\\"},{\"name\":\"eth0\",\\\"macAddress\\\":
\\\"06:A7:7D:3F:54:57\\\",\\\"ipAddress\\\":\\\"fe80::4a7:7dff:fe3f:5457\\\"},{\"name\":\"lo\",
\\\"macAddress\\\":\\\"00:00:00:00:00:00\\\",\\\"ipAddress\\\":\\\"::1\\\"}]",
        "server.performance.avgNetworkReadsPerSecondInKB":
"0.049153645833333333",
        "server.tags": "[]",
        "server.applications.hasMoreValues": "false",
        "server.timeOfCreation": "2016-10-28 23:44:00.0",
        "server.agentId": "i-4447bc1b",
        "server.performance.maxDiskWritesPerSecondInKB": "0.0",
        "server.performance.avgDiskReadIOPS": "0.0",
        "server.performance.avgFreeRAMInKB": "1547210.133333333333",
        "server.performance.avgDiskReadsPerSecondInKB": "0.0",
        "server.performance.avgDiskWriteIOPS": "0.0",
        "server.performance.numNetworkCards": "2",
        "server.hypervisor": "xen",
        "server.networkInterfaceInfo.hasMoreValues": "false",
        "server.performance.avgNetworkWritesPerSecondInKB": "0.1380859375",
        "server.osName": "Linux - Amazon Linux AMI release 2015.03",
        "server.performance.totalRAMInKB": "1694732.0",
        "server.cpuType": "x64"
    },
    {
        "server.performance.maxCpuUsagePct": "100.0",
        "server.performance.maxDiskReadIOPS": "0.0",
        "server.performance.avgCpuUsagePct": "14.733333333333333338",
        "server.type": "EC2",
        "server.performance.maxNetworkReadsPerSecondInKB": "13.400390625",
        "server.hostName": "ip-172-31-42-208",
        "server.configurationId": "d-server-099385097ef9fbcbf",
        "server.tags.hasMoreValues": "false",
        "server.performance.minFreeRAMInKB": "1531104.0",
        "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
        "server.performance.maxDiskReadsPerSecondInKB": "0.0",
        "server.applications": "[]",
        "server.performance.numDisks": "1",
        "server.performance.numCpus": "1",
        "server.performance.numCores": "1",
        "server.performance.maxDiskWriteIOPS": "1.0",
        "server.performance.maxNetworkWritesPerSecondInKB": "12.271484375",
    }

```

```

        "server.performance.avgDiskWritesPerSecondInKB":
        "0.5333333333333334",
        "server.networkInterfaceInfo": "[{"name":"eth0",
        \ "macAddress\":"06:4A:79:60:75:61",\ "ipAddress\":"172.31.42.208",\ "netMask
        \":"255.255.240.0"}, {"name":"eth0",\ "macAddress\":"06:4A:79:60:75:61",
        \ "ipAddress\":"fe80::44a:79ff:fe60:7561"}, {"name":"lo",\ "macAddress\":"
        \ "00:00:00:00:00:00",\ "ipAddress\":"::1"}, {"name":"lo",\ "macAddress\":"
        \ "00:00:00:00:00:00",\ "ipAddress\":"127.0.0.1",\ "netMask\":"255.0.0.0"}]",
        "server.performance.avgNetworkReadsPerSecondInKB":
        "2.8720052083333334",
        "server.tags": "[]",
        "server.applications.hasMoreValues": "false",
        "server.timeOfCreation": "2016-10-28 23:44:30.0",
        "server.agentId": "i-c142b99e",
        "server.performance.maxDiskWritesPerSecondInKB": "4.0",
        "server.performance.avgDiskReadIOPS": "0.0",
        "server.performance.avgFreeRAMInKB": "1534946.4",
        "server.performance.avgDiskReadsPerSecondInKB": "0.0",
        "server.performance.avgDiskWriteIOPS": "0.13333333333333336",
        "server.performance.numNetworkCards": "2",
        "server.hypervisor": "xen",
        "server.networkInterfaceInfo.hasMoreValues": "false",
        "server.performance.avgNetworkWritesPerSecondInKB":
        "1.7977864583333332",
        "server.osName": "Linux - Amazon Linux AMI release 2015.03",
        "server.performance.totalRAMInKB": "1694732.0",
        "server.cpuType": "x64"
    }
]
}

```

Beschreiben Sie die ausgewählten Asset-Konfigurationen

Dieser Beispielbefehl beschreibt die Konfigurationen von zwei angegebenen Anwendungen. Die Aktion erkennt den Asset-Typ anhand der Konfigurations-ID. Pro Befehl ist nur ein Asset-Typ zulässig.

Befehl:

```
aws discovery describe-configurations --configuration-ids "d-
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"
```

Ausgabe:

```
{
  "configurations": [
    {
      "application.serverCount": "0",
      "application.name": "Application-12345",
      "application.lastModifiedTime": "2016-12-13 23:53:27.0",
      "application.description": "",
      "application.timeOfCreation": "2016-12-13 23:53:27.0",
      "application.configurationId": "d-application-0ac39bc0e4fad0e42"
    },
    {
      "application.serverCount": "0",
      "application.name": "Application-67890",
      "application.lastModifiedTime": "2016-12-13 23:53:33.0",
      "application.description": "",
      "application.timeOfCreation": "2016-12-13 23:53:33.0",
      "application.configurationId": "d-application-02444a45288013764"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeConfigurations](#) in der AWS CLI Befehlsreferenz.

list-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-configurations`.

AWS CLI

Um alle erkannten Server aufzulisten, die eine Reihe von Filterbedingungen erfüllen

Dieser Beispielbefehl listet die erkannten Server auf, die einem von zwei Hostnamenmustern entsprechen und auf denen Ubuntu nicht ausgeführt wird.

Befehl:

```
aws discovery list-configurations --configuration-type SERVER --filters
name="server.hostName",values="172-31-35","172-31-42",condition="CONTAINS"
name="server.osName",values="Ubuntu",condition="NOT_CONTAINS"
```

Ausgabe:


```
{
  "configurations": [
    {
      "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
      "server.type": "EC2",
      "server.hostName": "ip-172-31-42-208",
      "server.timeOfCreation": "2016-10-28 23:44:30.0",
      "server.configurationId": "d-server-099385097ef9fbcfb",
      "server.osName": "Linux - Amazon Linux AMI release 2015.03",
      "server.agentId": "i-c142b99e"
    },
    {
      "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
      "server.type": "EC2",
      "server.hostName": "ip-172-31-35-152",
      "server.timeOfCreation": "2016-10-28 23:44:00.0",
      "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
      "server.osName": "Linux - Amazon Linux AMI release 2015.03",
      "server.agentId": "i-4447bc1b"
    }
  ]
}
```

- API-Einheiten finden Sie [ListConfigurations](#) in der AWS CLI Befehlsreferenz.

AppRegistry Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AppRegistry.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-attribute-group

Das folgende Codebeispiel zeigt die Verwendung `associate-attribute-group`.

AWS CLI

Um eine Attributgruppe zuzuordnen

Im folgenden `associate-attribute-group` Beispiel wird eine bestimmte Attributgruppe in Ihrem AWS Konto einer bestimmten Anwendung in Ihrem AWS Konto zugeordnet.

```
aws servicecatalog-appregistry associate-attribute-group \  
  --application "ExampleApplication" \  
  --attribute-group "ExampleAttributeGroup"
```

Ausgabe:

```
{  
  "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
  "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1"  
}
```

Weitere Informationen finden Sie unter [Attributgruppen zuordnen und deren Zuordnung aufheben](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [AssociateAttributeGroup](#).AWS CLI

create-application

Das folgende Codebeispiel zeigt die Verwendung `create-application`.

AWS CLI

Um eine Anwendung zu erstellen

Das folgende `create-application` Beispiel erstellt eine neue Anwendung in Ihrem AWS Konto.

```
aws servicecatalog-appregistry create-application \  
  --name "ExampleApplication"
```

Ausgabe:

```
{  
  "application": {  
    "id": "0ars38r6btoohvpvd9gqrptt91",  
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
    "name": "ExampleApplication",  
    "creationTime": "2023-02-28T21:10:10.820000+00:00",  
    "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen von Anwendungen](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [CreateApplication](#) in der AWS CLI Befehlsreferenz.

create-attribute-group

Das folgende Codebeispiel zeigt die Verwendung `create-attribute-group`.

AWS CLI

Um eine Attributgruppe zu erstellen

Das folgende `create-attribute-group` Beispiel erstellt eine neue Attributgruppe in Ihrem AWS Konto.

```
aws servicecatalog-appregistry create-attribute-group \  
  --name "ExampleAttributeGroup" \  
  --attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'
```

Ausgabe:

```
{
```

```
"attributeGroup": {
  "id": "01sj5xdwhbw54kejwnt09fnpc1",
  "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpc1",
  "name": "ExampleAttributeGroup",
  "creationTime": "2023-02-28T20:38:01.389000+00:00",
  "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",
  "tags": {}
}
}
```

Weitere Informationen finden Sie unter [Erstellen von Attributgruppen](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [CreateAttributeGroup](#) unter AWS CLI Befehlsreferenz.

delete-application

Das folgende Codebeispiel zeigt die Verwendung `delete-application`.

AWS CLI

So löschen Sie eine Anwendung

Im folgenden `delete-application` Beispiel wird eine bestimmte Anwendung in Ihrem AWS Konto gelöscht.

```
aws servicecatalog-appregistry delete-application \
  --application "ExampleApplication3"
```

Ausgabe:

```
{
  "application": {
    "id": "055gw7aynr1i5mbv7kjwtzx5945",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwtzx5945",
    "name": "ExampleApplication3",
    "creationTime": "2023-02-28T22:06:28.228000+00:00",
    "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Löschen von Anwendungen](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteApplication](#) in der AWS CLI Befehlsreferenz.

delete-attribute-group

Das folgende Codebeispiel zeigt die Verwendung `delete-attribute-group`.

AWS CLI

Beispiel 8: Um eine Attributgruppe zu löschen

Das folgende `delete-attribute-group` Beispiel löscht eine bestimmte Attributgruppe in Ihrem AWS Konto.

```
aws servicecatalog-appregistry delete-attribute-group \
  --attribute-group "ExampleAttributeGroup3"
```

Ausgabe:

```
{
  "attributeGroup": {
    "id": "011ge6y3emyjijt8dw8jn6r0hv",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
    "name": "ExampleAttributeGroup3",
    "creationTime": "2023-02-28T22:05:35.224000+00:00",
    "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
  }
}
```

Weitere Informationen finden Sie unter [Löschen von Attributgruppen](#) im AWS Service Catalog AppRegistry Administrator Guide.

- API-Einheiten finden Sie [DeleteAttributeGroup](#) unter AWS CLI Befehlsreferenz.

get-application

Das folgende Codebeispiel zeigt die Verwendung `get-application`.

AWS CLI

Um eine Bewerbung zu erhalten

Im folgenden `get-application` Beispiel werden Metadateninformationen zu einer bestimmten Anwendung in Ihrem AWS Konto abgerufen.

```
aws servicecatalog-appregistry get-application \  
  --application "ExampleApplication"
```

Ausgabe:

```
{  
  "id": "0ars38r6btoohvpvd9gqrptt91",  
  "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
  "name": "ExampleApplication",  
  "creationTime": "2023-02-28T21:10:10.820000+00:00",  
  "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",  
  "associatedResourceCount": 0,  
  "tags": {  
    "aws:servicecatalog:applicationName": "ExampleApplication"  
  },  
  "integrations": {  
    "resourceGroup": {  
      "state": "CREATE_COMPLETE",  
      "arn": "arn:aws:resource-groups:us-west-2:813737243517:group/  
AWS_AppRegistry_Application-ExampleApplication"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Verwenden von Anwendungsdetails](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [GetApplication](#) unter AWS CLI Befehlsreferenz.

get-attribute-group

Das folgende Codebeispiel zeigt die Verwendung `get-attribute-group`.

AWS CLI

Um eine Attributgruppe abzurufen

Im folgenden `get-attribute-group` Beispiel wird eine bestimmte Attributgruppe in Ihrem AWS Konto abgerufen.

```
aws servicecatalog-appregistry get-attribute-group \  
  --attribute-group "ExampleAttributeGroup"
```

Ausgabe:

```
{  
  "id": "01sj5xdwhbw54kejwnt09fnpc1",  
  "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
  "name": "ExampleAttributeGroup",  
  "attributes": "{\"SomeKey1\":\"SomeValue1\", \"SomeKey2\":\"SomeValue2\"}",  
  "creationTime": "2023-02-28T20:38:01.389000+00:00",  
  "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",  
  "tags": {  
    "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwalten von Metadaten für Attributgruppen](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [GetAttributeGroup](#) unter AWS CLI Befehlsreferenz.

list-applications

Das folgende Codebeispiel zeigt die Verwendung `list-applications`.

AWS CLI

Um Anwendungen aufzulisten

Im folgenden `list-applications` Beispiel wird eine Liste aller Anwendungen in Ihrem AWS Konto abgerufen.

```
aws servicecatalog-appregistry list-applications
```

Ausgabe:

```
{
  "applications": [
    {
      "id": "03axw94pjfj3uan00tcgbrxnkw",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/03axw94pjfj3uan00tcgbrxnkw",
      "name": "ExampleApplication2",
      "creationTime": "2023-02-28T21:59:34.094000+00:00",
      "lastUpdateTime": "2023-02-28T21:59:34.094000+00:00"
    },
    {
      "id": "055gw7aynr1i5mbv7kjwzx5945",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
      "name": "ExampleApplication3",
      "creationTime": "2023-02-28T22:06:28.228000+00:00",
      "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
    },
    {
      "id": "0ars38r6btoohvpvd9gqrptt91",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt91",
      "name": "ExampleApplication",
      "description": "This is an example application",
      "creationTime": "2023-02-28T21:10:10.820000+00:00",
      "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Anzeigen von Anwendungsdetails](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [ListApplications](#) unter AWS CLI Befehlsreferenz.

list-associated-attribute-groups

Das folgende Codebeispiel zeigt die Verwendung `list-associated-attribute-groups`.

AWS CLI

Um zugehörige Attributgruppen aufzulisten

Im folgenden `list-associated-attribute-groups` Beispiel wird eine Liste aller Attributgruppen in Ihrem AWS Konto abgerufen, die einer bestimmten Anwendung in Ihrem AWS Konto zugeordnet sind.

```
aws servicecatalog-appregistry list-associated-attribute-groups \  
  --application "ExampleApplication"
```

Ausgabe:

```
{  
  "attributeGroups": [  
    "01sj5xdwhbw54kejwnt09fnpc1"  
  ]  
}
```

Weitere Informationen finden Sie unter [Attributgruppen zuordnen und deren Zuordnung aufheben](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [ListAssociatedAttributeGroups](#).AWS CLI

list-attribute-groups-for-application

Das folgende Codebeispiel zeigt die Verwendung `list-attribute-groups-for-application`.

AWS CLI

Um Attributgruppen für eine Anwendung aufzulisten

Im folgenden `list-attribute-groups-for-application` Beispiel werden die Details aller Attributgruppen in Ihrem AWS Konto aufgeführt, die einer bestimmten Anwendung in Ihrem AWS Konto zugeordnet sind.

```
aws servicecatalog-appregistry list-attribute-groups-for-application \  
  --application "ExampleApplication"
```

Ausgabe:

```
{
  "attributeGroupsDetails": [
    {
      "id": "01sj5xdwhbw54kejwnt09fnpc1",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpc1",
      "name": "ExampleAttributeGroup"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Attributgruppendetails anzeigen](#) im AWS Service Catalog AppRegistry Administrator Guide.

- API-Einheiten finden Sie [ListAttributeGroupsForApplication](#) unter AWS CLI Befehlsreferenz.

list-attribute-groups

Das folgende Codebeispiel zeigt die Verwendung `list-attribute-groups`.

AWS CLI

Um Attributgruppen aufzulisten

Im folgenden `list-attribute-groups` Beispiel wird eine Liste aller Attributgruppen in Ihrem AWS Konto abgerufen.

```
aws servicecatalog-appregistry list-attribute-groups
```

Ausgabe:

```
{
  "attributeGroups": [
    {
      "id": "011ge6y3emyjijt8dw8jn6r0hv",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
      "name": "ExampleAttributeGroup3",
      "creationTime": "2023-02-28T22:05:35.224000+00:00",
      "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
    },
    {
```

```

        "id": "01sj5xdwhbw54kejwnt09fnpc1",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpc1",
        "name": "ExampleAttributeGroup",
        "description": "This is an example attribute group",
        "creationTime": "2023-02-28T20:38:01.389000+00:00",
        "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00"
    },
    {
        "id": "03n1yffgq6d18vwrzxf0c70nm3",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/03n1yffgq6d18vwrzxf0c70nm3",
        "name": "ExampleAttributeGroup2",
        "creationTime": "2023-02-28T21:57:30.687000+00:00",
        "lastUpdateTime": "2023-02-28T21:57:30.687000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Attributgruppendetails anzeigen](#) im AWS Service Catalog AppRegistry Administrator Guide.

- API-Einheiten finden Sie [ListAttributeGroups](#) unter AWS CLI Befehlsreferenz.

update-application

Das folgende Codebeispiel zeigt die Verwendung `update-application`.

AWS CLI

Um eine Anwendung zu aktualisieren

Im folgenden `update-application` Beispiel wird eine bestimmte Anwendung in Ihrem AWS Konto aktualisiert, sodass sie eine Beschreibung enthält.

```

aws servicecatalog-appregistry update-application \
  --application "ExampleApplication" \
  --description "This is an example application"

```

Ausgabe:

```

{
  "application": {

```

```

    "id": "0ars38r6btoohvpvd9gqrptt91",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt91",
    "name": "ExampleApplication",
    "description": "This is an example application",
    "creationTime": "2023-02-28T21:10:10.820000+00:00",
    "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00",
    "tags": {
      "aws:servicecatalog:applicationName": "ExampleApplication"
    }
  }
}

```

Weitere Informationen finden Sie unter [Bearbeiten von Anwendungen](#) im AWS Service AppRegistry Catalog-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateApplication](#) in der AWS CLI Befehlsreferenz.

update-attribute-group

Das folgende Codebeispiel zeigt die Verwendung `update-attribute-group`.

AWS CLI

Um eine Attributgruppe zu aktualisieren

Im folgenden `update-attribute-group` Beispiel wird eine bestimmte Attributgruppe in Ihrem AWS Konto aktualisiert, sodass sie eine Beschreibung enthält.

```

aws servicecatalog-appregistry update-attribute-group \
  --attribute-group "ExampleAttributeGroup" \
  --description "This is an example attribute group"

```

Ausgabe:

```

{
  "attributeGroup": {
    "id": "01sj5xdwhbw54kejwnt09fnpc1",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpc1",
    "name": "ExampleAttributeGroup",
    "description": "This is an example attribute group",
    "creationTime": "2023-02-28T20:38:01.389000+00:00",

```

```
"lastUpdateTime": "2023-02-28T21:02:04.559000+00:00",
"tags": {
  "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"
}
}
```

Weitere Informationen finden Sie unter [Bearbeiten von Attributgruppen](#) im AWS Service Catalog AppRegistry Administrator Guide.

- API-Einheiten finden Sie [UpdateAttributeGroup](#) in der AWS CLI Befehlsreferenz.

Athena-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Athena Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-get-named-query

Das folgende Codebeispiel zeigt die Verwendung `batch-get-named-query`.

AWS CLI

Um Informationen zu mehr als einer Abfrage zurückzugeben

Im folgenden `batch-get-named-query` Beispiel werden Informationen zu den benannten Abfragen zurückgegeben, die den angegebenen Wert haben IDs.

```
aws athena batch-get-named-query \
```

```
--named-query-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

Ausgabe:

```
{
  "NamedQueries": [
    {
      "Name": "Flights Select Query",
      "Description": "Sample query to get the top 10 airports with the most
number of departures since 2000",
      "Database": "sampledb",
      "QueryString": "SELECT origin, count(*) AS total_departures\nFROM
\nflights_parquet\nWHERE year >= '2000'\nGROUP BY origin\nORDER BY total_departures
DESC\nLIMIT 10;",
      "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "WorkGroup": "primary"
    },
    {
      "Name": "Load flights table partitions",
      "Description": "Sample query to load flights table partitions using MSCK
REPAIR TABLE statement",
      "Database": "sampledb",
      "QueryString": "MSCK REPAIR TABLE flights_parquet;",
      "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "WorkGroup": "primary"
    },
    {
      "Name": "CloudFront Select Query",
      "Description": "Sample query to view requests per operating system
during a particular time frame",
      "Database": "sampledb",
      "QueryString": "SELECT os, COUNT(*) count FROM cloudfront_logs WHERE
date BETWEEN date '2014-07-05' AND date '2014-08-05' GROUP BY os;",
      "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "WorkGroup": "primary"
    }
  ],
  "UnprocessedNamedQueryIds": []
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [BatchGetNamedQuery AWS CLI-Befehlsreferenz](#).

batch-get-query-execution

Das folgende Codebeispiel zeigt die Verwendung `batch-get-query-execution`.

AWS CLI

Um Informationen über eine oder mehrere Abfrageausführungen zurückzugeben

Im folgenden `batch-get-query-execution` Beispiel werden Informationen zur Abfrageausführung für die Abfragen zurückgegeben, die die angegebene Abfrage IDs haben.

```
aws athena batch-get-query-execution \  
  --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-  
  cdef-EXAMPLE22222
```

Ausgabe:

```
{  
  "QueryExecutions": [  
    {  
      "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Query": "create database if not exists webdata",  
      "StatementType": "DDL",  
      "ResultConfiguration": {  
        "OutputLocation": "s3://awsdoc-example-bucket/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111.txt"  
      },  
      "QueryExecutionContext": {},  
      "Status": {  
        "State": "SUCCEEDED",  
        "SubmissionDateTime": 1593470720.592,  
        "CompletionDateTime": 1593470720.902  
      },  
      "Statistics": {  
        "EngineExecutionTimeInMillis": 232,  
        "DataScannedInBytes": 0,  
        "TotalExecutionTimeInMillis": 310,  
        "ResultConfiguration": {  
          "QueryQueueTimeInMillis": 50,  
          "ServiceProcessingTimeInMillis": 28
```

```

    },
    "WorkGroup": "AthenaAdmin"
  },
  {
    "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Query": "select date, location, browser, uri, status from
cloudfront_logs where method = 'GET' and status = 200 and location like 'SF0%'
limit 10",
    "StatementType": "DML",
    "ResultConfiguration": {
      "OutputLocation": "s3://awsdoc-example-bucket/a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222.csv"
    },
    "QueryExecutionContext": {
      "Database": "mydatabase",
      "Catalog": "awsdatacatalog"
    },
    "Status": {
      "State": "SUCCEEDED",
      "SubmissionDateTime": 1593469842.665,
      "CompletionDateTime": 1593469846.486
    },
    "Statistics": {
      "EngineExecutionTimeInMillis": 3600,
      "DataScannedInBytes": 203089,
      "TotalExecutionTimeInMillis": 3821,
      "QueryQueueTimeInMillis": 267,
      "QueryPlanningTimeInMillis": 1175
    },
    "WorkGroup": "AthenaAdmin"
  }
],
"UnprocessedQueryExecutionIds": []
}

```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [BatchGetQueryExecution AWS CLI](#) Befehlsreferenz.

create-data-catalog

Das folgende Codebeispiel zeigt die Verwendung `create-data-catalog`.

AWS CLI

Um einen Datenkatalog zu erstellen

Im folgenden `create-data-catalog` Beispiel wird der `dynamo_db_catalog` Datenkatalog erstellt.

```
aws athena create-data-catalog \  
  --name dynamo_db_catalog \  
  --type LAMBDA \  
  --description "DynamoDB Catalog" \  
  --parameters function=arn:aws:lambda:us-west-2:111122223333:function:dynamo_db_lambda
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um das Ergebnis zu sehen, verwenden Sie `aws athena get-data-catalog --name dynamo_db_catalog`.

Weitere Informationen finden Sie unter [Registrierung eines Katalogs: create-data-catalog](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateDataCatalog AWS CLI Befehlsreferenz](#).

create-named-query

Das folgende Codebeispiel zeigt die Verwendung `create-named-query`.

AWS CLI

Um eine benannte Abfrage zu erstellen

Im folgenden `create-named-query` Beispiel wird eine gespeicherte Abfrage in der AthenaAdmin Arbeitsgruppe erstellt, die die `flights_parquet` Tabelle nach Flügen von Seattle nach New York im Januar 2016 abfragt, deren Abflug und Ankunft jeweils um mehr als zehn Minuten verspätet waren. Da es sich bei den Flughafencodewerten in der Tabelle um Zeichenfolgen handelt, die doppelte Anführungszeichen enthalten (z. B. "SEA,"), werden sie durch umgekehrte Schrägstriche maskiert und von einfachen Anführungszeichen umgeben.

```
aws athena create-named-query \  
  --name "SEA to JFK delayed flights Jan 2016" \  
  --description "Both arrival and departure delayed more than 10 minutes." \  
  --database sampledb \  
  --query "SELECT * FROM flights_parquet WHERE arrival_delay > 10 OR departure_delay > 10"
```

```
--query-string "SELECT flightdate, carrier, flightnum, origin, dest,
depdelayminutes, arrdelayminutes FROM sampledb.flights_parquet WHERE yr = 2016 AND
month = 1 AND origin = '\"SEA\"' AND dest = '\"JFK\"' AND depdelayminutes > 10 AND
arrdelayminutes > 10" \
--work-group AthenaAdmin
```

Ausgabe:

```
{
  "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateNamedQuery AWS CLI](#) Befehlsreferenz.

create-work-group

Das folgende Codebeispiel zeigt die Verwendung `create-work-group`.

AWS CLI

Um eine Arbeitsgruppe zu erstellen

Im folgenden `create-work-group` Beispiel wird eine Arbeitsgruppe mit dem Namen `erstelltData_Analyst_Group`, die den Speicherort für die Ausgabe der Abfrageergebnisse enthält. `s3://awsdoc-example-bucket` Mit dem Befehl wird eine Arbeitsgruppe erstellt, die die Client-Konfigurationseinstellungen außer Kraft setzt. Dazu gehört auch der Speicherort für die Ausgabe der Abfrageergebnisse. Der Befehl aktiviert außerdem CloudWatch Metriken und fügt der Arbeitsgruppe drei Schlüssel-Wert-Tag-Paare hinzu, um sie von anderen Arbeitsgruppen zu unterscheiden. Beachten Sie, dass das `--configuration` Argument keine Leerzeichen vor den Kommas enthält, die seine Optionen trennen.

```
aws athena create-work-group \
  --name Data_Analyst_Group \
  --configuration ResultConfiguration={OutputLocation="s3://awsdoc-example-
bucket"},EnforceWorkGroupConfiguration="true",PublishCloudWatchMetricsEnabled="true"
\
  --description "Workgroup for data analysts" \
```

```
--tags Key=Division,Value=West Key=Location,Value=Seattle Key=Team,Value="Big Data"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um die Ergebnisse zu sehen, verwenden Sie `aws athena get-work-group --work-group Data_Analyst_Group`.

Weitere Informationen finden Sie unter [Managing Workgroups](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateWorkGroup](#) in der AWS CLI Befehlsreferenz.

delete-data-catalog

Das folgende Codebeispiel zeigt die Verwendung `delete-data-catalog`.

AWS CLI

Um einen Datenkatalog zu löschen

Im folgenden `delete-data-catalog` Beispiel wird der `UnusedDataCatalog` Datenkatalog gelöscht.

```
aws athena delete-data-catalog \  
  --name UnusedDataCatalog
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Katalogs: delete-data-catalog](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteDataCatalog AWS CLI](#) Befehlsreferenz.

delete-named-query

Das folgende Codebeispiel zeigt die Verwendung `delete-named-query`.

AWS CLI

Um eine benannte Abfrage zu löschen

Im folgenden `delete-named-query` Beispiel wird die benannte Abfrage mit der angegebenen ID gelöscht.

```
aws athena delete-named-query \  
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena Athena-Benutzerhandbuch](#).

- APIEinzelheiten finden Sie unter [DeleteNamedQuery AWS CLI](#) Befehlsreferenz.

delete-work-group

Das folgende Codebeispiel zeigt die Verwendung `delete-work-group`.

AWS CLI

Um eine Arbeitsgruppe zu löschen

Im folgenden `delete-work-group` Beispiel wird die `TeamB` Arbeitsgruppe gelöscht.

```
aws athena delete-work-group \  
  --work-group TeamB
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um den Löschvorgang zu bestätigen, verwenden Sie `aws athena list-work-groups`

Weitere Informationen finden Sie unter [Managing Workgroups](#) im Amazon Athena Athena-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteWorkGroup](#) in der AWS CLI Befehlsreferenz.

get-data-catalog

Das folgende Codebeispiel zeigt die Verwendung `get-data-catalog`.

AWS CLI

Um Informationen über einen Datenkatalog zurückzugeben

Das folgende `get-data-catalog` Beispiel gibt Informationen über den `dynamo_db_catalog` Datenkatalog zurück.

```
aws athena get-data-catalog \  
  --name dynamo_db_catalog
```

Ausgabe:

```
{  
  "DataCatalog": {  
    "Name": "dynamo_db_catalog",  
    "Description": "DynamoDB Catalog",  
    "Type": "LAMBDA",  
    "Parameters": {  
      "catalog": "dynamo_db_catalog",  
      "metadata-function": "arn:aws:lambda:us-  
west-2:111122223333:function:dynamo_db_lambda",  
      "record-function": "arn:aws:lambda:us-  
west-2:111122223333:function:dynamo_db_lambda"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Katalogdetails anzeigen: get-data-catalog](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetDataCatalog AWS CLI Befehlsreferenz](#).

get-database

Das folgende Codebeispiel zeigt die Verwendung `get-database`.

AWS CLI

Um Informationen über eine Datenbank in einem Datenkatalog zurückzugeben

Das folgende `get-database` Beispiel gibt Informationen über die `sampledb` Datenbank im `AwsDataCatalog` Datenkatalog zurück.

```
aws athena get-database \  
  --catalog-name AwsDataCatalog \  
  --database-name sampledb
```

Ausgabe:

```
{
  "Database": {
    "Name": "sampledb",
    "Description": "Sample database",
    "Parameters": {
      "CreatedBy": "Athena",
      "EXTERNAL": "TRUE"
    }
  }
}
```

Weitere Informationen finden Sie unter [Showing Database Details: get-database](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetDatabase AWS CLI](#) Befehlsreferenz.

get-named-query

Das folgende Codebeispiel zeigt die Verwendung `get-named-query`.

AWS CLI

Um eine benannte Abfrage zurückzugeben

Das folgende `get-named-query` Beispiel gibt Informationen über die Abfrage zurück, die die angegebene ID hat.

```
aws athena get-named-query \
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "NamedQuery": {
    "Name": "CloudFront Logs - SF0",
    "Description": "Shows successful GET request data for SF0",
    "Database": "default",
    "QueryString": "select date, location, browser, uri, status from
cloudfront_logs where method = 'GET' and status = 200 and location like 'SF0%'
limit 10",
    "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "WorkGroup": "AthenaAdmin"
  }
}
```

```
}
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetNamedQuery AWS CLI-Befehlsreferenz](#).

get-query-execution

Das folgende Codebeispiel zeigt die Verwendung `get-query-execution`.

AWS CLI

Um Informationen über die Ausführung einer Abfrage zurückzugeben

Das folgende `get-query-execution` Beispiel gibt Informationen über die Abfrage zurück, die die angegebene Abfrage-ID hat.

```
aws athena get-query-execution \
  --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "QueryExecution": {
    "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Query": "select date, location, browser, uri, status from cloudfront_logs
where method = 'GET
' and status = 200 and location like 'SF0%' limit 10",
    "StatementType": "DML",
    "ResultConfiguration": {
      "OutputLocation": "s3://awsdoc-example-bucket/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111.csv"
    },
    "QueryExecutionContext": {
      "Database": "mydatabase",
      "Catalog": "awsdatacatalog"
    },
    "Status": {
      "State": "SUCCEEDED",
      "SubmissionDateTime": 1593469842.665,
      "CompletionDateTime": 1593469846.486
    }
  }
}
```

```
    },
    "Statistics": {
      "EngineExecutionTimeInMillis": 3600,
      "DataScannedInBytes": 203089,
      "TotalExecutionTimeInMillis": 3821,
      "QueryQueueTimeInMillis": 267,
      "QueryPlanningTimeInMillis": 1175
    },
    "WorkGroup": "AthenaAdmin"
  }
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetQueryExecution AWS CLI](#) Befehlsreferenz.

get-query-results

Das folgende Codebeispiel zeigt die Verwendung `get-query-results`.

AWS CLI

Um die Ergebnisse einer Abfrage zurückzugeben

Das folgende `get-query-results` Beispiel gibt die Ergebnisse der Abfrage mit der angegebenen Abfrage-ID zurück.

```
aws athena get-query-results \
  --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "ResultSet": {
    "Rows": [
      {
        "Data": [
          {
            "VarCharValue": "date"
          },
          {
            "VarCharValue": "location"
          }
        ]
      }
    ]
  }
}
```



```
    },
    {
      "VarCharValue": "browser"
    },
    {
      "VarCharValue": "uri"
    },
    {
      "VarCharValue": "status"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Safari"
    },
    {
      "VarCharValue": "/test-image-2.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Opera"
    },
    {
      "VarCharValue": "/test-image-2.jpeg"
    }
  ]
}
```

```
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Firefox"
    },
    {
      "VarCharValue": "/test-image-3.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Lynx"
    },
    {
      "VarCharValue": "/test-image-3.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
}
```

```
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "IE"
    },
    {
      "VarCharValue": "/test-image-2.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Opera"
    },
    {
      "VarCharValue": "/test-image-1.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
```

```
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Chrome"
      },
      {
        "VarCharValue": "/test-image-3.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Firefox"
      },
      {
        "VarCharValue": "/test-image-2.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Chrome"
      },
      {

```

```
        "VarCharValue": "/test-image-3.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "IE"
      },
      {
        "VarCharValue": "/test-image-2.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  }
],
"ResultSetMetadata": {
  "ColumnInfo": [
    {
      "CatalogName": "hive",
      "SchemaName": "",
      "TableName": "",
      "Name": "date",
      "Label": "date",
      "Type": "date",
      "Precision": 0,
      "Scale": 0,
      "Nullable": "UNKNOWN",
      "CaseSensitive": false
    },
    {
      "CatalogName": "hive",
      "SchemaName": "",
```

```
    "TableName": "",
    "Name": "location",
    "Label": "location",
    "Type": "varchar",
    "Precision": 2147483647,
    "Data": [
        {
            "Scale": 0,
            "Nullable": "UNKNOWN",
            "CaseSensitive": true
        },
        {
            "CatalogName": "hive",
            "SchemaName": "",
            "TableName": "",
            "Name": "browser",
            "Label": "browser",
            "Type": "varchar",
            "Precision": 2147483647,
            "Scale": 0,
            "Nullable": "UNKNOWN",
            "CaseSensitive": true
        },
        {
            "CatalogName": "hive",
            "SchemaName": "",
            "TableName": "",
            "Name": "uri",
            "Label": "uri",
            "Type": "varchar",
            "Precision": 2147483647,
            "Scale": 0,
            "Nullable": "UNKNOWN",
            "CaseSensitive": true
        },
        {
            "CatalogName": "hive",
            "SchemaName": "",
            "TableName": "",
            "Name": "status",
            "Label": "status",
            "Type": "integer",
            "Precision": 10,
            "Scale": 0,
```

```

        "Nullable": "UNKNOWN",
        "CaseSensitive": false
      }
    ]
  },
  "UpdateCount": 0
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Abfrageergebnissen, Ausgabedateien und Abfrageverlauf](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [GetQueryResults](#) in der AWS CLI Befehlsreferenz.

get-table-metadata

Das folgende Codebeispiel zeigt die Verwendung `get-table-metadata`.

AWS CLI

Um Metadateninformationen zu einer Tabelle zurückzugeben

Im folgenden `get-table-metadata` Beispiel werden Metadateninformationen über die `counties` Tabelle, einschließlich Spaltennamen und deren Datentypen, aus der `sampledb` Datenbank des `AwsDataCatalog` Datenkatalogs zurückgegeben.

```

aws athena get-table-metadata \
  --catalog-name AwsDataCatalog \
  --database-name sampledb \
  --table-name counties

```

Ausgabe:

```

{
  "TableMetadata": {
    "Name": "counties",
    "CreateTime": 1593559968.0,
    "LastAccessTime": 0.0,
    "TableType": "EXTERNAL_TABLE",
    "Columns": [
      {

```

```

        "Name": "name",
        "Type": "string",
        "Comment": "from deserializer"
    },
    {
        "Name": "boundaryshape",
        "Type": "binary",
        "Comment": "from deserializer"
    },
    {
        "Name": "motto",
        "Type": "string",
        "Comment": "from deserializer"
    },
    {
        "Name": "population",
        "Type": "int",
        "Comment": "from deserializer"
    }
],
"PartitionKeys": [],
"Parameters": {
    "EXTERNAL": "TRUE",
    "inputformat": "com.esri.json.hadoop.EnclosedJsonInputFormat",
    "location": "s3://awsdoc-example-bucket/json",
    "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.serialization.format": "1",
    "serde.serialization.lib": "com.esri.hadoop.hive.serde.JsonSerde",
    "transient_lastDdlTime": "1593559968"
}
}
}

```

Weitere Informationen finden Sie unter [Tabellendetails anzeigen: get-table-metadata](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetTableMetadata AWS CLI Befehlsreferenz](#).

get-work-group

Das folgende Codebeispiel zeigt die Verwendung `get-work-group`.

AWS CLI

Um Informationen über eine Arbeitsgruppe zurückzugeben

Das folgende `get-work-group` Beispiel gibt Informationen über die `AthenaAdmin` Arbeitsgruppe zurück.

```
aws athena get-work-group \  
  --work-group AthenaAdmin
```

Ausgabe:

```
{  
  "WorkGroup": {  
    "Name": "AthenaAdmin",  
    "State": "ENABLED",  
    "Configuration": {  
      "ResultConfiguration": {  
        "OutputLocation": "s3://awsdoc-example-bucket/"  
      },  
      "EnforceWorkGroupConfiguration": false,  
      "PublishCloudWatchMetricsEnabled": true,  
      "RequesterPaysEnabled": false  
    },  
    "Description": "Workgroup for Athena administrators",  
    "CreationTime": 1573677174.105  
  }  
}
```

Weitere Informationen finden Sie unter [Managing Workgroups](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [GetWorkGroup](#) in der AWS CLI Befehlsreferenz.

list-data-catalogs

Das folgende Codebeispiel zeigt die Verwendung `list-data-catalogs`.

AWS CLI

Um die bei Athena registrierten Datenkataloge aufzulisten

Das folgende `list-data-catalogs` Beispiel listet die bei Athena registrierten Datenkataloge auf.

```
aws athena list-data-catalogs
```

Ausgabe:

```
{
  "DataCatalogsSummary": [
    {
      "CatalogName": "AwsDataCatalog",
      "Type": "GLUE"
    },
    {
      "CatalogName": "cw_logs_catalog",
      "Type": "LAMBDA"
    },
    {
      "CatalogName": "cw_metrics_catalog",
      "Type": "LAMBDA"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Auflisten registrierter Kataloge: list-data-catalogs](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListDataCatalogs AWS CLI](#) Befehlsreferenz.

list-databases

Das folgende Codebeispiel zeigt die Verwendung `list-databases`.

AWS CLI

Um die Datenbanken in einem Datenkatalog aufzulisten

Das folgende `list-databases` Beispiel listet die Datenbanken im `AwsDataCatalog` Datenkatalog auf.

```
aws athena list-databases \
  --catalog-name AwsDataCatalog
```

Ausgabe:

```
{
  "DatabaseList": [
    {
      "Name": "default"
    },
    {
      "Name": "mydatabase"
    },
    {
      "Name": "newdb"
    },
    {
      "Name": "sampledb",
      "Description": "Sample database",
      "Parameters": {
        "CreatedBy": "Athena",
        "EXTERNAL": "TRUE"
      }
    },
    {
      "Name": "webdata"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Datenbanken in einem Katalog auflisten: Listendatenbanken](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [ListDatabases](#) in AWS CLI der Befehlsreferenz.

list-named-queries

Das folgende Codebeispiel zeigt die Verwendung `list-named-queries`.

AWS CLI

Um die benannten Abfragen für eine Arbeitsgruppe aufzulisten

Das folgende `list-named-queries` Beispiel listet die benannten Abfragen für die `AthenaAdmin` Arbeitsgruppe auf.

```
aws athena list-named-queries \  
  --work-group AthenaAdmin
```

Ausgabe:

```
{  
  "NamedQueryIds": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
  ]  
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [ListNamedQueries AWS CLI](#) Befehlsreferenz.

list-query-executions

Das folgende Codebeispiel zeigt die Verwendung `list-query-executions`.

AWS CLI

Um die Abfrage IDs der Abfragen in einer bestimmten Arbeitsgruppe aufzulisten

Im folgenden `list-query-executions` Beispiel werden maximal zehn der Abfragen IDs in der `AthenaAdmin` Arbeitsgruppe aufgeführt.

```
aws athena list-query-executions \  
  --work-group AthenaAdmin \  
  --max-items 10
```

Ausgabe:

```
{  
  "QueryExecutionIds": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11110",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11114",  
  ]  
}
```

```

    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11115",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11116",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11117",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11118",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11119"
  ],
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Abfrageergebnissen, Ausgabedateien und Abfrageverlauf](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [ListQueryExecutions](#) in der AWS CLI Befehlsreferenz.

list-table-metadata

Das folgende Codebeispiel zeigt die Verwendung `list-table-metadata`.

AWS CLI

Um die Metadaten für Tabellen in der angegebenen Datenbank eines Datenkatalogs aufzulisten

Im folgenden `list-table-metadata` Beispiel werden Metadateninformationen für maximal zwei Tabellen in der `geography` Datenbank des `AwsDataCatalog` Datenkatalogs zurückgegeben.

```

aws athena list-table-metadata \
  --catalog-name AwsDataCatalog \
  --database-name geography \
  --max-items 2

```

Ausgabe:

```

{
  "TableMetadataList": [
    {
      "Name": "country_codes",
      "CreateTime": 1586553454.0,
      "TableType": "EXTERNAL_TABLE",
      "Columns": [
        {
          "Name": "country",
          "Type": "string",

```

```

        "Comment": "geo id"
    },
    {
        "Name": "alpha-2 code",
        "Type": "string",
        "Comment": "geo id2"
    },
    {
        "Name": "alpha-3 code",
        "Type": "string",
        "Comment": "state name"
    },
    {
        "Name": "numeric code",
        "Type": "bigint",
        "Comment": ""
    },
    {
        "Name": "latitude",
        "Type": "bigint",
        "Comment": "location (latitude)"
    },
    {
        "Name": "longitude",
        "Type": "bigint",
        "Comment": "location (longitude)"
    }
],
"Parameters": {
    "areColumnsQuoted": "false",
    "classification": "csv",
    "columnsOrdered": "true",
    "delimiter": ",",
    "has_encrypted_data": "false",
    "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
    "location": "s3://awsdoc-example-bucket/csv/countrycode",
    "outputformat":
"org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.field.delim": ",",
    "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "skip.header.line.count": "1",
    "typeOfData": "file"
}

```

```
  },
  {
    "Name": "county_populations",
    "CreateTime": 1586553446.0,
    "TableType": "EXTERNAL_TABLE",
    "Columns": [
      {
        "Name": "id",
        "Type": "string",
        "Comment": "geo id"
      },
      {
        "Name": "country",

        "Name": "id2",
        "Type": "string",
        "Comment": "geo id2"
      },
      {
        "Name": "county",
        "Type": "string",
        "Comment": "county name"
      },
      {
        "Name": "state",
        "Type": "string",
        "Comment": "state name"
      },
      {
        "Name": "population estimate 2018",
        "Type": "string",
        "Comment": ""
      }
    ],
    "Parameters": {
      "areColumnsQuoted": "false",
      "classification": "csv",
      "columnsOrdered": "true",
      "delimiter": ",",
      "has_encrypted_data": "false",
      "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
      "location": "s3://awsdoc-example-bucket/csv/CountyPopulation",
      "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
```

```

        "serde.param.field.delim": ",",
        "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "skip.header.line.count": "1",
        "typeOfData": "file"
    }
}
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

Weitere Informationen finden Sie unter [Metadaten für alle Tabellen in einer Datenbank anzeigen: list-table-metadata](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListTableMetadata AWS CLI Befehlsreferenz](#).

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Beispiel 1: Um die Tags für eine Arbeitsgruppe aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für die `Data_Analyst_Group` Arbeitsgruppe auf.

```

aws athena list-tags-for-resource \
  --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/
Data_Analyst_Group

```

Ausgabe:

```

{
  "Tags": [
    {
      "Key": "Division",
      "Value": "West"
    },
    {
      "Key": "Team",
      "Value": "Big Data"
    }
  ]
}

```



```

    },
    {
      "Key": "Location",
      "Value": "Seattle"
    }
  ]
}

```

Beispiel 2: Um die Tags für einen Datenkatalog aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für den `dynamo_db_catalog` Datenkatalog auf.

```

aws athena list-tags-for-resource \
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog

```

Ausgabe:

```

{
  "Tags": [
    {
      "Key": "Division",
      "Value": "Mountain"
    },
    {
      "Key": "Organization",
      "Value": "Retail"
    },
    {
      "Key": "Product_Line",
      "Value": "Shoes"
    },
    {
      "Key": "Location",
      "Value": "Denver"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Auflisten der Tags für eine Ressource: list-tags-for-resource](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListTagsForResource AWS CLI-Befehlsreferenz](#).

list-work-groups

Das folgende Codebeispiel zeigt die Verwendung `list-work-groups`.

AWS CLI

Um Arbeitsgruppen aufzulisten

Das folgende `list-work-groups` Beispiel listet die Arbeitsgruppen im aktuellen Konto auf.

```
aws athena list-work-groups
```

Ausgabe:

```
{
  "WorkGroups": [
    {
      "Name": "Data_Analyst_Group",
      "State": "ENABLED",
      "Description": "",
      "CreationTime": 1578006683.016
    },
    {
      "Name": "AthenaAdmin",
      "State": "ENABLED",
      "Description": "",
      "CreationTime": 1573677174.105
    },
    {
      "Name": "primary",
      "State": "ENABLED",
      "Description": "",
      "CreationTime": 1567465222.723
    }
  ]
}
```

Weitere Informationen finden Sie unter [Managing Workgroups](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [ListWorkGroups](#) in der AWS CLI Befehlsreferenz.

start-query-execution

Das folgende Codebeispiel zeigt die Verwendung `start-query-execution`.

AWS CLI

Beispiel 1: Um in einer Arbeitsgruppe eine Abfrage für die angegebene Tabelle in der angegebenen Datenbank und dem angegebenen Datenkatalog auszuführen

Im folgenden `start-query-execution` Beispiel wird die `AthenaAdmin` Arbeitsgruppe verwendet, um eine Abfrage für die `cloudfront_logs` Tabelle `cflogsdatabase` im `AwsDataCatalog` Datenkatalog auszuführen.

```
aws athena start-query-execution \  
  --query-string "select date, location, browser, uri, status from cloudfront_logs  
  where method = 'GET' and status = 200 and location like 'SF0%' limit 10" \  
  --work-group "AthenaAdmin" \  
  --query-execution-context Database=cflogsdatabase,Catalog=AwsDataCatalog
```

Ausgabe:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [SQL-Abfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

Beispiel 2: Um eine Abfrage auszuführen, die eine angegebene Arbeitsgruppe verwendet, um eine Datenbank im angegebenen Datenkatalog zu erstellen

Im folgenden `start-query-execution` Beispiel wird die `AthenaAdmin` Arbeitsgruppe verwendet, um die Datenbank `newdb` im Standarddatenkatalog zu erstellen. `AwsDataCatalog`

```
aws athena start-query-execution \  
  --query-string "create database if not exists newdb" \  
  --work-group "AthenaAdmin"
```

Ausgabe:

```
{
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

Beispiel 3: Um eine Abfrage auszuführen, die eine Ansicht für eine Tabelle in der angegebenen Datenbank und dem angegebenen Datenkatalog erstellt

Im folgenden `start-query-execution` Beispiel wird eine `SELECT` Anweisung für die `cloudfront_logs` Tabelle in der `verwendetcflogsdatabase`, um die Ansicht zu erstellen `cf10`.

```
aws athena start-query-execution \
  --query-string "CREATE OR REPLACE VIEW cf10 AS SELECT * FROM cloudfront_logs
  limit 10" \
  --query-execution-context Database=cflogsdatabase
```

Ausgabe:

```
{
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"
}
```

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [StartQueryExecution AWS CLI](#) Befehlsreferenz.

stop-query-execution

Das folgende Codebeispiel zeigt die Verwendung `stop-query-execution`.

AWS CLI

Um eine laufende Abfrage zu beenden

Im folgenden `stop-query-execution` Beispiel wird die Abfrage mit der angegebenen Abfrage-ID beendet.

```
aws athena stop-query-execution \  
  --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [SQLAbfragen mit Amazon Athena ausführen im Amazon Athena](#) Athena-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [StopQueryExecution AWS CLI](#) Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einer Ressource einen Tag hinzu

Im folgenden `tag-resource` Beispiel werden dem `dynamo_db_catalog` Datenkatalog drei Tags hinzugefügt.

```
aws athena tag-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog \  
  --  
  tags Key=Organization,Value=Retail Key=Division,Value=Mountain Key=Product_Line,Value=Shoes
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um das Ergebnis zu sehen, verwenden Sie `aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog`.

Weitere Informationen finden Sie unter [Hinzufügen von Tags zu einer Ressource: Tag-Ressource](#) im Amazon Athena Athena-Benutzerhandbuch.

- APIEinzelheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden die Focus Schlüssel `Specialization` und die zugehörigen Werte aus der `dynamo_db_catalog` Datenkatalogressource entfernt.

```
aws athena untag-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog \  
  --tag-keys Specialization Focus
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `list-tags-for-resource` Befehl, um die Ergebnisse anzuzeigen.

Weitere Informationen finden Sie unter [Entfernen von Tags aus einer Ressource: untag-resource](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [UntagResource](#).AWS CLI

update-data-catalog

Das folgende Codebeispiel zeigt die Verwendung `update-data-catalog`.

AWS CLI

Um einen Datenkatalog zu aktualisieren

Das folgende `update-data-catalog` Beispiel aktualisiert die Lambda-Funktion und die Beschreibung des `cw_logs_catalog` Datenkatalogs.

```
aws athena update-data-catalog \  
  --name cw_logs_catalog \  
  --type LAMBDA \  
  --description "New CloudWatch Logs Catalog" \  
  --function=arn:aws:lambda:us-west-2:111122223333:function:new_cw_logs_lambda
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um das Ergebnis zu sehen, verwenden Sie `aws athena get-data-catalog --name cw_logs_catalog`.

Weitere Informationen finden Sie unter [Aktualisieren eines Katalogs: update-data-catalog](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UpdateDataCatalog AWS CLI](#) Befehlsreferenz.

update-work-group

Das folgende Codebeispiel zeigt die Verwendung `update-work-group`.

AWS CLI

Um eine Arbeitsgruppe zu aktualisieren

Im folgenden `update-work-group` Beispiel wird die `Data_Analyst_Group` Arbeitsgruppe deaktiviert. Benutzer können in der deaktivierten Arbeitsgruppe keine Abfragen ausführen oder erstellen, können aber trotzdem Metriken, Kontrollen von Datenverwendungsbeschränkungen, Arbeitsgruppeneinstellungen, den Abfrageverlauf und gespeicherte Abfragen einsehen.

```
aws athena update-work-group \  
  --work-group Data_Analyst_Group \  
  --state DISABLED
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um die Statusänderung zu überprüfen, verwenden `aws athena get-work-group --work-group Data_Analyst_Group` und überprüfen Sie die `State` Eigenschaft in der Ausgabe.

Weitere Informationen finden Sie unter [Managing Workgroups](#) im Amazon Athena Athena-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateWorkGroup](#) in der AWS CLI Befehlsreferenz.

Auto Scaling Scaling-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Auto Scaling Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

attach-instances

Das folgende Codebeispiel zeigt die Verwendung `attach-instances`.

AWS CLI

So hängen Sie eine Instance an eine Auto Scaling Scaling-Gruppe an

In diesem Beispiel wird die angegebene Instance an die angegebene Auto Scaling Scaling-Gruppe angehängt.

```
aws autoscaling attach-instances \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [AttachInstances AWS CLI](#) Befehlsreferenz.

attach-load-balancer-target-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `attach-load-balancer-target-groups`.

AWS CLI

Um eine Zielgruppe einer Auto Scaling-Gruppe zuzuordnen

In diesem Beispiel wird die angegebene Zielgruppe der angegebenen Auto Scaling Scaling-Gruppe zugeordnet.

```
aws autoscaling attach-load-balancer-target-groups \  
  --auto-scaling-group-name my-asg \  
  --target-group-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```


Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Elastic Load Balancing und Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [AttachLoadBalancerTargetGroups](#) in der AWS CLI Befehlsreferenz.

attach-load-balancers

Das folgende Codebeispiel zeigt, wie man es benutzt `attach-load-balancers`.

AWS CLI

So fügen Sie einen Classic Load Balancer einer Auto Scaling Scaling-Gruppe hinzu

In diesem Beispiel wird der angegebene Classic Load Balancer der angegebenen Auto Scaling Scaling-Gruppe zugeordnet.

```
aws autoscaling attach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Elastic Load Balancing und Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [AttachLoadBalancers](#) in der AWS CLI Befehlsreferenz.

cancel-instance-refresh

Das folgende Codebeispiel zeigt, wie man es benutzt `cancel-instance-refresh`.

AWS CLI

Um eine Instanzaktualisierung abubrechen

Im folgenden `cancel-instance-refresh` Beispiel wird eine laufende Instanzaktualisierung für die angegebene Auto Scaling Scaling-Gruppe abgebrochen.

```
aws autoscaling cancel-instance-refresh \  
  --auto-scaling-group-name my-asg
```

```
--auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Weitere Informationen finden Sie unter [Abbrechen einer Instance-Aktualisierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [CancelInstanceRefresh](#) in der AWS CLI Befehlsreferenz.

complete-lifecycle-action

Das folgende Codebeispiel zeigt, wie man es benutzt `complete-lifecycle-action`.

AWS CLI

Um die Lebenszyklus-Aktion abzuschließen

In diesem Beispiel wird Amazon EC2 Auto Scaling darüber informiert, dass die angegebene Lebenszyklusaktion abgeschlossen ist, sodass der Start oder das Beenden der Instance abgeschlossen werden kann.

```
aws autoscaling complete-lifecycle-action \  
  --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-action-result CONTINUE \  
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [CompleteLifecycleAction](#) in der AWS CLI Befehlsreferenz.

create-auto-scaling-group

Das folgende Codebeispiel zeigt, wie man es benutzt `create-auto-scaling-group`.

AWS CLI

Beispiel 1: So erstellen Sie eine Auto Scaling Scaling-Gruppe

Im folgenden `create-auto-scaling-group` Beispiel wird eine Auto Scaling Scaling-Gruppe in Subnetzen in mehreren Availability Zones innerhalb einer Region erstellt. Die Instances werden mit der Standardversion der angegebenen Startvorlage gestartet. Beachten Sie, dass Standardwerte für die meisten anderen Einstellungen verwendet werden, z. B. für die Kündigungsrichtlinien und die Konfiguration der Integritätsprüfung.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12 \  
  --min-size 1 \  
  --max-size 5 \  
  --vpc-zone-identifler "subnet-5ea0c127, subnet-6194ea3b, subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auto Scaling Scaling-Gruppen](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: So fügen Sie einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer an

In diesem Beispiel wird die ARN Zielgruppe für einen Load Balancer angegeben, der den erwarteten Traffic unterstützt. Der Integritätsprüfungstyp gibt an, ELB dass, wenn Elastic Load Balancing eine Instance als fehlerhaft meldet, die Auto Scaling Scaling-Gruppe sie ersetzt. Der Befehl definiert auch eine Übergangszeit von `600` Sekunden für die Integritätsprüfung. Die Übergangszeit trägt dazu bei, eine vorzeitige Kündigung neu gestarteter Instances zu verhindern.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12 \  
  --target-group-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/943f017f100becff \  
  --health-check-type ELB \  
  --health-check-grace-period 600 \  
  --min-size 1 \  
  --max-size 5 \  
  --vpc-zone-identifler "subnet-5ea0c127, subnet-6194ea3b, subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Elastic Load Balancing und Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um eine Platzierungsgruppe anzugeben und die neueste Version der Startvorlage zu verwenden

In diesem Beispiel werden Instances in einer Platzierungsgruppe innerhalb einer einzelnen Availability Zone gestartet. Dies kann für Gruppen mit niedriger Latenz und HPC Workloads nützlich sein. In diesem Beispiel werden auch die Mindestgröße, die Maximalgröße und die gewünschte Kapazität der Gruppe angegeben.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \  
  --min-size 1 \  
  --max-size 5 \  
  --desired-capacity 3 \  
  --placement-group my-placement-group \  
  --vpc-zone-identifier "subnet-6194ea3b"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Platzierungsgruppen](#) im EC2Amazon-Benutzerhandbuch für Linux-Instances.

Beispiel 4: Um eine Auto Scaling Scaling-Gruppe für eine einzelne Instanz anzugeben und eine bestimmte Version der Startvorlage zu verwenden

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe erstellt, deren Mindest- und Höchstkapazität auf festgelegt sind, 1 um zu erzwingen, dass eine Instance ausgeführt wird. Der Befehl gibt auch Version 1 einer Startvorlage an, in der die ID einer vorhandenen Vorlage angegeben ENI ist. Wenn Sie eine Startvorlage verwenden, die ein vorhandenes ENI für eth0 angibt, müssen Sie eine Availability Zone für die Auto Scaling Scaling-Gruppe angeben, die der Netzwerkschnittstelle entspricht, ohne auch eine Subnetz-ID in der Anfrage anzugeben.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg-single-instance \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1' \  
  --min-size 1 \  
  --max-size 1
```

```
--max-size 1 \  
--availability-zones us-west-2a
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auto Scaling Scaling-Gruppen](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 5: Um eine andere Kündigungsrichtlinie anzugeben

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mithilfe einer Startkonfiguration erstellt und die Kündigungsrichtlinie so festgelegt, dass die ältesten Instances zuerst beendet werden. Der Befehl weist der Gruppe und ihren Instances außerdem ein Tag mit dem Schlüssel `Role` und dem Wert von `zuWebServer`.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-configuration-name my-lc \  
  --min-size 1 \  
  --max-size 5 \  
  --termination-policies "OldestInstance" \  
  --tags "ResourceId=my-asg,ResourceType=auto-scaling-  
group,Key=Role,Value=WebServer,PropagateAtLaunch=true" \  
  --vpc-zone-identifler "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon EC2 Auto Scaling Scaling-Kündigungsrichtlinien](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 6: So geben Sie einen Launch-Lifecycle-Hook an

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mit einem Lifecycle-Hook erstellt, der eine benutzerdefinierte Aktion beim Instance-Start unterstützt.

```
aws autoscaling create-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Inhalt der `config.json` Datei:

```
{
```

```

"AutoScalingGroupName": "my-asg",
"LaunchTemplate": {
  "LaunchTemplateId": "lt-1234567890abcde12"
},
"LifecycleHookSpecificationList": [{
  "LifecycleHookName": "my-launch-hook",
  "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
  "NotificationTargetARN": "arn:aws:sqs:us-west-2:123456789012:my-sqs-queue",
  "RoleARN": "arn:aws:iam::123456789012:role/my-notification-role",
  "NotificationMetadata": "SQS message metadata",
  "HeartbeatTimeout": 4800,
  "DefaultResult": "ABANDON"
}],
"MinSize": 1,
"MaxSize": 5,
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
"Tags": [{
  "ResourceType": "auto-scaling-group",
  "ResourceId": "my-asg",
  "PropagateAtLaunch": true,
  "Value": "test",
  "Key": "environment"
}]
}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 7: So geben Sie einen Termination-Lifecycle-Hook an

In diesem Beispiel wird eine Auto Scaling Scaling-Gruppe mit einem Lifecycle-Hook erstellt, der eine benutzerdefinierte Aktion beim Beenden der Instanz unterstützt.

```

aws autoscaling create-auto-scaling-group \
  --cli-input-json file://~/config.json

```

Inhalt von config.json:

```

{
  "AutoScalingGroupName": "my-asg",

```

```

"LaunchTemplate": {
  "LaunchTemplateId": "lt-1234567890abcde12"
},
"LifecycleHookSpecificationList": [{
  "LifecycleHookName": "my-termination-hook",
  "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
  "HeartbeatTimeout": 120,
  "DefaultResult": "CONTINUE"
}],
"MinSize": 1,
"MaxSize": 5,
"TargetGroupARNs": [
  "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067"
],
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 8: So geben Sie eine benutzerdefinierte Kündigungsrichtlinie an

In diesem Beispiel wird eine Auto Scaling-Gruppe erstellt, die eine benutzerdefinierte Richtlinie zur Beendigung von Lambda-Funktionen spezifiziert, die Amazon EC2 Auto Scaling mitteilt, welche Instances sicher im großen Maßstab beendet werden können.

```

aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name my-asg-single-instance \
  --launch-template LaunchTemplateName=my-template-for-auto-scaling \
  --min-size 1 \
  --max-size 5 \
  --termination-policies "arn:aws:lambda:us-
west-2:123456789012:function>HelloFunction:prod" \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen einer benutzerdefinierten Kündigungsrichtlinie mit Lambda](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateAutoScalingGroup](#) in der AWS CLI Befehlsreferenz.

create-launch-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `create-launch-configuration`.

AWS CLI

Beispiel 1: Um eine Startkonfiguration zu erstellen

In diesem Beispiel wird eine einfache Startkonfiguration erstellt.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen einer Startkonfiguration](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: So erstellen Sie eine Startkonfiguration mit einer Sicherheitsgruppe, einem key pair und einem Bootstrapping-Skript

In diesem Beispiel wird eine Startkonfiguration mit einer Sicherheitsgruppe, einem key pair und einem Bootstrapping-Skript erstellt, die in den Benutzerdaten enthalten sind.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --security-groups sg-eb2af88example \  
  --key-name my-key-pair \  
  --user-data file://myuserdata.txt
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen einer Startkonfiguration](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: So erstellen Sie eine Startkonfiguration mit einer IAM Rolle

In diesem Beispiel wird eine Startkonfiguration mit dem Instanzprofilnamen einer IAM Rolle erstellt.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --iam-instance-profile my-autoscaling-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [IAMRolle für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden, im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 4: Um eine Startkonfiguration mit aktivierter detaillierter Überwachung zu erstellen

In diesem Beispiel wird eine Startkonfiguration mit aktivierter EC2 detaillierter Überwachung erstellt, an die innerhalb von einer Minute EC2 Metriken gesendet werden. CloudWatch

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --instance-monitoring Enabled=true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfiguration der Überwachung für Auto Scaling Scaling-Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 5: Um eine Startkonfiguration zu erstellen, die Spot-Instances startet

In diesem Beispiel wird eine Startkonfiguration erstellt, die Spot-Instances als einzige Kaufoption verwendet.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --instance-type m5.large \  
  --instance-type m5.large
```

```
--spot-price "0.50"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Spot-Instances anfordern](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 6: So erstellen Sie eine Startkonfiguration mithilfe einer EC2 Instance

In diesem Beispiel wird eine Startkonfiguration erstellt, die auf den Attributen einer vorhandenen Instance basiert. Sie setzt die Platzierungs-Tenancy außer Kraft und legt fest, ob eine öffentliche IP-Adresse festgelegt wurde, indem die Optionen `--placement-tenancy` und `--no-associate-public-ip-address` eingeschlossen werden.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc-from-instance \  
  --instance-id i-0123a456700123456 \  
  --instance-type m5.large \  
  --no-associate-public-ip-address \  
  --placement-tenancy dedicated
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen einer Startkonfiguration mithilfe einer EC2 Instance](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 7: So erstellen Sie eine Startkonfiguration mit einer Blockgerätezuweisung für ein EBS Amazon-Volume

In diesem Beispiel wird eine Startkonfiguration mit einer Blockgerätezuweisung für ein EBS gp3 Amazon-Volume mit dem Gerätenamen `/dev/sdh` und einer Volumegröße von 20 erstellt.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --block-device-mappings '["DeviceName":"/dev/sdh", "Ebs":  
{"VolumeSize":20, "VolumeType":"gp3"}']
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [EBS](#) in der Amazon EC2 Auto Scaling API Scaling-Referenz.

Informationen zur Syntax für die Angabe von Parameterwerten im JSON Format -in Anführungszeichen finden Sie [unter Verwenden von Anführungszeichen bei Zeichenfolgen im AWS CLI Benutzerhandbuch](#) für die AWS Befehlszeilenschnittstelle.

Beispiel 8: So erstellen Sie eine Startkonfiguration mit einer Blockgerätezuordnung für ein Instance-Speicher-Volume

In diesem Beispiel wird eine Startkonfiguration mit ephemeral1 einem Instance-Speicher-Volume mit dem Gerätenamen erstellt/dev/sdc.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --block-device-mappings '[{"DeviceName":"/dev/sdc","VirtualName":"ephemeral1"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [BlockDeviceMapping](#) in der Amazon EC2 Auto Scaling API Scaling-Referenz.

Informationen zur Syntax für die Angabe von Parameterwerten im JSON Format -in Anführungszeichen finden Sie [unter Verwenden von Anführungszeichen bei Zeichenfolgen im AWS CLI Benutzerhandbuch](#) für die AWS Befehlszeilenschnittstelle.

Beispiel 9: Um eine Startkonfiguration zu erstellen und zu verhindern, dass ein Gerät beim Start eine Verbindung herstellt

In diesem Beispiel wird eine Startkonfiguration erstellt, die ein Blockgerät unterdrückt, das durch die Blockgerätezuweisung von AMI (z. B./dev/sdf) angegeben ist.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --block-device-mappings '[{"DeviceName":"/dev/sdf","NoDevice":""}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [BlockDeviceMapping](#) in der Amazon EC2 Auto Scaling API Scaling-Referenz.

Informationen zur Syntax für die Angabe von Parameterwerten im JSON Format -in Anführungszeichen finden Sie [unter Verwenden von Anführungszeichen bei Zeichenfolgen im AWS CLI](#) Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- APIEinzelheiten finden Sie unter Befehlsreferenz [CreateLaunchConfiguration](#).AWS CLI

create-or-update-tags

Das folgende Codebeispiel zeigt, wie man es benutzt `create-or-update-tags`.

AWS CLI

So erstellen oder aktualisieren Sie Tags für eine Auto Scaling Scaling-Gruppe

In diesem Beispiel werden der angegebenen Auto Scaling Scaling-Gruppe zwei Tags hinzugefügt.

```
aws autoscaling create-or-update-tags \  
  --tags ResourceId=my-asg,ResourceType=auto-scaling-  
group,Key=Role,Value=WebServer,PropagateAtLaunch=true ResourceId=my-  
asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research,PropagateAtLaunch=true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CreateOrUpdateTags](#) in der AWS CLI Befehlsreferenz.

delete-auto-scaling-group

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-auto-scaling-group`.

AWS CLI

Beispiel 1: Um die angegebene Auto Scaling Scaling-Gruppe zu löschen

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe gelöscht.

```
aws autoscaling delete-auto-scaling-group \  
  --group-name my-asg
```

```
--auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen Ihrer Auto Scaling Scaling-Infrastruktur](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: So erzwingen Sie das Löschen der angegebenen Auto Scaling Scaling-Gruppe

Verwenden Sie die `--force-delete` Option, um die Auto Scaling Scaling-Gruppe zu löschen, ohne darauf zu warten, dass die Instances in der Gruppe beendet werden.

```
aws autoscaling delete-auto-scaling-group \  
--auto-scaling-group-name my-asg \  
--force-delete
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen Ihrer Auto Scaling Scaling-Infrastruktur](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAutoScalingGroup](#) in der AWS CLI Befehlsreferenz.

delete-launch-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-launch-configuration`.

AWS CLI

Um eine Startkonfiguration zu löschen

In diesem Beispiel wird die angegebene Startkonfiguration gelöscht.

```
aws autoscaling delete-launch-configuration \  
--launch-configuration-name my-launch-config
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen Ihrer Auto Scaling Scaling-Infrastruktur](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteLaunchConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-lifecycle-hook

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-lifecycle-hook`.

AWS CLI

Um einen Lifecycle-Hook zu löschen

In diesem Beispiel wird der angegebene Lebenszyklus-Hook gelöscht.

```
aws autoscaling delete-lifecycle-hook \  
  --lifecycle-hook-name my-lifecycle-hook \  
  --auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteLifecycleHook](#) in der AWS CLI Befehlsreferenz.

delete-notification-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-notification-configuration`.

AWS CLI

Um eine Auto Scaling Scaling-Benachrichtigung zu löschen

In diesem Beispiel wird die angegebene Benachrichtigung aus der angegebenen Auto Scaling Scaling-Gruppe gelöscht.

```
aws autoscaling delete-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen der Benachrichtigungskonfiguration](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteNotificationConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-policy`.

AWS CLI

Um eine Skalierungsrichtlinie zu löschen

In diesem Beispiel wird die angegebene Skalierungsrichtlinie gelöscht.

```
aws autoscaling delete-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name alb1000-target-tracking-scaling-policy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePolicy](#) in der AWS CLI Befehlsreferenz.

delete-scheduled-action

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-scheduled-action`.

AWS CLI

Um eine geplante Aktion aus einer Auto Scaling Scaling-Gruppe zu löschen

In diesem Beispiel wird die angegebene geplante Aktion aus der angegebenen Auto Scaling Scaling-Gruppe gelöscht.

```
aws autoscaling delete-scheduled-action \  
  --auto-scaling-group-name my-asg \  
  --scheduled-action-name my-scheduled-action
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteScheduledAction AWS CLI](#) Befehlsreferenz.

delete-tags

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-tags`.

AWS CLI

Um ein Tag aus einer Auto Scaling Scaling-Gruppe zu löschen

In diesem Beispiel wird das angegebene Tag aus der angegebenen Auto Scaling Scaling-Gruppe gelöscht.

```
aws autoscaling delete-tags \  
  --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteTags](#) in der AWS CLI Befehlsreferenz.

delete-warm-pool

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-warm-pool`.

AWS CLI

Beispiel 1: Um einen warmen Pool zu löschen

Im folgenden Beispiel wird der warme Pool für die angegebene Auto Scaling Scaling-Gruppe gelöscht.

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Warm Pools for Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: So erzwingen Sie das Löschen eines warmen Pools

Verwenden Sie die `--force-delete` Option, um den warmen Pool zu löschen, ohne darauf zu warten, dass seine Instanzen beendet werden.


```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Warm Pools for Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteWarmPool](#) in der AWS CLI Befehlsreferenz.

describe-account-limits

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-account-limits`.

AWS CLI

Um Ihre Amazon EC2 Auto Scaling Scaling-Kontolimits zu beschreiben

In diesem Beispiel werden die Amazon EC2 Auto Scaling Scaling-Limits für Ihr AWS Konto beschrieben.

```
aws autoscaling describe-account-limits
```

Ausgabe:

```
{  
  "NumberOfLaunchConfigurations": 5,  
  "MaxNumberOfLaunchConfigurations": 100,  
  "NumberOfAutoScalingGroups": 3,  
  "MaxNumberOfAutoScalingGroups": 20  
}
```

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling-Servicekontingente](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAccountLimits](#) in der AWS CLI Befehlsreferenz.

describe-adjustment-types

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-adjustment-types`.

AWS CLI

Um die verfügbaren Skalierungsanpassungstypen zu beschreiben

In diesem Beispiel werden die verfügbaren Anpassungstypen beschrieben.

```
aws autoscaling describe-adjustment-types
```

Ausgabe:

```
{
  "AdjustmentTypes": [
    {
      "AdjustmentType": "ChangeInCapacity"
    },
    {
      "AdjustmentType": "ExactCapacity"
    },
    {
      "AdjustmentType": "PercentChangeInCapacity"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Skalierungsanpassungstypen](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAdjustmentTypes](#) in der AWS CLI Befehlsreferenz.

describe-auto-scaling-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-auto-scaling-groups`.

AWS CLI

Beispiel 1: Um die angegebene Auto Scaling Scaling-Gruppe zu beschreiben

Dieses Beispiel beschreibt die angegebene Auto Scaling Scaling-Gruppe.

```
aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "my-asg",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-1234567890abcde12"
      },
      "MinSize": 0,
      "MaxSize": 1,
      "DesiredCapacity": 1,
      "DefaultCooldown": 300,
      "AvailabilityZones": [
        "us-west-2a",
        "us-west-2b",
        "us-west-2c"
      ],
      "LoadBalancerNames": [],
      "TargetGroupARNs": [],
      "HealthCheckType": "EC2",
      "HealthCheckGracePeriod": 0,
      "Instances": [
        {
          "InstanceId": "i-06905f55584de02da",
          "InstanceType": "t2.micro",
          "AvailabilityZone": "us-west-2a",
          "HealthStatus": "Healthy",
          "LifecycleState": "InService",
          "ProtectedFromScaleIn": false,
          "LaunchTemplate": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1",
            "LaunchTemplateId": "lt-1234567890abcde12"
          }
        }
      ],
      "CreatedTime": "2023-10-28T02:39:22.152Z",
      "SuspendedProcesses": [],
    }
  ]
}
```

```

    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "EnabledMetrics": [],
    "Tags": [],
    "TerminationPolicies": [
      "Default"
    ],
    "NewInstancesProtectedFromScaleIn": false,
    "ServiceLinkedRoleARN": "arn",
    "TrafficSources": []
  }
]
}

```

Beispiel 2: Um die ersten 100 angegebenen Auto Scaling Scaling-Gruppe zu beschreiben

In diesem Beispiel werden die angegebenen Auto Scaling Scaling-Gruppen beschrieben. Es ermöglicht Ihnen, bis zu 100 Gruppennamen anzugeben.

```

aws autoscaling describe-auto-scaling-groups \
  --max-items 100 \
  --auto-scaling-group-name "group1" "group2" "group3" "group4"

```

In Beispiel 1 finden Sie eine Beispielausgabe.

Beispiel 3: Um eine Auto Scaling Scaling-Gruppe in der angegebenen Region zu beschreiben

Dieses Beispiel beschreibt die Auto Scaling Scaling-Gruppen in der angegebenen Region, bis zu einem Maximum von 75 Gruppen.

```

aws autoscaling describe-auto-scaling-groups \
  --max-items 75 \
  --region us-east-1

```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 4: Um die angegebene Anzahl von Auto Scaling Scaling-Gruppen zu beschreiben

Um eine bestimmte Anzahl von Auto Scaling Scaling-Gruppen zurückzugeben, verwenden Sie die `--max-items` Option.

```

aws autoscaling describe-auto-scaling-groups \

```

```
--max-items 1
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Wenn die Ausgabe ein NextToken Feld enthält, gibt es mehr Gruppen. Um die zusätzlichen Gruppen abzurufen, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-auto-scaling-groups \  
--starting-token Z3M3LMPEXAMPLE
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 5: Um Auto Scaling Scaling-Gruppen zu beschreiben, die Startkonfigurationen verwenden

In diesem Beispiel wird die `--query` Option verwendet, um Auto Scaling Scaling-Gruppen zu beschreiben, die Startkonfigurationen verwenden.

```
aws autoscaling describe-auto-scaling-groups \  
--query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Ausgabe:

```
[  
  {  
    "AutoScalingGroupName": "my-asg",  
    "AutoScalingGroupARN": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:930d940e-891e-4781-a11a-7b0acd480f03:autoScalingGroupName/my-asg",  
    "LaunchConfigurationName": "my-lc",  
    "MinSize": 0,  
    "MaxSize": 1,  
    "DesiredCapacity": 1,  
    "DefaultCooldown": 300,  
    "AvailabilityZones": [  
      "us-west-2a",  
      "us-west-2b",  
      "us-west-2c"  
    ],  
    "LoadBalancerNames": [],  
    "TargetGroupARNs": [],  
    "HealthCheckType": "EC2",
```

```
"HealthCheckGracePeriod": 0,
"Instances": [
  {
    "InstanceId": "i-088c57934a6449037",
    "InstanceType": "t2.micro",
    "AvailabilityZone": "us-west-2c",
    "HealthStatus": "Healthy",
    "LifecycleState": "InService",
    "LaunchConfigurationName": "my-lc",
    "ProtectedFromScaleIn": false
  }
],
"CreatedTime": "2023-10-28T02:39:22.152Z",
"SuspendedProcesses": [],
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
"EnabledMetrics": [],
"Tags": [],
"TerminationPolicies": [
  "Default"
],
"NewInstancesProtectedFromScaleIn": false,
"ServiceLinkedRoleARN": "arn",
"TrafficSources": []
}
]
```

Weitere Informationen finden Sie unter [AWS CLIAusgabe filtern](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DescribeAutoScalingGroups](#) in der AWS CLI Befehlsreferenz.

describe-auto-scaling-instances

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-auto-scaling-instances`.

AWS CLI

Beispiel 1: Um eine oder mehrere Instanzen zu beschreiben

Dieses Beispiel beschreibt die angegebene Instanz.

```
aws autoscaling describe-auto-scaling-instances \
  --instance-ids i-06905f55584de02da
```

Ausgabe:

```
{
  "AutoScalingInstances": [
    {
      "InstanceId": "i-06905f55584de02da",
      "InstanceType": "t2.micro",
      "AutoScalingGroupName": "my-asg",
      "AvailabilityZone": "us-west-2b",
      "LifecycleState": "InService",
      "HealthStatus": "HEALTHY",
      "ProtectedFromScaleIn": false,
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-1234567890abcde12",
        "LaunchTemplateName": "my-launch-template",
        "Version": "1"
      }
    }
  ]
}
```

Beispiel 2: Um eine oder mehrere Instanzen zu beschreiben

In diesem Beispiel wird mithilfe der `--max-items` Option angegeben, wie viele Instanzen mit diesem Aufruf zurückgegeben werden sollen.

```
aws autoscaling describe-auto-scaling-instances \
  --max-items 1
```

Wenn die Ausgabe ein `NextToken` Feld enthält, gibt es mehr Instanzen. Um die zusätzlichen Instanzen abzurufen, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-auto-scaling-instances \
  --starting-token Z3M3LMPEXAMPLE
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 3: Um Instances zu beschreiben, die Startkonfigurationen verwenden

In diesem Beispiel wird die `--query` Option verwendet, um Instances zu beschreiben, die Startkonfigurationen verwenden.

```
aws autoscaling describe-auto-scaling-instances \
  --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

Ausgabe:

```
[
  {
    "InstanceId": "i-088c57934a6449037",
    "InstanceType": "t2.micro",
    "AutoScalingGroupName": "my-asg",
    "AvailabilityZone": "us-west-2c",
    "LifecycleState": "InService",
    "HealthStatus": "HEALTHY",
    "LaunchConfigurationName": "my-lc",
    "ProtectedFromScaleIn": false
  }
]
```

Weitere Informationen finden Sie unter [AWS CLIAusgabe filtern](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DescribeAutoScalingInstances](#) in der AWS CLI Befehlsreferenz.

describe-auto-scaling-notification-types

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-auto-scaling-notification-types`.

AWS CLI

Um die verfügbaren Benachrichtigungstypen zu beschreiben

In diesem Beispiel werden die verfügbaren Benachrichtigungstypen beschrieben.

```
aws autoscaling describe-auto-scaling-notification-types
```

Ausgabe:

```
{
  "AutoScalingNotificationTypes": [
    "autoscaling:EC2_INSTANCE_LAUNCH",
```



```

    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
    "autoscaling:EC2_INSTANCE_TERMINATE",
    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
    "autoscaling:TEST_NOTIFICATION"
  ]
}

```

Weitere Informationen finden Sie unter [SNS Amazon-Benachrichtigungen erhalten, wenn Ihre Auto Scaling-Gruppe skaliert](#) wird im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API Einzelheiten finden Sie [DescribeAutoScalingNotificationTypes](#) in der AWS CLI Befehlsreferenz.

describe-instance-refreshes

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-instance-refreshes`.

AWS CLI

Um Instanzaktualisierungen zu beschreiben

Das folgende `describe-instance-refreshes` Beispiel gibt eine Beschreibung aller Instanzaktualisierungsanforderungen für die angegebene Auto Scaling Scaling-Gruppe zurück, einschließlich der Statusmeldung und (falls verfügbar) des Statusgrundes.

```

aws autoscaling describe-instance-refreshes \
  --auto-scaling-group-name my-asg

```

Ausgabe:

```

{
  "InstanceRefreshes": [
    {
      "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",
      "AutoScalingGroupName": "my-asg",
      "Status": "InProgress",
      "StatusReason": "Waiting for instances to warm up before continuing. For example: 0e69cc3f05f825f4f is warming up.",
      "EndTime": "2023-03-23T16:42:55Z",
      "PercentageComplete": 0,
      "InstancesToUpdate": 0,
      "Preferences": {

```

```

        "MinHealthyPercentage": 100,
        "InstanceWarmup": 300,
        "CheckpointPercentages": [
            50
        ],
        "CheckpointDelay": 3600,
        "SkipMatching": false,
        "AutoRollback": true,
        "ScaleInProtectedInstances": "Ignore",
        "StandbyInstances": "Ignore"
    }
},
{
    "InstanceRefreshId": "dd7728d0-5bc4-4575-96a3-1b2c52bf8bb1",
    "AutoScalingGroupName": "my-asg",
    "Status": "Successful",
    "EndTime": "2022-06-02T16:53:37Z",
    "PercentageComplete": 100,
    "InstancesToUpdate": 0,
    "Preferences": {
        "MinHealthyPercentage": 90,
        "InstanceWarmup": 300,
        "SkipMatching": true,
        "AutoRollback": true,
        "ScaleInProtectedInstances": "Ignore",
        "StandbyInstances": "Ignore"
    }
}
]
}

```

Weitere Informationen finden [Sie unter Überprüfen des Status einer Instance-Aktualisierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstanceRefreshes](#) unter AWS CLI Befehlsreferenz.

describe-launch-configurations

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-launch-configurations`.

AWS CLI

Beispiel 1: Um die angegebene Startkonfiguration zu beschreiben

Dieses Beispiel beschreibt die angegebene Startkonfiguration.

```
aws autoscaling describe-launch-configurations \
  --launch-configuration-names my-launch-config
```

Ausgabe:

```
{
  "LaunchConfigurations": [
    {
      "LaunchConfigurationName": "my-launch-config",
      "LaunchConfigurationARN": "arn:aws:autoscaling:us-
west-2:123456789012:launchConfiguration:98d3b196-4cf9-4e88-8ca1-8547c24ced8b:launchConfigura
my-launch-config",
      "ImageId": "ami-0528a5175983e7f28",
      "KeyName": "my-key-pair-uswest2",
      "SecurityGroups": [
        "sg-05eaec502fcdadc2e"
      ],
      "ClassicLinkVPCSecurityGroups": [],
      "UserData": "",
      "InstanceType": "t2.micro",
      "KernelId": "",
      "RamdiskId": "",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/xvda",
          "Ebs": {
            "SnapshotId": "snap-06c1606ba5ca274b1",
            "VolumeSize": 8,
            "VolumeType": "gp2",
            "DeleteOnTermination": true,
            "Encrypted": false
          }
        }
      ],
      "InstanceMonitoring": {
        "Enabled": true
      },
      "CreatedTime": "2020-10-28T02:39:22.321Z",
      "EbsOptimized": false,
      "AssociatePublicIpAddress": true,
      "MetadataOptions": {
```

```
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "disabled"
      }
    ]
  }
```

Beispiel 2: Um eine bestimmte Anzahl von Startkonfigurationen zu beschreiben

Verwenden Sie die `--max-items` Option, um eine bestimmte Anzahl von Startkonfigurationen zurückzugeben.

```
aws autoscaling describe-launch-configurations \
  --max-items 1
```

Wenn die Ausgabe ein `NextToken` Feld enthält, gibt es mehr Startkonfigurationen. Um die zusätzlichen Startkonfigurationen abzurufen, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-launch-configurations \
  --starting-token Z3M3LMPEXAMPLE
```

- API-Einheiten finden Sie [DescribeLaunchConfigurations](#) in der AWS CLI Befehlsreferenz.

describe-lifecycle-hook-types

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-lifecycle-hook-types`.

AWS CLI

Um die verfügbaren Lifecycle-Hook-Typen zu beschreiben

In diesem Beispiel werden die verfügbaren Lifecycle-Hook-Typen beschrieben.

```
aws autoscaling describe-lifecycle-hook-types
```

Ausgabe:

```
{
  "LifecycleHookTypes": [
```

```

    "autoscaling:EC2_INSTANCE_LAUNCHING",
    "autoscaling:EC2_INSTANCE_TERMINATING"
  ]
}

```

- API-Einheiten finden Sie [DescribeLifecycleHookTypes](#) in der AWS CLI Befehlsreferenz.

describe-lifecycle-hooks

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-lifecycle-hooks`.

AWS CLI

Um Ihre Lifecycle-Hooks zu beschreiben

In diesem Beispiel werden die Lifecycle-Hooks für die angegebene Auto Scaling Group beschrieben.

```

aws autoscaling describe-lifecycle-hooks \
  --auto-scaling-group-name my-asg

```

Ausgabe:

```

{
  "LifecycleHooks": [
    {
      "GlobalTimeout": 3000,
      "HeartbeatTimeout": 30,
      "AutoScalingGroupName": "my-asg",
      "LifecycleHookName": "my-launch-hook",
      "DefaultResult": "ABANDON",
      "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING"
    },
    {
      "GlobalTimeout": 6000,
      "HeartbeatTimeout": 60,
      "AutoScalingGroupName": "my-asg",
      "LifecycleHookName": "my-termination-hook",
      "DefaultResult": "CONTINUE",
      "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING"
    }
  ]
}

```

```
}
```

- API-Einheiten finden Sie [DescribeLifecycleHooks](#) in der AWS CLI Befehlsreferenz.

describe-load-balancer-target-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-load-balancer-target-groups`.

AWS CLI

Um die Load Balancer-Zielgruppen für eine Auto Scaling Scaling-Gruppe zu beschreiben

In diesem Beispiel werden die Load Balancer-Zielgruppen beschrieben, die der angegebenen Auto Scaling Scaling-Gruppe zugeordnet sind.

```
aws autoscaling describe-load-balancer-target-groups \
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{
  "LoadBalancerTargetGroups": [
    {
      "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
      "State": "Added"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeLoadBalancerTargetGroups](#) in der AWS CLI Befehlsreferenz.

describe-load-balancers

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-load-balancers`.

AWS CLI

Um die Classic Load Balancers für eine Auto Scaling Scaling-Gruppe zu beschreiben

In diesem Beispiel werden die Classic Load Balancers für die angegebene Auto Scaling Scaling-Gruppe beschrieben.

```
aws autoscaling describe-load-balancers \  
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "LoadBalancers": [  
    {  
      "State": "Added",  
      "LoadBalancerName": "my-load-balancer"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribeLoadBalancers](#) in der AWS CLI Befehlsreferenz.

describe-metric-collection-types

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-metric-collection-types`.

AWS CLI

Um die verfügbaren Arten der Erfassung von Metriken zu beschreiben

In diesem Beispiel werden die verfügbaren Arten der Erfassung von Metriken beschrieben.

```
aws autoscaling describe-metric-collection-types
```

Ausgabe:

```
{  
  "Metrics": [  
    {  
      "Metric": "GroupMinSize"  
    },  
    {  
      "Metric": "GroupMaxSize"  
    },  
    {
```

```
    "Metric": "GroupDesiredCapacity"
  },
  {
    "Metric": "GroupInServiceInstances"
  },
  {
    "Metric": "GroupInServiceCapacity"
  },
  {
    "Metric": "GroupPendingInstances"
  },
  {
    "Metric": "GroupPendingCapacity"
  },
  {
    "Metric": "GroupTerminatingInstances"
  },
  {
    "Metric": "GroupTerminatingCapacity"
  },
  {
    "Metric": "GroupStandbyInstances"
  },
  {
    "Metric": "GroupStandbyCapacity"
  },
  {
    "Metric": "GroupTotalInstances"
  },
  {
    "Metric": "GroupTotalCapacity"
  }
],
"Granularities": [
  {
    "Granularity": "1Minute"
  }
]
}
```

Weitere Informationen finden Sie unter [Auto Scaling Scaling-Gruppenmetriken](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMetricCollectionTypes](#) in der AWS CLI Befehlsreferenz.

describe-notification-configurations

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-notification-configurations`.

AWS CLI

Beispiel 1: Um die Benachrichtigungskonfigurationen einer bestimmten Gruppe zu beschreiben

In diesem Beispiel werden die Benachrichtigungskonfigurationen für die angegebene Auto Scaling Scaling-Gruppe beschrieben.

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{
  "NotificationConfigurations": [
    {
      "AutoScalingGroupName": "my-asg",
      "NotificationType": "autoscaling:TEST_NOTIFICATION",
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
    },
    {
      "AutoScalingGroupName": "my-asg",
      "NotificationType": "autoscaling:TEST_NOTIFICATION",
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
    }
  ]
}
```

Weitere Informationen finden Sie unter [SNS Amazon-Benachrichtigungen erhalten, wenn Ihre Auto Scaling-Gruppe skaliert](#) wird im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 1: Um eine bestimmte Anzahl von Benachrichtigungskonfigurationen zu beschreiben

Verwenden Sie den `max-items` Parameter, um eine bestimmte Anzahl von Benachrichtigungskonfigurationen zurückzugeben.

```
aws autoscaling describe-notification-configurations \
```

```
--auto-scaling-group-name my-auto-scaling-group \  
--max-items 1
```

Ausgabe:

```
{  
  "NotificationConfigurations": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "NotificationType": "autoscaling:TEST_NOTIFICATION",  
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"  
    },  
    {  
      "AutoScalingGroupName": "my-asg",  
      "NotificationType": "autoscaling:TEST_NOTIFICATION",  
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"  
    }  
  ]  
}
```

Wenn die Ausgabe ein `NextToken` Feld enthält, gibt es mehr Benachrichtigungskonfigurationen. Um die zusätzlichen Benachrichtigungskonfigurationen abzurufen, verwenden Sie den Wert dieses Felds zusammen mit dem `starting-token` Parameter in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-notification-configurations \  
  --auto-scaling-group-name my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

Weitere Informationen finden Sie unter [SNSAmazon-Benachrichtigungen erhalten, wenn Ihre Auto Scaling-Gruppe skaliert](#) wird im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeNotificationConfigurations](#) in der AWS CLI Befehlsreferenz.

describe-policies

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-policies`.

AWS CLI

Beispiel 1: Um die Skalierungsrichtlinien einer bestimmten Gruppe zu beschreiben

In diesem Beispiel werden die Skalierungsrichtlinien für die angegebene Auto Scaling Scaling-Gruppe beschrieben.

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "ScalingPolicies": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "PolicyName": "alb1000-target-tracking-scaling-policy",  
      "PolicyARN": "arn:aws:autoscaling:us-  
west-2:123456789012:scalingPolicy:3065d9c8-9969-4bec-  
bb6a-3fbe5550fde6:autoScalingGroupName/my-asg:policyName/alb1000-target-tracking-  
scaling-policy",  
      "PolicyType": "TargetTrackingScaling",  
      "StepAdjustments": [],  
      "Alarms": [  
        {  
          "AlarmName": "TargetTracking-my-asg-  
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196",  
          "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:123456789012:alarm:TargetTracking-my-asg-  
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196"  
        },  
        {  
          "AlarmName": "TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-  
a010-c1aaa35da296",  
          "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-a010-  
c1aaa35da296"  
        }  
      ],  
      "TargetTrackingConfiguration": {  
        "PredefinedMetricSpecification": {  
          "PredefinedMetricType": "ALBRequestCountPerTarget",  
          "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-  
alb-target-group/943f017f100becff"  
        },  
        "TargetValue": 1000.0,  
        "DisableScaleIn": false  
      }  
    }  
  ]  
}
```

```

    },
    "Enabled": true
  },
  {
    "AutoScalingGroupName": "my-asg",
    "PolicyName": "cpu40-target-tracking-scaling-policy",
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:5fd26f71-39d4-4690-82a9-
b8515c45cdde:autoScalingGroupName/my-asg:policyName/cpu40-target-tracking-scaling-
policy",
    "PolicyType": "TargetTrackingScaling",
    "StepAdjustments": [],
    "Alarms": [
      {
        "AlarmName": "TargetTracking-my-asg-
AlarmHigh-139f9789-37b9-42ad-bea5-b5b147d7f473",
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-139f9789-37b9-42ad-bea5-
b5b147d7f473"
      },
      {
        "AlarmName": "TargetTracking-my-asg-AlarmLow-bd681c67-
fc18-4c56-8468-fb8e413009c9",
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-bd681c67-fc18-4c56-8468-
fb8e413009c9"
      }
    ],
    "TargetTrackingConfiguration": {
      "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ASGAverageCPUUtilization"
      },
      "TargetValue": 40.0,
      "DisableScaleIn": false
    },
    "Enabled": true
  }
]
}

```

Weitere Informationen finden Sie unter [Dynamische Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um die Skalierungsrichtlinien eines bestimmten Namens zu beschreiben

Verwenden Sie die `--policy-names` Option, um bestimmte Skalierungsrichtlinien zurückzugeben.

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg \  
  --policy-names cpu40-target-tracking-scaling-policy
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Dynamische Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um eine Reihe von Skalierungsrichtlinien zu beschreiben

Verwenden Sie die `--max-items` Option, um eine bestimmte Anzahl von Richtlinien zurückzugeben.

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg \  
  --max-items 1
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Wenn die Ausgabe ein `NextToken` Feld enthält, verwenden Sie den Wert dieses Felds zusammen mit der `--starting-token` Option in einem nachfolgenden Aufruf, um die zusätzlichen Richtlinien abzurufen.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-  
token Z3M3LMPEXAMPLE
```

Weitere Informationen finden Sie unter [Dynamische Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribePolicies](#) in der AWS CLI Befehlsreferenz.

describe-scaling-activities

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-scaling-activities`.

AWS CLI

Beispiel 1: Um Skalierungsaktivitäten für die angegebene Gruppe zu beschreiben

In diesem Beispiel werden die Skalierungsaktivitäten für die angegebene Auto Scaling Scaling-Gruppe beschrieben.

```
aws autoscaling describe-scaling-activities \  
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "Activities": [  
    {  
      "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",  
      "Description": "Launching a new EC2 instance: i-0d44425630326060f",  
      "AutoScalingGroupName": "my-asg",  
      "Cause": "At 2020-10-30T19:35:51Z a user request update of  
AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired  
capacity from 0 to 16. At 2020-10-30T19:36:07Z an instance was started in response  
to a difference between desired and actual capacity, increasing the capacity from 0  
to 16.",  
      "StartTime": "2020-10-30T19:36:09.766Z",  
      "EndTime": "2020-10-30T19:36:41Z",  
      "StatusCode": "Successful",  
      "Progress": 100,  
      "Details": "{\"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\":  
\"us-west-2b\"}"  
    }  
  ]  
}
```

Weitere Informationen finden [Sie unter Überprüfen einer Skalierungsaktivität für eine Auto Scaling Scaling-Gruppe](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um die Skalierungsaktivitäten für eine gelöschte Gruppe zu beschreiben

Um Skalierungsaktivitäten zu beschreiben, nachdem die Auto Scaling Scaling-Gruppe gelöscht wurde, fügen Sie die `--include-deleted-groups` Option hinzu.

```
aws autoscaling describe-scaling-activities \  
  --auto-scaling-group-name my-asg \  
  --include-deleted-groups
```

--include-deleted-groups

Ausgabe:

```
{
  "Activities": [
    {
      "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
      "Description": "Launching a new EC2 instance. Status Reason: Your Spot request price of 0.001 is lower than the minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance failed.",
      "AutoScalingGroupName": "my-asg",
      "Cause": "At 2021-01-13T20:47:24Z a user request update of AutoScalingGroup constraints to min: 1, max: 5, desired: 3 changing the desired capacity from 0 to 3. At 2021-01-13T20:47:27Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 3.",
      "StartTime": "2021-01-13T20:47:30.094Z",
      "EndTime": "2021-01-13T20:47:30Z",
      "StatusCode": "Failed",
      "StatusMessage": "Your Spot request price of 0.001 is lower than the minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance failed.",
      "Progress": 100,
      "Details": "{\"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\": \"us-west-2b\"}",
      "AutoScalingGroupState": "Deleted",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:283179a2-f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Problembehandlung bei Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um eine bestimmte Anzahl von Skalierungsaktivitäten zu beschreiben

Verwenden Sie die `--max-items` Option, um eine bestimmte Anzahl von Aktivitäten zurückzugeben.

```
aws autoscaling describe-scaling-activities \
```

```
--max-items 1
```

Ausgabe:

```
{
  "Activities": [
    {
      "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",
      "Description": "Launching a new EC2 instance: i-0d44425630326060f",
      "AutoScalingGroupName": "my-asg",
      "Cause": "At 2020-10-30T19:35:51Z a user request update of
AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired
capacity from 0 to 16. At 2020-10-30T19:36:07Z an instance was started in response
to a difference between desired and actual capacity, increasing the capacity from 0
to 16.",
      "StartTime": "2020-10-30T19:36:09.766Z",
      "EndTime": "2020-10-30T19:36:41Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\":
\\\"us-west-2b\\\"}"
    }
  ]
}
```

Wenn die Ausgabe ein NextToken Feld enthält, gibt es mehr Aktivitäten. Um die zusätzlichen Aktivitäten abzurufen, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-scaling-activities \
  --starting-token Z3M3LMPEXAMPLE
```

Weitere Informationen finden [Sie unter Überprüfen einer Skalierungsaktivität für eine Auto Scaling Scaling-Gruppe](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeScalingActivities](#) in der AWS CLI Befehlsreferenz.

describe-scaling-process-types

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-scaling-process-types`.

AWS CLI

Um die verfügbaren Prozesstypen zu beschreiben

In diesem Beispiel werden die verfügbaren Prozesstypen beschrieben.

```
aws autoscaling describe-scaling-process-types
```

Ausgabe:

```
{
  "Processes": [
    {
      "ProcessName": "AZRebalance"
    },
    {
      "ProcessName": "AddToLoadBalancer"
    },
    {
      "ProcessName": "AlarmNotification"
    },
    {
      "ProcessName": "HealthCheck"
    },
    {
      "ProcessName": "InstanceRefresh"
    },
    {
      "ProcessName": "Launch"
    },
    {
      "ProcessName": "ReplaceUnhealthy"
    },
    {
      "ProcessName": "ScheduledActions"
    },
    {
      "ProcessName": "Terminate"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Aussetzen und Wiederaufnehmen von Skalierungsprozessen im Amazon EC2 Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeScalingProcessTypes](#) in der AWS CLI Befehlsreferenz.

describe-scheduled-actions

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-scheduled-actions`.

AWS CLI

Beispiel 1: Um alle geplanten Aktionen zu beschreiben

Dieses Beispiel beschreibt all Ihre geplanten Aktionen.

```
aws autoscaling describe-scheduled-actions
```

Ausgabe:

```
{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
      "StartTime": "2023-12-01T04:00:00Z",
      "Time": "2023-12-01T04:00:00Z",
      "MinSize": 1,
      "MaxSize": 6,
      "DesiredCapacity": 4,
      "TimeZone": "America/New_York"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um geplante Aktionen für die angegebene Gruppe zu beschreiben

Verwenden Sie die `--auto-scaling-group-name` Option, um die geplanten Aktionen für eine bestimmte Auto Scaling Scaling-Gruppe zu beschreiben.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "ScheduledUpdateGroupActions": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "ScheduledActionName": "my-recurring-action",  
      "Recurrence": "30 0 1 1,6,12 *",  
      "ScheduledActionARN": "arn:aws:autoscaling:us-  
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-  
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",  
      "StartTime": "2023-12-01T04:00:00Z",  
      "Time": "2023-12-01T04:00:00Z",  
      "MinSize": 1,  
      "MaxSize": 6,  
      "DesiredCapacity": 4,  
      "TimeZone": "America/New_York"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um die angegebene geplante Aktion zu beschreiben

Verwenden Sie die `--scheduled-action-names` Option, um eine bestimmte geplante Aktion zu beschreiben.

```
aws autoscaling describe-scheduled-actions \  
  --scheduled-action-names my-recurring-action
```

Ausgabe:

```
{
```

```

    "ScheduledUpdateGroupActions": [
      {
        "AutoScalingGroupName": "my-asg",
        "ScheduledActionName": "my-recurring-action",
        "Recurrence": "30 0 1 1,6,12 *",
        "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
        "StartTime": "2023-12-01T04:00:00Z",
        "Time": "2023-12-01T04:00:00Z",
        "MinSize": 1,
        "MaxSize": 6,
        "DesiredCapacity": 4,
        "TimeZone": "America/New_York"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 4: Um geplante Aktionen mit einer bestimmten Startzeit zu beschreiben

Verwenden Sie die `--start-time` Option, um die geplanten Aktionen zu beschreiben, die zu einer bestimmten Zeit beginnen.

```

aws autoscaling describe-scheduled-actions \
  --start-time "2023-12-01T04:00:00Z"

```

Ausgabe:

```

{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
      "StartTime": "2023-12-01T04:00:00Z",
      "Time": "2023-12-01T04:00:00Z",
    }
  ]
}

```

```

        "MinSize": 1,
        "MaxSize": 6,
        "DesiredCapacity": 4,
        "TimeZone": "America/New_York"
    }
]
}

```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 5: Um geplante Aktionen zu beschreiben, die zu einem bestimmten Zeitpunkt enden

Verwenden Sie die `--end-time` Option, um die geplanten Aktionen zu beschreiben, die zu einem bestimmten Zeitpunkt enden.

```

aws autoscaling describe-scheduled-actions \
  --end-time "2023-12-01T04:00:00Z"

```

Ausgabe:

```

{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
      "StartTime": "2023-12-01T04:00:00Z",
      "Time": "2023-12-01T04:00:00Z",
      "MinSize": 1,
      "MaxSize": 6,
      "DesiredCapacity": 4,
      "TimeZone": "America/New_York"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 6: Um eine bestimmte Anzahl von geplanten Aktionen zu beschreiben

Verwenden Sie die `--max-items` Option, um eine bestimmte Anzahl von geplanten Aktionen zurückzugeben.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg \  
  --max-items 1
```

Ausgabe:

```
{  
  "ScheduledUpdateGroupActions": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "ScheduledActionName": "my-recurring-action",  
      "Recurrence": "30 0 1 1,6,12 *",  
      "ScheduledActionARN": "arn:aws:autoscaling:us-  
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-  
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",  
      "StartTime": "2023-12-01T04:00:00Z",  
      "Time": "2023-12-01T04:00:00Z",  
      "MinSize": 1,  
      "MaxSize": 6,  
      "DesiredCapacity": 4,  
      "TimeZone": "America/New_York"  
    }  
  ]  
}
```

Wenn die Ausgabe ein `NextToken` Feld enthält, gibt es mehr geplante Aktionen. Um die zusätzlichen geplanten Aktionen abzurufen, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

Weitere Informationen finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeScheduledActions](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-tags`.

AWS CLI

Um alle Tags zu beschreiben

In diesem Beispiel werden alle Ihre Tags beschrieben.

```
aws autoscaling describe-tags
```

Ausgabe:

```
{
  "Tags": [
    {
      "ResourceType": "auto-scaling-group",
      "ResourceId": "my-asg",
      "PropagateAtLaunch": true,
      "Value": "Research",
      "Key": "Dept"
    },
    {
      "ResourceType": "auto-scaling-group",
      "ResourceId": "my-asg",
      "PropagateAtLaunch": true,
      "Value": "WebServer",
      "Key": "Role"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um Tags für eine bestimmte Gruppe zu beschreiben

Verwenden Sie die `--filters` Option, um Tags für eine bestimmte Auto Scaling Scaling-Gruppe zu beschreiben.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

Weitere Informationen finden Sie unter [Tagging Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um die angegebene Anzahl von Tags zu beschreiben

Um eine bestimmte Anzahl von Tags zurückzugeben, verwenden Sie die `--max-items` Option.

```
aws autoscaling describe-tags \  
  --max-items 1
```

Wenn die Ausgabe ein `NextToken` Feld enthält, gibt es mehr Tags. Um die zusätzlichen Tags zu erhalten, verwenden Sie den Wert dieses Felds mit der `--starting-token` Option in einem nachfolgenden Aufruf wie folgt.

```
aws autoscaling describe-tags \  
  --filters Name=auto-scaling-group,Values=my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

Weitere Informationen finden Sie unter [Tagging Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTags](#) in der AWS CLI Befehlsreferenz.

describe-termination-policy-types

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-termination-policy-types`.

AWS CLI

Um die verfügbaren Typen von Kündigungsrichtlinien zu beschreiben

In diesem Beispiel werden die verfügbaren Arten von Kündigungsrichtlinien beschrieben.

```
aws autoscaling describe-termination-policy-types
```

Ausgabe:

```
{  
  "TerminationPolicyTypes": [  
    "AllocationStrategy",  
    "ClosestToNextInstanceHour",
```



```
    "Default",
    "NewestInstance",
    "OldestInstance",
    "OldestLaunchConfiguration",
    "OldestLaunchTemplate"
  ]
}
```

Weitere Informationen finden Sie unter [Steuern, welche Auto Scaling-Instances während der Skalierung beendet werden im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DescribeTerminationPolicyTypes](#) in der AWS CLI Befehlsreferenz.

describe-warm-pool

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-warm-pool`.

AWS CLI

Um einen warmen Pool zu beschreiben

In diesem Beispiel wird der warme Pool für die angegebene Auto Scaling Scaling-Gruppe beschrieben.

```
aws autoscaling describe-warm-pool \
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{
  "WarmPoolConfiguration": {
    "MinSize": 2,
    "PoolState": "Stopped"
  },
  "Instances": [
    {
      "InstanceId": "i-070a5bbc7e7f40dc5",
      "InstanceType": "t2.micro",
      "AvailabilityZone": "us-west-2c",
      "LifecycleState": "Warmed:Pending",
      "HealthStatus": "Healthy",
      "LaunchTemplate": {
```

```

        "LaunchTemplateId": "lt-00a731f6e9fa48610",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "Version": "6"
    }
},
{
    "InstanceId": "i-0b52f061814d3bd2d",
    "InstanceType": "t2.micro",
    "AvailabilityZone": "us-west-2b",
    "LifecycleState": "Warmup:Pending",
    "HealthStatus": "Healthy",
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-00a731f6e9fa48610",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "Version": "6"
    }
}
]
}

```

Weitere Informationen finden Sie unter [Warm Pools for Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeWarmPool](#) in der AWS CLI Befehlsreferenz.

detach-instances

Das folgende Codebeispiel zeigt, wie man es benutzt `detach-instances`.

AWS CLI

So trennen Sie eine Instance von einer Auto Scaling Scaling-Gruppe

In diesem Beispiel wird die angegebene Instance von der angegebenen Auto Scaling Scaling-Gruppe getrennt.

```

aws autoscaling detach-instances \
  --instance-ids i-030017cfa84b20135 \
  --auto-scaling-group-name my-asg \
  --should-decrement-desired-capacity

```

Ausgabe:

```
{
  "Activities": [
    {
      "ActivityId": "5091cb52-547a-47ce-a236-c9ccbc2cb2c9",
      "AutoScalingGroupName": "my-asg",
      "Description": "Detaching EC2 instance: i-030017cfa84b20135",
      "Cause": "At 2020-10-31T17:35:04Z instance i-030017cfa84b20135 was
detached in response to a user request, shrinking the capacity from 2 to 1.",
      "StartTime": "2020-04-12T15:02:16.179Z",
      "StatusCode": "InProgress",
      "Progress": 50,
      "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":
\\\"us-west-2c\\\"}"
    }
  ]
}
```

- API-Einheiten finden Sie unter [DetachInstances AWS CLI Befehlsreferenz](#).

detach-load-balancer-target-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `detach-load-balancer-target-groups`.

AWS CLI

Um eine Load Balancer-Zielgruppe von einer Auto Scaling Scaling-Gruppe zu trennen

In diesem Beispiel wird die angegebene Load Balancer-Zielgruppe von der angegebenen Auto Scaling Scaling-Gruppe getrennt.

```
aws autoscaling detach-load-balancer-target-groups \
  --auto-scaling-group-name my-asg \
  --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Dieser Befehl erzeugt keine Ausgabe

Weitere Informationen finden Sie unter [Einen Load Balancer zu Ihrer Auto Scaling Scaling-Gruppe hinzufügen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DetachLoadBalancerTargetGroups](#) in der AWS CLI Befehlsreferenz.

detach-load-balancers

Das folgende Codebeispiel zeigt, wie man es benutzt detach-load-balancers.

AWS CLI

So trennen Sie einen Classic Load Balancer von einer Auto Scaling Scaling-Gruppe

In diesem Beispiel wird der angegebene Classic Load Balancer von der angegebenen Auto Scaling Scaling-Gruppe getrennt.

```
aws autoscaling detach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einen Load Balancer zu Ihrer Auto Scaling Scaling-Gruppe hinzufügen im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DetachLoadBalancers](#) in der AWS CLI Befehlsreferenz.

disable-metrics-collection

Das folgende Codebeispiel zeigt, wie man es benutzt disable-metrics-collection.

AWS CLI

So deaktivieren Sie die Erfassung von Metriken für eine Auto Scaling Scaling-Gruppe

In diesem Beispiel wird die Erfassung der GroupDesiredCapacity Metrik für die angegebene Auto Scaling Scaling-Gruppe deaktiviert.

```
aws autoscaling disable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [CloudWatch Monitoring-Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DisableMetricsCollection](#) in der AWS CLI Befehlsreferenz.

enable-metrics-collection

Das folgende Codebeispiel zeigt, wie man es benutzt `enable-metrics-collection`.

AWS CLI

Beispiel 1: So aktivieren Sie die Erfassung von Metriken für eine Auto Scaling Scaling-Gruppe

In diesem Beispiel wird die Datenerfassung für die angegebene Auto Scaling Scaling-Gruppe aktiviert.

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --granularity "1Minute"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [CloudWatch Monitoring-Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um Daten für die angegebene Metrik für eine Auto Scaling Scaling-Gruppe zu sammeln

Verwenden Sie die `--metrics` Option, um Daten für eine bestimmte Metrik zu sammeln.

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity --granularity "1Minute"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [CloudWatch Monitoring-Metriken für Ihre Auto Scaling Scaling-Gruppen und -Instances](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableMetricsCollection](#) in der AWS CLI Befehlsreferenz.

enter-standby

Das folgende Codebeispiel zeigt, wie man es benutzt `enter-standby`.

AWS CLI

Um Instanzen in den Standby-Modus zu versetzen

In diesem Beispiel wird die angegebene Instanz in den Standby-Modus versetzt. Dies ist nützlich, um eine Instanz zu aktualisieren oder Fehler zu beheben, die derzeit in Betrieb ist.

```
aws autoscaling enter-standby \
  --instance-ids i-061c63c5eb45f0416 \
  --auto-scaling-group-name my-asg \
  --should-decrement-desired-capacity
```

Ausgabe:

```
{
  "Activities": [
    {
      "ActivityId": "ffa056b4-6ed3-41ba-ae7c-249dfae6eba1",
      "AutoScalingGroupName": "my-asg",
      "Description": "Moving EC2 instance to Standby: i-061c63c5eb45f0416",
      "Cause": "At 2020-10-31T20:31:00Z instance i-061c63c5eb45f0416 was moved
to standby in response to a user request, shrinking the capacity from 1 to 0.",
      "StartTime": "2020-10-31T20:31:00.949Z",
      "StatusCode": "InProgress",
      "Progress": 50,
      "Details": "{\"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\":
us-west-2c\"}"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Scaling-Instance-Lebenszyklus](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [EnterStandby](#) in der AWS CLI Befehlsreferenz.

execute-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `execute-policy`.

AWS CLI

Um eine Skalierungsrichtlinie auszuführen

In diesem Beispiel wird die Skalierungsrichtlinie ausgeführt, die `my-step-scale-out-policy` für die angegebene Auto Scaling Scaling-Gruppe benannt ist.

```
aws autoscaling execute-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name my-step-scale-out-policy \  
  --metric-value 95 \  
  --breach-threshold 80
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Step and Simple Scaling Policies](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [ExecutePolicy](#) in der AWS CLI Befehlsreferenz.

exit-standby

Das folgende Codebeispiel zeigt, wie man es benutzt `exit-standby`.

AWS CLI

Um Instanzen aus dem Standby-Modus zu verschieben

In diesem Beispiel wird die angegebene Instanz aus dem Standby-Modus versetzt.

```
aws autoscaling exit-standby \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "Activities": [  
    {  
      "ActivityId": "142928e1-a2dc-453a-9b24-b85ad6735928",  
      "AutoScalingGroupName": "my-asg",  
      "Description": "Moving EC2 instance out of Standby:  
i-061c63c5eb45f0416",  
      "Cause": "At 2020-10-31T20:32:50Z instance i-061c63c5eb45f0416 was moved  
out of standby in response to a user request, increasing the capacity from 0 to  
1.",  
      "StartTime": "2020-10-31T20:32:50.222Z",  
      "StatusCode": "PreInService",  
      "Progress": 30,  
    }  
  ]  
}
```

```
        "Details": "{ \"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\":  
        \"us-west-2c\" }"  
    ]  
}
```

Weitere Informationen finden Sie unter [Vorübergehendes Entfernen von Instances aus Ihrer Auto Scaling Group](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [ExitStandby](#) in der AWS CLI Befehlsreferenz.

put-lifecycle-hook

Das folgende Codebeispiel zeigt, wie man es benutzt `put-lifecycle-hook`.

AWS CLI

Beispiel 1: Um einen Lifecycle-Hook zu erstellen

In diesem Beispiel wird ein Lifecycle-Hook erstellt, der bei allen neu gestarteten Instances mit einem Timeout von 4800 Sekunden aufgerufen wird. Dies ist nützlich, um die Instanzen im Wartezustand zu halten, bis die Benutzerdatenskripts abgeschlossen sind, oder um eine AWS Lambda-Funktion mit aufzurufen. EventBridge

```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --heartbeat-timeout 4800
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Wenn bereits ein Lifecycle-Hook mit demselben Namen existiert, wird er durch den neuen Lifecycle-Hook überschrieben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um eine SNS Amazon-E-Mail-Nachricht zu senden, um Sie über Instance-Statusübergänge zu informieren

In diesem Beispiel wird ein Lifecycle-Hook mit dem SNS Thema und der IAM Rolle von Amazon erstellt, der verwendet werden soll, um Benachrichtigungen beim Start der Instance zu erhalten.


```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --notification-target-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \  
  --role-arn arn:aws:iam::123456789012:role/my-auto-scaling-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um eine Nachricht in einer SQS Amazon-Warteschlange zu veröffentlichen

In diesem Beispiel wird ein Lifecycle-Hook erstellt, der eine Nachricht mit Metadaten in der angegebenen SQS Amazon-Warteschlange veröffentlicht.

```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --notification-target-arn arn:aws:sqs:us-west-2:123456789012:my-sqs-queue \  
  --role-arn arn:aws:iam::123456789012:role/my-notification-role \  
  --notification-metadata "SQS message metadata"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutLifecycleHook](#) in der AWS CLI Befehlsreferenz.

put-notification-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `put-notification-configuration`.

AWS CLI

Um eine Benachrichtigung hinzuzufügen

In diesem Beispiel wird die angegebene Benachrichtigung der angegebenen Auto Scaling Scaling-Gruppe hinzugefügt.

```
aws autoscaling put-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \  
  --notification-type autoscaling:TEST_NOTIFICATION
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [SNS Amazon-Benachrichtigungen erhalten, wenn Ihre Auto Scaling-Gruppe skaliert](#) wird im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutNotificationConfiguration](#) in der AWS CLI Befehlsreferenz.

put-scaling-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `put-scaling-policy`.

AWS CLI

So fügen Sie einer Auto Scaling Scaling-Gruppe eine Skalierungsrichtlinie für die Zielverfolgung hinzu

Im folgenden `put-scaling-policy` Beispiel wird eine Skalierungsrichtlinie für die Zielverfolgung auf die angegebene Auto Scaling Scaling-Gruppe angewendet. Die Ausgabe enthält die Namen ARNs und der beiden CloudWatch Alarmer, die in Ihrem Namen erstellt wurden. Wenn bereits eine Skalierungsrichtlinie mit demselben Namen existiert, wird sie durch die neue Skalierungsrichtlinie überschrieben.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \  
  --policy-name alb1000-target-tracking-scaling-policy \  
  --policy-type TargetTrackingScaling \  
  --target-tracking-configuration file://config.json
```

Inhalt von `config.json`:

```
{  
  "TargetValue": 1000.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ALBRequestCountPerTarget",  
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-  
group/943f017f100becff"  
  }  
}
```

```
}

```

Ausgabe:

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:228f02c2-
c665-4bfd-aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/alb1000-target-
tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
      "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
    },
    {
      "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
      "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
    }
  ]
}
```

Weitere Beispiele finden Sie unter [Beispiel für Skalierungsrichtlinien für die AWS Befehlszeilenschnittstelle \(AWS CLI\)](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutScalingPolicy](#) in der AWS CLI Befehlsreferenz.

put-scheduled-update-group-action

Das folgende Codebeispiel zeigt, wie man es benutzt `put-scheduled-update-group-action`.

AWS CLI

Beispiel 1: So fügen Sie einer Auto Scaling Scaling-Gruppe eine geplante Aktion hinzu

In diesem Beispiel wird die angegebene geplante Aktion der angegebenen Auto Scaling Scaling-Gruppe hinzugefügt.

```
aws autoscaling put-scheduled-update-group-action \
  --auto-scaling-group-name my-asg \
```

```
--scheduled-action-name my-scheduled-action \  
--start-time "2023-05-12T08:00:00Z" \  
--min-size 2 \  
--max-size 6 \  
--desired-capacity 4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Wenn eine geplante Aktion mit demselben Namen bereits existiert, wird sie durch die neue geplante Aktion überschrieben.

Weitere Beispiele finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um einen wiederkehrenden Zeitplan anzugeben

In diesem Beispiel wird eine geplante Aktion zur Skalierung nach einem wiederkehrenden Zeitplan erstellt, der jedes Jahr am ersten Januar, Juni und Dezember um 00:30 Uhr ausgeführt werden soll.

```
aws autoscaling put-scheduled-update-group-action \  
  --auto-scaling-group-name my-asg \  
  --scheduled-action-name my-recurring-action \  
  --recurrence "30 0 1 1,6,12 *" \  
  --min-size 2 \  
  --max-size 6 \  
  --desired-capacity 4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Wenn bereits eine geplante Aktion mit demselben Namen existiert, wird sie durch die neue geplante Aktion überschrieben.

Weitere Beispiele finden Sie unter [Geplante Skalierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutScheduledUpdateGroupAction](#) in der AWS CLI Befehlsreferenz.

put-warm-pool

Das folgende Codebeispiel zeigt, wie man es benutzt `put-warm-pool`.

AWS CLI

Um einen warmen Pool zu erstellen

Im folgenden Beispiel wird ein warmer Pool für die angegebene Auto Scaling Scaling-Gruppe erstellt.

```
aws autoscaling put-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --min-size 2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Wenn bereits ein warmer Pool vorhanden ist, wird er aktualisiert.

Weitere Informationen finden Sie unter [Warm Pools for Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [PutWarmPool](#) in der AWS CLI Befehlsreferenz.

record-lifecycle-action-heartbeat

Das folgende Codebeispiel zeigt, wie man es benutzt `record-lifecycle-action-heartbeat`.

AWS CLI

Um einen Lifecycle-Aktions-Heartbeat aufzuzeichnen

In diesem Beispiel wird ein Lifecycle-Aktions-Heartbeat aufgezeichnet, um die Instance im Status „Ausstehend“ zu halten.

```
aws autoscaling record-lifecycle-action-heartbeat \  
  --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon EC2 Auto Scaling Lifecycle Hooks](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [RecordLifecycleActionHeartbeat](#) in der AWS CLI Befehlsreferenz.

resume-processes

Das folgende Codebeispiel zeigt, wie man es benutzt `resume-processes`.

AWS CLI

Um unterbrochene Prozesse wieder aufzunehmen

In diesem Beispiel wird der angegebene unterbrochene Skalierungsprozess für die angegebene Auto Scaling Scaling-Gruppe wieder aufgenommen.

```
aws autoscaling resume-processes \  
  --auto-scaling-group-name my-asg \  
  --scaling-processes AlarmNotification
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aussetzen und Wiederaufnehmen von Skalierungsprozessen im Amazon EC2 Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [ResumeProcesses](#) in der AWS CLI Befehlsreferenz.

rollback-instance-refresh

Das folgende Codebeispiel zeigt, wie man es benutzt `rollback-instance-refresh`.

AWS CLI

Um eine Instanzaktualisierung rückgängig zu machen

Im folgenden `rollback-instance-refresh` Beispiel wird eine laufende Instanzaktualisierung für die angegebene Auto Scaling-Gruppe rückgängig gemacht.

```
aws autoscaling rollback-instance-refresh \  
  --auto-scaling-group-name my-asg
```

Ausgabe:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Weitere Informationen finden Sie unter [Rückgängigmachen von Änderungen mit einem Rollback](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [RollbackInstanceRefresh](#) in der AWS CLI Befehlsreferenz.

set-desired-capacity

Das folgende Codebeispiel zeigt, wie man es benutzt `set-desired-capacity`.

AWS CLI

So legen Sie die gewünschte Kapazität für eine Auto Scaling Scaling-Gruppe fest

In diesem Beispiel wird die gewünschte Kapazität für die angegebene Auto Scaling Scaling-Gruppe festgelegt.

```
aws autoscaling set-desired-capacity \  
  --auto-scaling-group-name my-asg \  
  --desired-capacity 2 \  
  --honor-cooldown
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

- API-Einheiten finden Sie [SetDesiredCapacity](#) in der AWS CLI Befehlsreferenz.

set-instance-health

Das folgende Codebeispiel zeigt, wie man es benutzt `set-instance-health`.

AWS CLI

Um den Integritätsstatus einer Instanz festzulegen

In diesem Beispiel wird der Integritätsstatus der angegebenen Instanz auf `Unhealthy` festgelegt.

```
aws autoscaling set-instance-health \  
  --instance-id i-061c63c5eb45f0416 \  
  --health-status Unhealthy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SetInstanceHealth](#) in der AWS CLI Befehlsreferenz.

set-instance-protection

Das folgende Codebeispiel zeigt, wie man es benutzt `set-instance-protection`.

AWS CLI

Beispiel 1: Um die Instanzschutzeinstellung für eine Instanz zu aktivieren

In diesem Beispiel wird der Instanzschutz für die angegebene Instanz aktiviert.

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg --protected-from-scale-in
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die Instanzschutzeinstellung für eine Instanz zu deaktivieren

In diesem Beispiel wird der Instanzschutz für die angegebene Instanz deaktiviert.

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg \  
  --no-protected-from-scale-in
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SetInstanceProtection](#) in der AWS CLI Befehlsreferenz.

start-instance-refresh

Das folgende Codebeispiel zeigt, wie man es benutzt `start-instance-refresh`.

AWS CLI

Beispiel 1: Um eine Instanzaktualisierung mithilfe von Befehlszeilenparametern zu starten

Im folgenden `start-instance-refresh` Beispiel wird eine Instanzaktualisierung mithilfe von Befehlszeilenargumenten gestartet. Der optionale `preferences` Parameter gibt eine Zahl `InstanceWarmup` von 60 Sekunden und eine Zahl `MinHealthyPercentage` von 50 Prozent an.

```
aws autoscaling start-instance-refresh \  
  --auto-scaling-group-name my-asg \  
  --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```


Ausgabe:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Weitere Informationen finden Sie unter [Starten einer Instance-Aktualisierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um eine Instance-Aktualisierung mithilfe einer JSON Datei zu starten

Im folgenden `start-instance-refresh` Beispiel wird eine Instanzaktualisierung mithilfe einer JSON Datei gestartet. Sie können die Auto Scaling Group angeben und Ihre gewünschte Konfiguration und Einstellungen in einer JSON Datei definieren, wie im folgenden Beispiel gezeigt.

```
aws autoscaling start-instance-refresh \
  --cli-input-json file://config.json
```

Inhalt von `config.json`:

```
{
  "AutoScalingGroupName": "my-asg",
  "DesiredConfiguration": {
    "LaunchTemplate": {
      "LaunchTemplateId": "lt-068f72b729example",
      "Version": "$Default"
    }
  },
  "Preferences": {
    "InstanceWarmup": 60,
    "MinHealthyPercentage": 50,
    "AutoRollback": true,
    "ScaleInProtectedInstances": Ignore,
    "StandbyInstances": Terminate
  }
}
```

Ausgabe:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

```
}
```

Weitere Informationen finden Sie unter [Starten einer Instance-Aktualisierung](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [StartInstanceRefresh](#) in der AWS CLI Befehlsreferenz.

suspend-processes

Das folgende Codebeispiel zeigt, wie man es benutzt `suspend-processes`.

AWS CLI

So setzen Sie Auto Scaling Scaling-Prozesse aus

In diesem Beispiel wird der angegebene Skalierungsprozess für die angegebene Auto Scaling Scaling-Gruppe unterbrochen.

```
aws autoscaling suspend-processes \  
  --auto-scaling-group-name my-asg \  
  --scaling-processes AlarmNotification
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aussetzen und Wiederaufnehmen von Skalierungsprozessen im Amazon EC2 Auto Scaling](#) Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [SuspendProcesses](#) in der AWS CLI Befehlsreferenz.

terminate-instance-in-auto-scaling-group

Das folgende Codebeispiel zeigt, wie man es benutzt `terminate-instance-in-auto-scaling-group`.

AWS CLI

Um eine Instance in einer Auto Scaling Scaling-Gruppe zu beenden

In diesem Beispiel wird die angegebene Instance aus der angegebenen Auto Scaling Scaling-Gruppe beendet, ohne die Größe der Gruppe zu aktualisieren. Amazon EC2 Auto Scaling startet eine Ersatz-Instance, nachdem die angegebene Instance beendet wurde.

```
aws autoscaling terminate-instance-in-auto-scaling-group \  
  --instance-id i-061c63c5eb45f0416 \  
  --no-should-decrement-desired-capacity
```

Ausgabe:

```
{  
  "Activities": [  
    {  
      "ActivityId": "8c35d601-793c-400c-fcd0-f64a27530df7",  
      "AutoScalingGroupName": "my-asg",  
      "Description": "Terminating EC2 instance: i-061c63c5eb45f0416",  
      "Cause": "",  
      "StartTime": "2020-10-31T20:34:25.680Z",  
      "StatusCode": "InProgress",  
      "Progress": 0,  
      "Details": "{\"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\":  
        \"us-west-2c\"}"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [TerminateInstanceInAutoScalingGroup](#) in der AWS CLI Befehlsreferenz.

update-auto-scaling-group

Das folgende Codebeispiel zeigt, wie man es benutzt `update-auto-scaling-group`.

AWS CLI

Beispiel 1: So aktualisieren Sie die Größenbeschränkungen einer Auto Scaling Scaling-Gruppe

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe mit einer Mindestgröße von 2 und einer Maximalgröße von 10 aktualisiert.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --min-size 2 \  
  --max-size 10
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Festlegen von Kapazitätsgrenzen für Ihre Auto Scaling Scaling-Gruppe](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 2: Um Elastic Load Balancing Health Checks hinzuzufügen und anzugeben, welche Availability Zones und Subnetze verwendet werden sollen

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe aktualisiert, um Elastic Load Balancing Health Checks hinzuzufügen. Dieser Befehl aktualisiert auch den Wert von `--vpc-zone-identifier` mit einer Liste von Subnetzen IDs in mehreren Availability Zones.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --health-check-type ELB \  
  --health-check-grace-period 600 \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Elastic Load Balancing und Amazon EC2 Auto Scaling](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 3: Um die Platzierungsgruppe und die Kündigungsrichtlinie zu aktualisieren

In diesem Beispiel werden die zu verwendende Platzierungsgruppe und die Kündigungsrichtlinie aktualisiert.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --placement-group my-placement-group \  
  --termination-policies "OldestInstance"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auto Scaling Scaling-Gruppen](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 4: Um die neueste Version der Startvorlage zu verwenden

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe aktualisiert, sodass sie die neueste Version der angegebenen Startvorlage verwendet.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Vorlagen starten](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 5: Um eine bestimmte Version der Startvorlage zu verwenden

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe so aktualisiert, dass sie eine bestimmte Version einer Startvorlage anstelle der neuesten Version oder Standardversion verwendet.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Vorlagen starten](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Beispiel 6: Um eine Richtlinie für gemischte Instanzen zu definieren und einen Kapazitätsausgleich zu ermöglichen

In diesem Beispiel wird die angegebene Auto Scaling Scaling-Gruppe so aktualisiert, dass sie eine Richtlinie für gemischte Instanzen verwendet, und ermöglicht einen Kapazitätsausgleich. Mit dieser Struktur können Sie Gruppen mit Spot- und On-Demand-Kapazitäten angeben und unterschiedliche Startvorlagen für unterschiedliche Architekturen verwenden.

```
aws autoscaling update-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Inhalt von `config.json`:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "CapacityRebalance": true,  
  "MixedInstancesPolicy": {
```

```
"LaunchTemplate": {
  "LaunchTemplateSpecification": {
    "LaunchTemplateName": "my-launch-template-for-x86",
    "Version": "$Latest"
  },
  "Overrides": [
    {
      "InstanceType": "c6g.large",
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template-for-arm",
        "Version": "$Latest"
      }
    },
    {
      "InstanceType": "c5.large"
    },
    {
      "InstanceType": "c5a.large"
    }
  ]
},
"InstancesDistribution": {
  "OnDemandPercentageAboveBaseCapacity": 50,
  "SpotAllocationStrategy": "capacity-optimized"
}
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auto Scaling Scaling-Gruppen mit mehreren Instance-Typen und Kaufoptionen](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAutoScalingGroup](#) in der AWS CLI Befehlsreferenz.

Beispiele für Auto Scaling-Pläne mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Auto Scaling-Plänen Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-scaling-plan

Das folgende Codebeispiel zeigt die Verwendung `create-scaling-plan`.

AWS CLI

Um einen Skalierungsplan zu erstellen

Im folgenden `create-scaling-plan` Beispiel wird ein Skalierungsplan erstellt, der `my-scaling-plan` mithilfe einer bereits erstellten JSON Datei (namens `config.json`) benannt wird. Die Struktur des Skalierungsplans umfasst eine Skalierungsanweisung für eine Auto Scaling Scaling-Gruppe mit dem Namen `my-asg`. Er gibt die `TagFilters`-Eigenschaft als Anwendungsquelle an und ermöglicht prädiktive Skalierung und dynamische Skalierung.

```
aws autoscaling-plans create-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --cli-input-json file://~/config.json
```

Inhalt der `config.json` Datei:

```
{  
  "ApplicationSource": {  
    "TagFilters": [  
      {  
        "Key": "purpose",  
        "Values": [  
          "my-application"  
        ]  
      }  
    ]  
  },  
  "ScalingInstructions": [  
    {  
      "Group": "my-asg",  
      "MinSize": 1,  
      "MaxSize": 10,  
      "DesiredSize": 1,  
      "Cooldown": 300,  
      "TargetScalingType": "Predictive",  
      "DynamicScaling": {  
        "Enabled": true,  
        "MinSize": 1,  
        "MaxSize": 10,  
        "DesiredSize": 1,  
        "Cooldown": 300,  
        "TargetScalingType": "Predictive"  
      }  
    }  
  ]  
}
```

```

    {
      "ServiceNamespace": "autoscaling",
      "ResourceId": "autoScalingGroup/my-asg",
      "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",
      "ScheduledActionBufferTime": 300,
      "PredictiveScalingMaxCapacityBehavior":
"SetForecastCapacityToMaxCapacity",
      "PredictiveScalingMode": "ForecastAndScale",
      "PredefinedLoadMetricSpecification": {
        "PredefinedLoadMetricType": "ASGTotalCPUUtilization"
      },
      "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
      "MinCapacity": 1,
      "MaxCapacity": 4,
      "TargetTrackingConfigurations": [
        {
          "PredefinedScalingMetricSpecification": {
            "PredefinedScalingMetricType": "ASGAverageCPUUtilization"
          },
          "TargetValue": 50
        }
      ]
    }
  ]
}

```

Ausgabe:

```

{
  "ScalingPlanVersion": 1
}

```

Weitere Informationen finden Sie im [AWS Auto Scaling Scaling-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateScalingPlan](#) in der AWS CLI Befehlsreferenz.

delete-scaling-plan

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-scaling-plan`.

AWS CLI

Um einen Skalierungsplan zu löschen

Im folgenden `delete-scaling-plan` Beispiel wird der angegebene Skalierungsplan gelöscht.

```
aws autoscaling-plans delete-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [AWS Auto Scaling Scaling-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DeleteScalingPlan](#) in der AWS CLI Befehlsreferenz.

`describe-scaling-plan-resources`

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-scaling-plan-resources`.

AWS CLI

Um die skalierbaren Ressourcen für einen Skalierungsplan zu beschreiben

Im folgenden `describe-scaling-plan-resources` Beispiel werden Details zu der einzelnen skalierbaren Ressource (einer Auto Scaling Scaling-Gruppe) angezeigt, die dem angegebenen Skalierungsplan zugeordnet ist.

```
aws autoscaling-plans describe-scaling-plan-resources \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1
```

Ausgabe:

```
{  
  "ScalingPlanResources": [  
    {  
      "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",  
      "ScalingPlanVersion": 1,  
      "ResourceId": "autoScalingGroup/my-asg",  
      "ScalingStatusCode": "Active",  
      "ScalingStatusMessage": "Target tracking scaling policies have been  
applied to the resource.",  
      "ScalingPolicies": [  
        {  
          "PolicyName": "AutoScaling-my-asg-b1ab65ae-4be3-4634-bd64-  
c7471662b251",
```

```

        "PolicyType": "TargetTrackingScaling",
        "TargetTrackingConfiguration": {
            "PredefinedScalingMetricSpecification": {
                "PredefinedScalingMetricType":
"ALBRequestCountPerTarget",
                "ResourceLabel": "app/my-alb/f37c06a68c1748aa/
targetgroup/my-target-group/6d4ea56ca2d6a18d"
            },
            "TargetValue": 40.0
        }
    ],
    "ServiceNamespace": "autoscaling",
    "ScalingPlanName": "my-scaling-plan"
}
]
}

```

Weitere Informationen finden Sie unter [Was ist AWS Auto Scaling?](#) im AWS Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeScalingPlanResources](#) in der AWS CLI Befehlsreferenz.

describe-scaling-plans

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-scaling-plans`.

AWS CLI

Um einen Skalierungsplan zu beschreiben

Im folgenden `describe-scaling-plans` Beispiel werden die Details des angegebenen Skalierungsplans angezeigt.

```

aws autoscaling-plans describe-scaling-plans \
  --scaling-plan-names scaling-plan-with-asg-and-ddb

```

Ausgabe:

```

{
  "ScalingPlans": [
    {
      "LastMutatingRequestTime": 1565388443.963,

```

```
"ScalingPlanVersion": 1,
"CreationTime": 1565388443.963,
"ScalingInstructions": [
  {
    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
    "ScalableDimension":
"autoscaling:autoScalingGroup:DesiredCapacity",
    "TargetTrackingConfigurations": [
      {
        "PredefinedScalingMetricSpecification": {
          "PredefinedScalingMetricType":
"ASGAverageCPUUtilization"
        },
        "TargetValue": 50.0,
        "EstimatedInstanceWarmup": 300,
        "DisableScaleIn": false
      }
    ],
    "ResourceId": "autoScalingGroup/my-asg",
    "DisableDynamicScaling": false,
    "MinCapacity": 1,
    "ServiceNamespace": "autoscaling",
    "MaxCapacity": 10
  },
  {
    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
    "ScalableDimension": "dynamodb:table:ReadCapacityUnits",
    "TargetTrackingConfigurations": [
      {
        "PredefinedScalingMetricSpecification": {
          "PredefinedScalingMetricType":
"DynamoDBReadCapacityUtilization"
        },
        "TargetValue": 50.0,
        "ScaleInCooldown": 60,
        "DisableScaleIn": false,
        "ScaleOutCooldown": 60
      }
    ],
    "ResourceId": "table/my-table",
    "DisableDynamicScaling": false,
    "MinCapacity": 5,
    "ServiceNamespace": "dynamodb",
    "MaxCapacity": 10000
  }
]
```

```

    },
    {
      "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
      "TargetTrackingConfigurations": [
        {
          "PredefinedScalingMetricSpecification": {
            "PredefinedScalingMetricType":
"DynamoDBWriteCapacityUtilization"
          },
          "TargetValue": 50.0,
          "ScaleInCooldown": 60,
          "DisableScaleIn": false,
          "ScaleOutCooldown": 60
        }
      ],
      "ResourceId": "table/my-table",
      "DisableDynamicScaling": false,
      "MinCapacity": 5,
      "ServiceNamespace": "dynamodb",
      "MaxCapacity": 10000
    }
  ],
  "ApplicationSource": {
    "TagFilters": [
      {
        "Values": [
          "my-application-id"
        ],
        "Key": "application"
      }
    ]
  },
  "StatusStartTime": 1565388455.836,
  "ScalingPlanName": "scaling-plan-with-asg-and-ddb",
  "StatusMessage": "Scaling plan has been created and applied to all
resources.",
  "StatusCode": "Active"
}
]
}

```

Weitere Informationen finden Sie unter [Was ist AWS Auto Scaling?](#) im AWS Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeScalingPlans](#) in der AWS CLI Befehlsreferenz.

get-scaling-plan-resource-forecast-data

Das folgende Codebeispiel zeigt, wie man es benutzt `get-scaling-plan-resource-forecast-data`.

AWS CLI

Um Daten zur Lastprognose abzurufen

In diesem Beispiel werden Lastprognosedaten für eine skalierbare Ressource (eine Auto Scaling Scaling-Gruppe) abgerufen, die dem angegebenen Skalierungsplan zugeordnet ist.

```
aws autoscaling-plans get-scaling-plan-resource-forecast-data \
  --scaling-plan-name my-scaling-plan \
  --scaling-plan-version 1 \
  --service-namespace "autoscaling" \
  --resource-id autoScalingGroup/my-asg \
  --scalable-dimension "autoscaling:autoScalingGroup:DesiredCapacity" \
  --forecast-data-type "LoadForecast" \
  --start-time "2019-08-30T00:00:00Z" \
  --end-time "2019-09-06T00:00:00Z"
```

Ausgabe:

```
{
  "Datapoints": [...]
}
```

Weitere Informationen finden Sie unter [Was ist AWS Auto Scaling](#) im AWS Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [GetScalingPlanResourceForecastData](#) in der AWS CLI Befehlsreferenz.

update-scaling-plan

Das folgende Codebeispiel zeigt, wie man es benutzt `update-scaling-plan`.

AWS CLI

Um einen Skalierungsplan zu aktualisieren

Im folgenden `update-scaling-plan` Beispiel wird die Skalierungsmetrik für eine Auto Scaling Scaling-Gruppe im angegebenen Skalierungsplan geändert.

```
aws autoscaling-plans update-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1 \  
  --scaling-instructions  
  '{"ScalableDimension":"autoscaling:autoScalingGroup:DesiredCapacity","ResourceId":"autoScal  
my-asg","ServiceNamespace":"autoscaling","TargetTrackingConfigurations":  
[{"PredefinedScalingMetricSpecification":  
  {"PredefinedScalingMetricType":"ALBRequestCountPerTarget","ResourceLabel":"app/my-  
alb/f37c06a68c1748aa/targetgroup/my-target-  
group/6d4ea56ca2d6a18d"},"TargetValue":40.0}],"MinCapacity": 1,"MaxCapacity": 10}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Was ist AWS Auto Scaling?](#) im AWS Auto Scaling Scaling-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateScalingPlan](#) in der AWS CLI Befehlsreferenz.

AWS Backup Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Backup.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-backup-plan

Das folgende Codebeispiel zeigt die Verwendung `create-backup-plan`.

AWS CLI

Um einen Backup-Plan zu erstellen

Im folgenden `create-backup-plan` Beispiel wird der angegebene Backup-Plan mit einer Aufbewahrung von 35 Tagen erstellt.

```
aws backup create-backup-plan \
--backup-plan "{\"BackupPlanName\":\"Example-Backup-Plan\",\"Rules\":[{\"RuleName\":
\"DailyBackups\",\"ScheduleExpression\":\"cron(0 5 ? * * *)\",\"StartWindowMinutes
\":480,\"TargetBackupVaultName\":\"Default\",\"Lifecycle\":{\"DeleteAfterDays
\":35}}]}"
```

Ausgabe:

```
{
  "BackupPlanId": "1fa3895c-a7f5-484a-a371-2dd6a1a9f729",
  "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:1fa3895c-
a7f5-484a-a371-2dd6a1a9f729",
  "CreationDate": 1568928754.747,
  "VersionId": "ZjQ2ZTI5YWQtZDg5Yi00MzYzLWJmZTAtMDI1Mzh1MDhjYjEz"
}
```

Weitere Informationen finden Sie unter [Erstellen eines Backup-Plans](#) im AWS Backup Developer Guide.

- API-Einheiten finden Sie [CreateBackupPlan](#) in der AWS CLI Befehlsreferenz.

create-backup-vault

Das folgende Codebeispiel zeigt die Verwendung `create-backup-vault`.

AWS CLI

Um einen Backup-Tresor zu erstellen

Im folgenden `create-backup-vault` Beispiel wird ein Backup-Tresor mit dem angegebenen Namen erstellt.

```
aws backup create-backup-vault
  --backup-vault-name sample-vault
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "BackupVaultName": "sample-vault",
  "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-
vault",
  "CreationDate": 1568928338.385
}
```

Weitere Informationen finden Sie unter [Creating a Backup Vault](#) im AWS Backup Developer Guide.

- API-Einheiten finden Sie [CreateBackupVault](#) in der AWS CLI Befehlsreferenz.

get-backup-plan-from-template

Das folgende Codebeispiel zeigt die Verwendung von `get-backup-plan-from-template`.

AWS CLI

Um einen vorhandenen Backup-Plan aus einer Vorlage abzurufen

Im folgenden `get-backup-plan-from-template` Beispiel wird ein vorhandener Sicherungsplan aus einer Vorlage abgerufen, die ein tägliches Backup mit einer Aufbewahrung von 35 Tagen spezifiziert.

```
aws backup get-backup-plan-from-template \
  --backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

Ausgabe:

```
{
  "BackupPlanDocument": {
    "Rules": [
      {
```



```

    "RuleName": "DailyBackups",
    "ScheduleExpression": "cron(0 5 ? * * *)",
    "StartWindowMinutes": 480,
    "Lifecycle": {
      "DeleteAfterDays": 35
    }
  ]
}

```

Weitere Informationen finden Sie unter [Erstellen eines Backup-Plans](#) im AWS Backup Developer Guide.

- API-Einheiten finden Sie [GetBackupPlanFromTemplate](#) in der AWS CLI Befehlsreferenz.

get-backup-plan

Das folgende Codebeispiel zeigt die Verwendung `get-backup-plan`.

AWS CLI

Um die Details eines Backup-Plans abzurufen

Im folgenden `get-backup-plan` Beispiel werden die Details des angegebenen Sicherungsplans angezeigt.

```

aws backup get-backup-plan \
  --backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"

```

Ausgabe:

```

{
  "BackupPlan": {
    "BackupPlanName": "Example-Backup-Plan",
    "Rules": [
      {
        "RuleName": "DailyBackups",
        "TargetBackupVaultName": "Default",
        "ScheduleExpression": "cron(0 5 ? * * *)",
        "StartWindowMinutes": 480,
        "CompletionWindowMinutes": 10080,
        "Lifecycle": {

```

```
        "DeleteAfterDays": 35
      },
      "RuleId": "70e0ccdc-e9df-4e83-82ad-c1e5a9471cc3"
    }
  ]
},
"BackupPlanId": "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5",
"BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:fcbf5d8f-
bd77-4f3a-9c97-f24fb3d373a5",
"VersionId": "NjQ2ZTZkODktMGVhNy00MmQ0LWE4YjktZTkxNTQ3OTkyYTcw",
"CreationDate": 1568926091.57
}
```

Weitere Informationen finden Sie unter [Erstellen eines Backup-Plans](#) im AWS Backup Developer Guide.

- API-Einheiten finden Sie [GetBackupPlan](#) in der AWS CLI Befehlsreferenz.

list-backup-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-backup-jobs`.

AWS CLI

Beispiel 1: Um alle Backup-Jobs aufzulisten

Im folgenden `list-backup-jobs` Beispiel werden Metadaten zu Ihren Backup-Jobs in Ihrem AWS Konto zurückgegeben.

```
aws backup list-backup-jobs
```

Ausgabe:

```
{
  "BackupJobs": [
    {
      "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "BackupVaultName": "Default",
      "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-12345678901234567",
      "CreationDate": 1600721892.929,
    }
  ]
}
```

```

        "State": "CREATED",
        "PercentDone": "0.0",
        "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
        "StartBy": 1600725492.929,
        "ResourceType": "EC2"
    },
    {
        "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "BackupVaultName": "Default",
        "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
        "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
        "CreationDate": 1600721724.77,
        "CompletionDate": 1600721744.488,
        "State": "COMPLETED",
        "PercentDone": "100.0",
        "BackupSizeInBytes": 71,
        "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
        "StartBy": 1600725324.77,
        "ResourceType": "EFS"
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen eines Backups](#) im AWS Backup Developer Guide.

Beispiel 2: Um abgeschlossene Backup-Jobs aufzulisten

Im folgenden `list-backup-jobs` Beispiel werden Metadaten zu Ihren abgeschlossenen Backup-Jobs in Ihrem AWS Konto zurückgegeben.

```
aws backup list-backup-jobs \
  --by-state COMPLETED
```

Ausgabe:

```
{
```

```
"BackupJobs": [  
  {  
    "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "BackupVaultName": "Default",  
    "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-  
vault:Default",  
    "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-  
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
    "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-  
system/fs-12345678",  
    "CreationDate": 1600721724.77,  
    "CompletionDate": 1600721744.488,  
    "State": "COMPLETED",  
    "PercentDone": "100.0",  
    "BackupSizeInBytes": 71,  
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/  
AWSBackupDefaultServiceRole",  
    "StartBy": 1600725324.77,  
    "ResourceType": "EFS"  
  }  
]
```

Weitere Informationen finden Sie unter [Erstellen eines Backups](#) im AWS Backup Developer Guide.

- API-Einheiten finden Sie [ListBackupJobs](#) in der AWS CLI Befehlsreferenz.

AWS Batch Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Batch.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-job`.

AWS CLI

Um einen Job zu stornieren

In diesem Beispiel wird ein Job mit der angegebenen Job-ID storniert.

Befehl:

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --  
reason "Cancelling job."
```

- API-Einheiten finden Sie unter [CancelJob AWS CLI Befehlsreferenz](#).

create-compute-environment

Das folgende Codebeispiel zeigt die Verwendung `create-compute-environment`.

AWS CLI

Um eine verwaltete Rechenumgebung mit On-Demand-Instanzen zu erstellen

In diesem Beispiel wird eine verwaltete Rechenumgebung mit bestimmten C4-Instanztypen erstellt, die bei Bedarf gestartet werden. Die Rechenumgebung heißt OnDemand C4.

Befehl:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/  
C4OnDemand.json
```

JSON-Dateiformat:

```
{  
  "computeEnvironmentName": "C4OnDemand",  
  "type": "MANAGED",
```

```
"state": "ENABLED",
"computeResources": {
  "type": "EC2",
  "minvCpus": 0,
  "maxvCpus": 128,
  "desiredvCpus": 48,
  "instanceTypes": [
    "c4.large",
    "c4.xlarge",
    "c4.2xlarge",
    "c4.4xlarge",
    "c4.8xlarge"
  ],
  "subnets": [
    "subnet-220c0e0a",
    "subnet-1a95556d",
    "subnet-978f6dce"
  ],
  "securityGroupIds": [
    "sg-cf5093b2"
  ],
  "ec2KeyPair": "id_rsa",
  "instanceRole": "ecsInstanceRole",
  "tags": {
    "Name": "Batch Instance - C4OnDemand"
  }
},
"serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

Ausgabe:

```
{
  "computeEnvironmentName": "C4OnDemand",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

Um eine verwaltete Rechenumgebung mit Spot-Instances zu erstellen

In diesem Beispiel wird eine verwaltete Rechenumgebung mit dem Instance-Typ M4 erstellt, die gestartet wird, wenn der Spot-Angebotspreis 20% des On-Demand-Preises für den Instance-Typ erreicht oder darunter liegt. Die Rechenumgebung heißt M4Spot.

Befehl:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/M4Spot.json
```

JSONDateiformat:

```
{
  "computeEnvironmentName": "M4Spot",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "type": "SPOT",
    "spotIamFleetRole": "arn:aws:iam::012345678910:role/aws-ec2-spot-fleet-role",
    "minvCpus": 0,
    "maxvCpus": 128,
    "desiredvCpus": 4,
    "instanceTypes": [
      "m4"
    ],
    "bidPercentage": 20,
    "subnets": [
      "subnet-220c0e0a",
      "subnet-1a95556d",
      "subnet-978f6dce"
    ],
    "securityGroupIds": [
      "sg-cf5093b2"
    ],
    "ec2KeyPair": "id_rsa",
    "instanceRole": "ecsInstanceRole",
    "tags": {
      "Name": "Batch Instance - M4Spot"
    }
  },
  "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

Ausgabe:

```
{
  "computeEnvironmentName": "M4Spot",
```

```
"computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-  
environment/M4Spot"  
}
```

- API-Einheiten finden Sie [CreateComputeEnvironment](#) in der AWS CLI Befehlsreferenz.

create-job-queue

Das folgende Codebeispiel zeigt die Verwendung `create-job-queue`.

AWS CLI

Um eine Auftragswarteschlange mit niedriger Priorität mit einer einzigen Rechenumgebung zu erstellen

In diesem Beispiel wird eine Job-Warteschlange mit dem Namen `LowPriority` erstellt, die die `M4Spot`-Rechenumgebung verwendet.

Befehl:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/  
LowPriority.json
```

JSON-Dateiformat:

```
{  
  "jobQueueName": "LowPriority",  
  "state": "ENABLED",  
  "priority": 10,  
  "computeEnvironmentOrder": [  
    {  
      "order": 1,  
      "computeEnvironment": "M4Spot"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",
```



```
"jobQueueName": "LowPriority"
}
```

Um eine Auftragswarteschlange mit hoher Priorität und zwei Rechenumgebungen zu erstellen

In diesem Beispiel wird eine Jobwarteschlange mit dem Namen erstellt `HighPriority`, die die `OnDemand C4`-Rechenumgebung mit der Reihenfolge 1 und die `M4Spot`-Rechenumgebung mit der Reihenfolge 2 verwendet. Der Scheduler versucht zuerst, Jobs in der `OnDemand C4`-Computerumgebung zu platzieren.

Befehl:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/HighPriority.json
```

JSONDateiformat:

```
{
  "jobQueueName": "HighPriority",
  "state": "ENABLED",
  "priority": 1,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "C4OnDemand"
    },
    {
      "order": 2,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

Ausgabe:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
  "jobQueueName": "HighPriority"
}
```

- APIEinzelheiten finden Sie [CreateJobQueue](#) in der AWS CLI Befehlsreferenz.

delete-compute-environment

Das folgende Codebeispiel zeigt die Verwendung `delete-compute-environment`.

AWS CLI

Um eine Rechenumgebung zu löschen

In diesem Beispiel wird die OnDemand P2-Rechenumgebung gelöscht.

Befehl:

```
aws batch delete-compute-environment --compute-environment P2onDemand
```

- API-Einheiten finden Sie [DeleteComputeEnvironment](#) in der AWS CLI Befehlsreferenz.

delete-job-queue

Das folgende Codebeispiel zeigt die Verwendung `delete-job-queue`.

AWS CLI

Um eine Job-Warteschlange zu löschen

In diesem Beispiel wird die GPGPU Job-Warteschlange gelöscht.

Befehl:

```
aws batch delete-job-queue --job-queue GPGPU
```

- API-Einheiten finden Sie [DeleteJobQueue](#) in der AWS CLI Befehlsreferenz.

deregister-job-definition

Das folgende Codebeispiel zeigt die Verwendung `deregister-job-definition`.

AWS CLI

Um die Registrierung einer Jobdefinition aufzuheben

In diesem Beispiel wird die Registrierung einer Jobdefinition namens `sleep10` aufgehoben.

Befehl:

```
aws batch deregister-job-definition --job-definition sleep10
```

- API-Einheiten finden Sie [DeregisterJobDefinition](#) in AWS CLI der Befehlsreferenz.

describe-compute-environments

Das folgende Codebeispiel zeigt die Verwendung `describe-compute-environments`.

AWS CLI

Um eine Rechenumgebung zu beschreiben

Dieses Beispiel beschreibt die OnDemand P2-Rechenumgebung.

Befehl:

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

Ausgabe:

```
{
  "computeEnvironments": [
    {
      "status": "VALID",
      "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole",
      "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/P2OnDemand",
      "computeResources": {
        "subnets": [
          "subnet-220c0e0a",
          "subnet-1a95556d",
          "subnet-978f6dce"
        ],
        "tags": {
          "Name": "Batch Instance - P2OnDemand"
        },
        "desiredvCpus": 48,
        "minvCpus": 0,
        "instanceTypes": [
```

```

        "p2"
      ],
      "securityGroupIds": [
        "sg-cf5093b2"
      ],
      "instanceRole": "ecsInstanceRole",
      "maxvCpus": 128,
      "type": "EC2",
      "ec2KeyPair": "id_rsa"
    },
    "statusReason": "ComputeEnvironment Healthy",
    "ecsClusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/
P20nDemand_Batch_2c06f29d-d1fe-3a49-879d-42394c86effc",
    "state": "ENABLED",
    "computeEnvironmentName": "P20nDemand",
    "type": "MANAGED"
  }
]
}

```

- API-Einheiten finden Sie [DescribeComputeEnvironments](#) in der AWS CLI Befehlsreferenz.

describe-job-definitions

Das folgende Codebeispiel zeigt die Verwendung `describe-job-definitions`.

AWS CLI

Um aktive Jobdefinitionen zu beschreiben

In diesem Beispiel werden alle Ihre aktiven Jobdefinitionen beschrieben.

Befehl:

```
aws batch describe-job-definitions --status ACTIVE
```

Ausgabe:

```
{
  "jobDefinitions": [
    {
      "status": "ACTIVE",

```

```
    "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-
definition/sleep60:1",
    "containerProperties": {
      "mountPoints": [],
      "parameters": {},
      "image": "busybox",
      "environment": {},
      "vcpus": 1,
      "command": [
        "sleep",
        "60"
      ],
      "volumes": [],
      "memory": 128,
      "ulimits": []
    },
    "type": "container",
    "jobDefinitionName": "sleep60",
    "revision": 1
  }
]
```

- API-Einheiten finden Sie [DescribeJobDefinitions](#) in der AWS CLI Befehlsreferenz.

describe-job-queues

Das folgende Codebeispiel zeigt die Verwendung `describe-job-queues`.

AWS CLI

Um eine Job-Warteschlange zu beschreiben

Dieses Beispiel beschreibt die HighPriority Job-Warteschlange.

Befehl:

```
aws batch describe-job-queues --job-queues HighPriority
```

Ausgabe:

```
{
```

```

    "jobQueues": [
      {
        "status": "VALID",
        "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
        "computeEnvironmentOrder": [
          {
            "computeEnvironment": "arn:aws:batch:us-east-1:012345678910:compute-environment/C4OnDemand",
            "order": 1
          }
        ],
        "statusReason": "JobQueue Healthy",
        "priority": 1,
        "state": "ENABLED",
        "jobQueueName": "HighPriority"
      }
    ]
  }
}

```

- API-Einheiten finden Sie [DescribeJobQueues](#) in der AWS CLI Befehlsreferenz.

describe-jobs

Das folgende Codebeispiel zeigt die Verwendung `describe-jobs`.

AWS CLI

Um einen Job zu beschreiben

Das folgende `describe-jobs` Beispiel beschreibt einen Job mit der angegebenen Job-ID.

```

aws batch describe-jobs \
  --jobs bcf0b186-a532-4122-842e-2ccab8d54efb

```

Ausgabe:

```

{
  "jobs": [
    {
      "status": "SUBMITTED",
      "container": {

```

```

        "mountPoints": [],
        "image": "busybox",
        "environment": [],
        "vcpus": 1,
        "command": [
            "sleep",
            "60"
        ],
        "volumes": [],
        "memory": 128,
        "ulimits": []
    },
    "parameters": {},
    "jobDefinition": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep60:1",
    "jobQueue": "arn:aws:batch:us-east-1:012345678910:job-queue/
HighPriority",
    "jobId": "bcf0b186-a532-4122-842e-2ccab8d54efb",
    "dependsOn": [],
    "jobName": "example",
    "createdAt": 1480483387803
}
]
}

```

- API-Einheiten finden Sie [DescribeJobs](#) unter AWS CLI Befehlsreferenz.

list-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-jobs`.

AWS CLI

Um laufende Jobs aufzulisten

In diesem Beispiel werden die laufenden Jobs in der HighPriority Job-Warteschlange aufgelistet.

Befehl:

```
aws batch list-jobs --job-queue HighPriority
```

Ausgabe:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
    }
  ]
}
```

Um übermittelte Jobs aufzulisten

In diesem Beispiel werden Jobs in der HighPriority Auftragswarteschlange aufgeführt, die sich im SUBMITTED Auftragsstatus befinden.

Befehl:

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

Ausgabe:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
    }
  ]
}
```

- API-Einheiten finden Sie [ListJobs](#) unter AWS CLI Befehlsreferenz.

register-job-definition

Das folgende Codebeispiel zeigt die Verwendung `register-job-definition`.

AWS CLI

Um eine Jobdefinition zu registrieren

In diesem Beispiel wird eine Jobdefinition für einen einfachen Container-Job registriert.

Befehl:

```
aws batch register-job-definition --job-definition-name sleep30 --type container --  
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":  
[ "sleep", "30" ]}'
```

Ausgabe:

```
{  
  "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/  
sleep30:1",  
  "jobDefinitionName": "sleep30",  
  "revision": 1  
}
```

- API-Einheiten finden Sie [RegisterJobDefinition](#) unter AWS CLI Befehlsreferenz.

submit-job

Das folgende Codebeispiel zeigt die Verwendung `submit-job`.

AWS CLI

Um einen Job einzureichen

In diesem Beispiel wird ein einfacher Container-Job namens `example` an die `HighPriority` Auftragswarteschlange gesendet.

Befehl:

```
aws batch submit-job --job-name example --job-queue HighPriority --job-  
definition sleep60
```

Ausgabe:

```
{  
  "jobName": "example",  
  "jobId": "876da822-4198-45f2-a252-6cea32512ea8"  
}
```

- API-Einheiten finden Sie [SubmitJob](#) in der AWS CLI Befehlsreferenz.

terminate-job

Das folgende Codebeispiel zeigt die Verwendung `terminate-job`.

AWS CLI

Um einen Job zu beenden

In diesem Beispiel wird ein Job mit der angegebenen Job-ID beendet.

Befehl:

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --  
reason "Terminating job."
```

- API-Einheiten finden Sie unter [TerminateJob AWS CLI](#) Befehlsreferenz.

update-compute-environment

Das folgende Codebeispiel zeigt die Verwendung `update-compute-environment`.

AWS CLI

Um eine Rechenumgebung zu aktualisieren

In diesem Beispiel wird die OnDemand P2-Rechenumgebung deaktiviert, sodass sie gelöscht werden kann.

Befehl:

```
aws batch update-compute-environment --compute-environment P2OnDemand --  
state DISABLED
```

Ausgabe:

```
{  
  "computeEnvironmentName": "P2OnDemand",  
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-  
environment/P2OnDemand"
```

```
}
```

- API-Einheiten finden Sie [UpdateComputeEnvironment](#) in der AWS CLI Befehlsreferenz.

update-job-queue

Das folgende Codebeispiel zeigt die Verwendung `update-job-queue`.

AWS CLI

Um eine Job-Warteschlange zu aktualisieren

In diesem Beispiel wird eine Auftragswarteschlange deaktiviert, sodass sie gelöscht werden kann.

Befehl:

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

Ausgabe:

```
{  
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",  
  "jobQueueName": "GPGPU"  
}
```

- API-Einheiten finden Sie [UpdateJobQueue](#) in der AWS CLI Befehlsreferenz.

AWS Budgets Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Budgets.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-budget

Das folgende Codebeispiel zeigt die Verwendung `create-budget`.

AWS CLI

Um ein Kosten- und Nutzungsbudget zu erstellen

Mit dem folgenden `create-budget` Befehl wird ein Kosten- und Nutzungsbudget erstellt.

```
aws budgets create-budget \  
  --account-id 111122223333 \  
  --budget file://budget.json \  
  --notifications-with-subscribers file://notifications-with-subscribers.json
```

Inhalt von `budget.json`:

```
{  
  "BudgetLimit": {  
    "Amount": "100",  
    "Unit": "USD"  
  },  
  "BudgetName": "Example Tag Budget",  
  "BudgetType": "COST",  
  "CostFilters": {  
    "TagKeyValue": [  
      "user:Key$value1",  
      "user:Key$value2"  
    ]  
  },  
  "CostTypes": {  
    "IncludeCredit": true,  
    "IncludeDiscount": true,  
    "IncludeOtherSubscription": true,  
    "IncludeRecurring": true,  
    "IncludeRefund": true,  
    "IncludeSubscription": true,  
    "IncludeSupport": true,  
  }  
}
```

```

    "IncludeTax": true,
    "IncludeUpfront": true,
    "UseBlended": false
  },
  "TimePeriod": {
    "Start": 1477958399,
    "End": 3706473600
  },
  "TimeUnit": "MONTHLY"
}

```

Inhalt von `notifications-with-subscribers.json`:

```

[
  {
    "Notification": {
      "ComparisonOperator": "GREATER_THAN",
      "NotificationType": "ACTUAL",
      "Threshold": 80,
      "ThresholdType": "PERCENTAGE"
    },
    "Subscribers": [
      {
        "Address": "example@example.com",
        "SubscriptionType": "EMAIL"
      }
    ]
  }
]

```

- API-Einheiten finden Sie [CreateBudget](#) in der AWS CLI Befehlsreferenz.

create-notification

Das folgende Codebeispiel zeigt die Verwendung `create-notification`.

AWS CLI

Um eine Benachrichtigung für das angegebene Kosten- und Nutzungsbudget zu erstellen

In diesem Beispiel wird eine Benachrichtigung für das angegebene Kosten- und Nutzungsbudget erstellt.

Befehl:

```
aws budgets create-notification --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- API-Einheiten finden Sie [CreateNotification](#) unter AWS CLI Befehlsreferenz.

create-subscriber

Das folgende Codebeispiel zeigt die Verwendung `create-subscriber`.

AWS CLI

Um einen Abonnenten für eine Benachrichtigung zu erstellen, die mit einem Kosten- und Nutzungsbudget verknüpft ist

In diesem Beispiel wird ein Abonnent für die angegebene Benachrichtigung erstellt.

Befehl:

```
aws budgets create-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- API-Einheiten finden Sie [CreateSubscriber](#) in der AWS CLI Befehlsreferenz.

delete-budget

Das folgende Codebeispiel zeigt die Verwendung `delete-budget`.

AWS CLI

Um ein Kosten- und Nutzungsbudget zu löschen

In diesem Beispiel wird das angegebene Kosten- und Nutzungsbudget gelöscht.

Befehl:

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- API-Einheiten finden Sie unter [DeleteBudget AWS CLI](#) Befehlsreferenz.

delete-notification

Das folgende Codebeispiel zeigt die Verwendung `delete-notification`.

AWS CLI

Um eine Benachrichtigung aus einem Budget zu löschen

In diesem Beispiel wird die angegebene Benachrichtigung aus dem angegebenen Budget gelöscht.

Befehl:

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT
```

- API-Einheiten finden Sie unter [DeleteNotification AWS CLI](#) Befehlsreferenz.

delete-subscriber

Das folgende Codebeispiel zeigt die Verwendung `delete-subscriber`.

AWS CLI

Um einen Abonnenten aus einer Benachrichtigung zu löschen

In diesem Beispiel wird der angegebene Abonnent aus der angegebenen Benachrichtigung gelöscht.

Befehl:

```
aws budgets delete-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- API-Einheiten finden Sie unter [DeleteSubscriber AWS CLI](#) Befehlsreferenz.

describe-budget

Das folgende Codebeispiel zeigt die Verwendung `describe-budget`.

AWS CLI

Um ein mit einem Konto verknüpftes Budget abzurufen

In diesem Beispiel wird das angegebene Kosten- und Nutzungsbudget abgerufen.

Befehl:

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

Ausgabe:

```
{
  "Budget": {
    "CalculatedSpend": {
      "ForecastedSpend": {
        "Amount": "2641.548000000000022919266484677791595458984375",
        "Unit": "USD"
      },
      "ActualSpend": {
        "Amount": "604.45600000000000172803993336856365203857421875",
        "Unit": "USD"
      }
    },
    "BudgetType": "COST",
    "BudgetLimit": {
      "Amount": "100",
      "Unit": "USD"
    },
    "BudgetName": "Example Budget",
    "CostTypes": {
      "IncludeOtherSubscription": true,
      "IncludeUpfront": true,
      "IncludeRefund": true,
      "UseBlended": false,
      "IncludeDiscount": true,
    }
  }
}
```



```
        "UseAmortized": false,
        "IncludeTax": true,
        "IncludeCredit": true,
        "IncludeSupport": true,
        "IncludeRecurring": true,
        "IncludeSubscription": true
    },
    "TimeUnit": "MONTHLY",
    "TimePeriod": {
        "Start": 1477958399.0,
        "End": 3706473600.0
    },
    "CostFilters": {
        "AZ": [
            "us-east-1"
        ]
    }
}
}
```

- API-Einheiten finden Sie unter [DescribeBudget AWS CLI-Befehlsreferenz](#).

describe-budgets

Das folgende Codebeispiel zeigt die Verwendung `describe-budgets`.

AWS CLI

Um die mit einem Konto verknüpften Budgets abzurufen

In diesem Beispiel werden die Kosten- und Nutzungsbudgets für ein Konto abgerufen.

Befehl:

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

Ausgabe:

```
{
  "Budgets": [
    {
```

```

    "CalculatedSpend": {
      "ForecastedSpend": {
        "Amount": "2641.548000000000022919266484677791595458984375",
        "Unit": "USD"
      },
      "ActualSpend": {
        "Amount": "604.45600000000000172803993336856365203857421875",
        "Unit": "USD"
      }
    },
    "BudgetType": "COST",
    "BudgetLimit": {
      "Amount": "100",
      "Unit": "USD"
    },
    "BudgetName": "Example Budget",
    "CostTypes": {
      "IncludeOtherSubscription": true,
      "IncludeUpfront": true,
      "IncludeRefund": true,
      "UseBlended": false,
      "IncludeDiscount": true,
      "UseAmortized": false,
      "IncludeTax": true,
      "IncludeCredit": true,
      "IncludeSupport": true,
      "IncludeRecurring": true,
      "IncludeSubscription": true
    },
    "TimeUnit": "MONTHLY",
    "TimePeriod": {
      "Start": 1477958399.0,
      "End": 3706473600.0
    },
    "CostFilters": {
      "AZ": [
        "us-east-1"
      ]
    }
  }
]
}

```

- API-Einheiten finden Sie unter [DescribeBudgets AWS CLI-Befehlsreferenz](#).

describe-notifications-for-budget

Das folgende Codebeispiel zeigt die Verwendung `describe-notifications-for-budget`.

AWS CLI

Um die Benachrichtigungen für ein Budget abzurufen

In diesem Beispiel werden die Benachrichtigungen für ein Kosten- und Nutzungsbudget abgerufen.

Befehl:

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-name "Example Budget" --max-results 5
```

Ausgabe:

```
{
  "Notifications": [
    {
      "Threshold": 80.0,
      "ComparisonOperator": "GREATER_THAN",
      "NotificationType": "ACTUAL"
    }
  ]
}
```

- API-Einheiten finden Sie unter [DescribeNotificationsForBudget AWS CLI](#) Befehlsreferenz.

describe-subscribers-for-notification

Das folgende Codebeispiel zeigt die Verwendung `describe-subscribers-for-notification`.

AWS CLI

Um die Abonnenten für eine Budgetbenachrichtigung abzurufen

In diesem Beispiel werden die Abonnenten für eine Budgetbenachrichtigung zu Kosten und Nutzung abgerufen.

Befehl:

```
aws budgets describe-subscribers-for-notification --  
account-id 111122223333 --budget-name "Example Budget" --  
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE  
--max-results 5
```

Ausgabe:

```
{  
  "Subscribers": [  
    {  
      "SubscriptionType": "EMAIL",  
      "Address": "example2@example.com"  
    },  
    {  
      "SubscriptionType": "EMAIL",  
      "Address": "example@example.com"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [DescribeSubscribersForNotification AWS CLI Befehlsreferenz](#).

update-budget

Das folgende Codebeispiel zeigt die Verwendung `update-budget`.

AWS CLI

Um ein Budget durch ein Kosten- und Nutzungsbudget zu ersetzen

In diesem Beispiel wird ein Kosten- und Nutzungsbudget durch ein neues Budget ersetzt.

Befehl:

```
aws budgets update-budget --account-id 111122223333 --new-budget file://new-budget.json
```

new-budget.json:

```
{
```

```
"BudgetLimit": {
  "Amount": "100",
  "Unit": "USD"
},
"BudgetName": "Example Budget",
"BudgetType": "COST",
"CostFilters": {
  "AZ" : [ "us-east-1" ]
},
"CostTypes": {
  "IncludeCredit": false,
  "IncludeDiscount": true,
  "IncludeOtherSubscription": true,
  "IncludeRecurring": true,
  "IncludeRefund": true,
  "IncludeSubscription": true,
  "IncludeSupport": true,
  "IncludeTax": true,
  "IncludeUpfront": true,
  "UseBlended": false,
  "UseAmortized": true
},
"TimePeriod": {
  "Start": 1477958399,
  "End": 3706473600
},
"TimeUnit": "MONTHLY"
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [UpdateBudget](#).AWS CLI

update-notification

Das folgende Codebeispiel zeigt die Verwendung `update-notification`.

AWS CLI

Um eine Benachrichtigung für ein Kosten- und Nutzungsbudget zu ersetzen

In diesem Beispiel wird eine 80-%-Benachrichtigung für ein Kosten- und Nutzungsbudget durch eine 90-%-Benachrichtigung ersetzt.

Befehl:

```
aws budgets update-notification --account-id 111122223333 --budget-name "Example Budget" --old-notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT --new-notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=90,ThresholdType=PERCENT
```

- API-Einheiten finden Sie [UpdateNotification](#) unter AWS CLI Befehlsreferenz.

update-subscriber

Das folgende Codebeispiel zeigt die Verwendung `update-subscriber`.

AWS CLI

Um einen Abonnenten gegen ein Kosten- und Nutzungsbudget zu ersetzen

In diesem Beispiel wird der Abonnent durch ein Kosten- und Nutzungsbudget ersetzt.

Befehl:

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- API-Einheiten finden Sie [UpdateSubscriber](#) in der AWS CLI Befehlsreferenz.

Amazon Chime Chime-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie Amazon Chime verwenden. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-phone-number-with-user

Das folgende Codebeispiel zeigt die Verwendung `associate-phone-number-with-user`.

AWS CLI

Um einem Benutzer eine Telefonnummer zuzuordnen

Im folgenden `associate-phone-number-with-user` Beispiel wird die angegebene Telefonnummer einem Benutzer zugeordnet.

```
aws chime associate-phone-number-with-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \  
  --e164-phone-number " +12065550100"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Benutzertelefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [AssociatePhoneNumberWithUser](#) in der AWS CLI Befehlsreferenz.

associate-phone-numbers-with-voice-connector-group

Das folgende Codebeispiel zeigt, wie man es benutzt `associate-phone-numbers-with-voice-connector-group`.

AWS CLI

So verknüpfen Sie Telefonnummern mit einer Amazon Chime Voice Connector-Gruppe

Im folgenden `associate-phone-numbers-with-voice-connector-group` Beispiel werden die angegebenen Telefonnummern einer Amazon Chime Voice Connector-Gruppe zugeordnet.

```
aws chime associate-phone-numbers-with-voice-connector-group \  
  --voice-connector-group-id 12345678-9012-3456-7890-123456789012 \  
  --phone-numbers " +12065550100 +12065550101 +12065550102"
```

```
--voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
--e164-phone-numbers "+12065550100" "+12065550101" \  
--force-associate
```

Ausgabe:

```
{  
  "PhoneNumberErrors": []  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [AssociatePhoneNumbersWithVoiceConnectorGroup](#) in der AWS CLI Befehlsreferenz.

associate-phone-numbers-with-voice-connector

Das folgende Codebeispiel zeigt, wie man es benutzt `associate-phone-numbers-with-voice-connector`.

AWS CLI

So verknüpfen Sie Telefonnummern mit einem Amazon Chime Voice Connector

Das folgende `associate-phone-numbers-with-voice-connector` Beispiel verknüpft die angegebenen Telefonnummern mit einem Amazon Chime Voice Connector.

```
aws chime associate-phone-numbers-with-voice-connector \  
--voice-connector-id abcdef1ghij2klmno3pqr4 \  
--e164-phone-numbers "+12065550100" "+12065550101" \  
--force-associate
```

Ausgabe:

```
{  
  "PhoneNumberErrors": []  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [AssociatePhoneNumbersWithVoiceConnector](#) in der AWS CLI Befehlsreferenz.

associate-signin-delegate-groups-with-account

Das folgende Codebeispiel zeigt, wie man es benutzt `associate-signin-delegate-groups-with-account`.

AWS CLI

Um Gruppen von Anmeldedelegierten zuzuordnen

Das folgende `associate-signin-delegate-groups-with-account` Beispiel verknüpft die angegebene Gruppe von Anmeldedelegierten mit dem angegebenen Amazon Chime Chime-Konto.

```
aws chime associate-signin-delegate-groups-with-account \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --signin-delegate-groups GroupName=my_users
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerzugriff und Benutzerberechtigungen verwalten](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [AssociateSigninDelegateGroupsWithAccount](#) in der AWS CLI Befehlsreferenz.

batch-create-room-membership

Das folgende Codebeispiel zeigt, wie man es benutzt `batch-create-room-membership`.

AWS CLI

Um Mitgliedschaften in mehreren Räumen zu erstellen

Im folgenden `batch-create-room-membership` Beispiel werden einem Chatroom mehrere Benutzer als Chatroom-Mitglieder hinzugefügt. Außerdem werden den Benutzern Administrator- und Mitgliederrollen zugewiesen.

```
aws chime batch-create-room-membership \  
  --room-id my-room-id \  
  --members my-member-1 my-member-2 my-member-3 \  
  --roles my-member-1:Administrator my-member-2:Administrator my-member-3:Member
```

```
--account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
--room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
--membership-item-list "MemberId=1ab2345c-67de-8901-  
f23g-45h678901j2k,Role=Administrator" "MemberId=2ab2345c-67de-8901-  
f23g-45h678901j2k,Role=Member"
```

Ausgabe:

```
{  
  "ResponseMetadata": {  
    "RequestId": "169ba401-d886-475f-8b3f-e01eac6fadfb",  
    "HTTPStatusCode": 201,  
    "HTTPHeaders": {  
      "x-amzn-requestid": "169ba401-d886-475f-8b3f-e01eac6fadfb",  
      "content-type": "application/json",  
      "content-length": "13",  
      "date": "Mon, 02 Dec 2019 22:46:58 GMT",  
      "connection": "keep-alive"  
    },  
    "RetryAttempts": 0  
  },  
  "Errors": []  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchCreateRoomMembership](#) in der AWS CLI Befehlsreferenz.

batch-delete-phone-number

Das folgende Codebeispiel zeigt, wie man es benutzt `batch-delete-phone-number`.

AWS CLI

Um mehrere Telefonnummern zu löschen

Im folgenden `batch-delete-phone-number` Beispiel werden alle angegebenen Telefonnummern gelöscht.

```
aws chime batch-delete-phone-number \  

```

```
--phone-number-ids "%2B12065550100" "%2B12065550101"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "PhoneNumberErrors": []
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- APIEinzelheiten finden Sie [BatchDeletePhoneNumber](#) in der AWS CLI Befehlsreferenz.

batch-suspend-user

Das folgende Codebeispiel zeigt, wie man es benutzt `batch-suspend-user`.

AWS CLI

Um mehrere Benutzer zu sperren

Im folgenden `batch-suspend-user` Beispiel werden die aufgelisteten Benutzer vom angegebenen Amazon Chime Chime-Konto gesperrt.

```
aws chime batch-suspend-user \
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

Ausgabe:

```
{
  "UserErrors": []
}
```

- APIEinzelheiten finden Sie [BatchSuspendUser](#) in der AWS CLI Befehlsreferenz.

batch-unsuspend-user

Das folgende Codebeispiel zeigt, wie man es benutzt `batch-unsuspend-user`.

AWS CLI

Um die Sperre mehrerer Benutzer aufzuheben

Im folgenden `batch-unsuspend-user` Beispiel wird jede vorherige Sperrung für die aufgelisteten Benutzer des angegebenen Amazon Chime Chime-Kontos aufgehoben.

```
aws chime batch-unsuspend-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

Ausgabe:

```
{  
  "UserErrors": []  
}
```

- API-Einheiten finden Sie unter [BatchUnsuspendUser AWS CLI](#) Befehlsreferenz.

batch-update-phone-number

Das folgende Codebeispiel zeigt, wie man es benutzt `batch-update-phone-number`.

AWS CLI

Um mehrere Telefonnummern-Produkttypen gleichzeitig zu aktualisieren

Im folgenden `batch-update-phone-number` Beispiel werden die Produkttypen für alle angegebenen Telefonnummern aktualisiert.

```
aws chime batch-update-phone-number \  
  --update-phone-number-request-items PhoneNumberId=  
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=  
%2B12065550101,ProductType=BusinessCalling
```

Ausgabe:

```
{
```

```
"PhoneNumberErrors": []
}
```

Um mehrere Rufnummern, die Namen anrufen, gleichzeitig zu aktualisieren

Im folgenden `batch-update-phone-number` Beispiel werden die Anrufernamen für alle angegebenen Telefonnummern aktualisiert.

```
aws chime batch-update-phone-number \
  --update-phone-number-request-items PhoneNumberId=
%2B14013143874,CallingName=phonenumber1 PhoneNumberId=
%2B14013144061,CallingName=phonenumber2
```

Ausgabe:

```
{
  "PhoneNumberErrors": []
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [BatchUpdatePhoneNumber](#) in der AWS CLI Befehlsreferenz.

batch-update-user

Das folgende Codebeispiel zeigt, wie man es benutzt `batch-update-user`.

AWS CLI

Um mehrere Benutzer mit einem einzigen Befehl zu aktualisieren

Das folgende `batch-update-user` Beispiel aktualisiert die `LicenseType` für jeden der aufgelisteten Benutzer im angegebenen Amazon Chime Chime-Konto.

```
aws chime batch-update-user \
  --account-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
  --update-user-request-items "UserId=a1b2c3d4-5678-90ab-
cdef-2222EXAMPLE,LicenseType=Basic" "UserId=a1b2c3d4-5678-90ab-
cdef-3333EXAMPLE,LicenseType=Basic"
```

Ausgabe:

```
{
  "UserErrors": []
}
```

- API-Einheiten finden Sie unter [BatchUpdateUser AWS CLI](#) Befehlsreferenz.

create-account

Das folgende Codebeispiel zeigt, wie man es benutzt `create-account`.

AWS CLI

Um ein Konto zu erstellen

Im folgenden `create-account` Beispiel wird ein Amazon Chime Chime-Konto unter dem AWS Administratorkonto erstellt.

```
aws chime create-account \
  --name MyChimeAccount
```

Ausgabe:

```
{
  "Account": {
    "AwsAccountId": "111122223333",
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "Name": "MyChimeAccount",
    "AccountType": "Team",
    "CreatedTimestamp": "2019-01-04T17:11:22.003Z",
    "DefaultLicense": "Pro",
    "SupportedLicenses": [
      "Basic",
      "Pro"
    ],
    "SigninDelegateGroups": [
      {
        "GroupName": "myGroup"
      }
    ]
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Erste Schritte](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [CreateAccount](#) in der AWS CLI Befehlsreferenz.

create-bot

Das folgende Codebeispiel zeigt, wie man es benutzt `create-bot`.

AWS CLI

So erstellen Sie einen Amazon Chime Chime-Bot

Im folgenden `create-bot` Beispiel wird ein Bot für das angegebene Amazon Chime Enterprise-Konto erstellt.

```
aws chime create-bot \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --display-name "myBot" \
  --domain "example.com"
```

Ausgabe:

```
{
  "Bot": {
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
    "DisplayName": "myBot (Bot)",
    "BotType": "ChatBot",
    "Disabled": false,
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
    "BotEmail": "myBot-chimebot@example.com",
    "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
  }
}
```

Weitere Informationen finden Sie unter [Integrieren eines Chat-Bot mit Amazon Chime](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [CreateBot](#) in der AWS CLI Befehlsreferenz.

create-phone-number-order

Das folgende Codebeispiel zeigt, wie man es benutzt `create-phone-number-order`.

AWS CLI

Um eine Bestellung mit einer Telefonnummer zu erstellen

Im folgenden `create-phone-number-order` Beispiel wird eine Rufnummernreihenfolge für die angegebenen Rufnummern erstellt.

```
aws chime create-phone-number-order \
  --product-type VoiceConnector \
  --e164-phone-numbers " +12065550100" "+12065550101" "+12065550102"
```

Ausgabe:

```
{
  "PhoneNumberOrder": {
    "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
    "ProductType": "VoiceConnector",
    "Status": "Processing",
    "OrderedPhoneNumbers": [
      {
        "E164PhoneNumber": "+12065550100",
        "Status": "Processing"
      },
      {
        "E164PhoneNumber": "+12065550101",
        "Status": "Processing"
      },
      {
        "E164PhoneNumber": "+12065550102",
        "Status": "Processing"
      }
    ],
    "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
    "UpdatedTimestamp": "2019-08-09T21:35:22.408Z"
  }
}
```


Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [CreatePhoneNumberOrder](#) in der AWS CLI Befehlsreferenz.

create-proxy-session

Das folgende Codebeispiel zeigt, wie man es benutzt `create-proxy-session`.

AWS CLI

Um eine Proxysitzung zu erstellen

Im folgenden `create-proxy-session` Beispiel wird eine Proxysitzung mit Sprache und SMS Funktionen erstellt.

```
aws chime create-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --participant-phone-numbers " +14015550101" "+12065550100" \  
  --capabilities "Voice" "SMS"
```

Ausgabe:

```
{  
  "ProxySession": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",  
    "Status": "Open",  
    "ExpiryMinutes": 60,  
    "Capabilities": [  
      "SMS",  
      "Voice"  
    ],  
    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "UpdatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "Participants": [  
      {  
        "PhoneNumber": "+12065550100",  
        "ProxyPhoneNumber": "+19135550199"  
      },  
      {  
        "PhoneNumber": "+14015550101",
```

```

        "ProxyPhoneNumber": "+19135550199"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [CreateProxySession](#) in der AWS CLI Befehlsreferenz.

create-room-membership

Das folgende Codebeispiel zeigt, wie man es benutzt `create-room-membership`.

AWS CLI

Um eine Raummitgliedschaft zu erstellen

Im folgenden `create-room-membership` Beispiel wird der angegebene Benutzer dem Chatroom als Chatroom-Mitglied hinzugefügt.

```

aws chime create-room-membership \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k

```

Ausgabe:

```

{
  "RoomMembership": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Member": {
      "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
      "MemberType": "User",
      "Email": "janed@example.com",
      "FullName": "Jane Doe",
      "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
    },
    "Role": "Member",
    "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
  }
}

```

```

    "UpdatedTimestamp": "2019-12-02T22:36:41.969Z"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateRoomMembership](#) in der AWS CLI Befehlsreferenz.

create-room

Das folgende Codebeispiel zeigt, wie man es benutzt `create-room`.

AWS CLI

Um einen Chatraum zu erstellen

Im folgenden `create-room` Beispiel wird ein Chatroom für das angegebene Amazon Chime Chime-Konto erstellt.

```

aws chime create-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --name chatRoom

```

Ausgabe:

```

{
  "Room": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Name": "chatRoom",
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
    "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateRoom](#) in der AWS CLI Befehlsreferenz.

create-user

Das folgende Codebeispiel zeigt, wie man es benutzt `create-user`.

AWS CLI

Um ein Benutzerprofil für ein gemeinsam genutztes Gerät zu erstellen

Im folgenden `create-user` Beispiel wird ein gemeinsam genutztes Geräteprofil für die angegebene E-Mail-Adresse erstellt.

```
aws chime create-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --email roomdevice@example.com \  
  --user-type SharedDevice
```

Ausgabe:

```
{  
  "User": {  
    "UserId": "1ab2345c-67de-8901-f23g-45h678901j2k",  
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",  
    "PrimaryEmail": "roomdevice@example.com",  
    "DisplayName": "Room Device",  
    "LicenseType": "Pro",  
    "UserType": "SharedDevice",  
    "UserRegistrationStatus": "Registered",  
    "RegisteredOn": "2020-01-15T22:38:09.806Z",  
    "AlexaForBusinessMetadata": {  
      "IsAlexaForBusinessEnabled": false  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Vorbereiten der Installation](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [CreateUser](#) in der AWS CLI Befehlsreferenz.

create-voice-connector-group

Das folgende Codebeispiel zeigt, wie man es benutzt `create-voice-connector-group`.

AWS CLI

So erstellen Sie eine Amazon Chime Voice Connector-Gruppe

Im folgenden `create-voice-connector-group` Beispiel wird eine Amazon Chime Voice Connector-Gruppe erstellt, die den angegebenen Amazon Chime Voice Connector enthält.

```
aws chime create-voice-connector-group \  
  --name myGroup \  
  --voice-connector-items VoiceConnectorId=abcdefghijklmno3pqr4,Priority=2
```

Ausgabe:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "myGroup",  
    "VoiceConnectorItems": [],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [CreateVoiceConnectorGroup](#) in der AWS CLI Befehlsreferenz.

`create-voice-connector`

Das folgende Codebeispiel zeigt, wie man es benutzt `create-voice-connector`.

AWS CLI

So erstellen Sie einen Amazon Chime Voice Connector

Das folgende `create-voice-connector` Beispiel erstellt einen Amazon Chime Voice Connector in der angegebenen AWS Region mit aktivierter Verschlüsselung.

```
aws chime create-voice-connector \  
  --region us-east-1 \  
  --voice-connector-name myConnector \  
  --voice-connector-configuration ConfigurationArn=arn:aws:chime:us-east-1:123456789012:voice-connector-configuration:myConnector
```

```
--name newVoiceConnector \  
--aws-region us-west-2 \  
--require-encryption
```

Ausgabe:

```
{  
  "VoiceConnector": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "AwsRegion": "us-west-2",  
    "Name": "newVoiceConnector",  
    "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",  
    "RequireEncryption": true,  
    "CreatedTimestamp": "2019-09-18T20:34:01.352Z",  
    "UpdatedTimestamp": "2019-09-18T20:34:01.352Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [CreateVoiceConnector](#) in der AWS CLI Befehlsreferenz.

delete-account

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-account`.

AWS CLI

Um ein Konto zu löschen

Im folgenden `delete-account` Beispiel wird das angegebene Konto gelöscht.

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen Ihres Kontos](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteAccount](#) in der AWS CLI Befehlsreferenz.

delete-phone-number

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-phone-number`.

AWS CLI

Um eine Telefonnummer zu löschen

Im folgenden `delete-phone-number` Beispiel wird die angegebene Telefonnummer in die Löschwarteschlange verschoben.

```
aws chime delete-phone-number \  
  --phone-number-id "+12065550100"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeletePhoneNumber](#) in der AWS CLI Befehlsreferenz.

delete-proxy-session

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-proxy-session`.

AWS CLI

Um eine Proxysitzung zu löschen

Im folgenden `delete-proxy-session` Beispiel wird die angegebene Proxysitzung gelöscht.

```
aws chime delete-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk56789l
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [DeleteProxySession](#) in der AWS CLI Befehlsreferenz.

delete-room-membership

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-room-membership`.

AWS CLI

Um einen Benutzer als Mitglied eines Chatrooms zu entfernen

Im folgenden `delete-room-membership` Beispiel wird das angegebene Mitglied aus dem angegebenen Chatroom entfernt.

```
aws chime delete-room-membership \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteRoomMembership](#) in der AWS CLI Befehlsreferenz.

delete-room

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-room`.

AWS CLI

Um einen Chatraum zu löschen

Das folgende `delete-room` Beispiel löscht den angegebenen Chatroom und entfernt die Chatroom-Mitgliedschaften.

```
aws chime delete-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteRoom](#) in der AWS CLI Befehlsreferenz.

delete-voice-connector-group

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector-group`.

AWS CLI

Titel

Das folgende `delete-voice-connector-group` Beispiel löscht die angegebene Amazon Chime Voice Connector-Gruppe.

```
aws chime delete-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteVoiceConnectorGroup](#) in der AWS CLI Befehlsreferenz.

delete-voice-connector-origination

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector-origination`.

AWS CLI

Um die Ursprungseinstellungen zu löschen

Im folgenden `delete-voice-connector-origination` Beispiel werden der ursprüngliche Host, der Port, das Protokoll, die Priorität und das Gewicht aus dem angegebenen Amazon Chime Voice Connector gelöscht.

```
aws chime delete-voice-connector-origination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteVoiceConnectorOrigination](#) in der AWS CLI Befehlsreferenz.

delete-voice-connector-proxy

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector-proxy`.

AWS CLI

Um eine Proxykonfiguration zu löschen

Das folgende `delete-voice-connector-proxy` Beispiel löscht die Proxykonfiguration aus Ihrem Amazon Chime Voice Connector.

```
aws chime delete-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [DeleteVoiceConnectorProxy](#) in der AWS CLI Befehlsreferenz.

delete-voice-connector-streaming-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector-streaming-configuration`.

AWS CLI

Um eine Streaming-Konfiguration zu löschen

Im folgenden `delete-voice-connector-streaming-configuration` Beispiel wird die Streaming-Konfiguration für den angegebenen Amazon Chime Voice Connector gelöscht.

```
aws chime delete-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Streaming von Amazon Chime Voice Connector-Daten nach Kinesis](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteVoiceConnectorStreamingConfiguration](#) in AWS CLI der Befehlsreferenz.

delete-voice-connector-termination-credentials

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector-termination-credentials`.

AWS CLI

Um die Anmeldeinformationen für die Kündigung zu löschen

Im folgenden `delete-voice-connector-termination-credentials` Beispiel werden die Kündigungsdaten für den angegebenen Benutzernamen und Amazon Chime Voice Connector gelöscht.

```
aws chime delete-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --usernames "jdoe"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteVoiceConnectorTerminationCredentials](#) in der AWS CLI Befehlsreferenz.

delete-voice-connector-termination

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector-termination`.

AWS CLI

Um die Einstellungen für die Kündigung zu löschen

Im folgenden `delete-voice-connector-termination` Beispiel werden die Terminierungseinstellungen für den angegebenen Amazon Chime Voice Connector gelöscht.

```
aws chime delete-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteVoiceConnectorTermination](#) in der AWS CLI Befehlsreferenz.

delete-voice-connector

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-voice-connector`.

AWS CLI

So löschen Sie einen Amazon Chime Voice Connector

Das folgende `delete-voice-connector` Beispiel macht das

```
aws chime delete-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteVoiceConnector](#) in der AWS CLI Befehlsreferenz.

disassociate-phone-number-from-user

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-phone-number-from-user`.

AWS CLI

Um die Zuordnung einer Telefonnummer zu einem Benutzer zu trennen

Im folgenden `disassociate-phone-number-from-user` Beispiel wird die Zuordnung einer Telefonnummer zu dem angegebenen Benutzer aufgehoben.

```
aws chime disassociate-phone-number-from-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Benutzertelefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DisassociatePhoneNumberFromUser](#) in der AWS CLI Befehlsreferenz.

`disassociate-phone-numbers-from-voice-connector-group`

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-phone-numbers-from-voice-connector-group`.

AWS CLI

So trennen Sie die Zuordnung von Telefonnummern zu einer Amazon Chime Voice Connector-Gruppe

Das folgende `disassociate-phone-numbers-from-voice-connector-group` Beispiel trennt die Zuordnung der angegebenen Telefonnummern zu einer Amazon Chime Voice Connector-Gruppe.

```
aws chime disassociate-phone-numbers-from-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --e164-phone-numbers "+12065550100" "+12065550101"
```

Ausgabe:

```
{  
  "PhoneNumberErrors": []  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DisassociatePhoneNumbersFromVoiceConnectorGroup](#) in der AWS CLI Befehlsreferenz.

disassociate-phone-numbers-from-voice-connector

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-phone-numbers-from-voice-connector`.

AWS CLI

So trennen Sie Telefonnummern von einem Amazon Chime Voice Connector

Im folgenden `disassociate-phone-numbers-from-voice-connector` Beispiel werden die angegebenen Telefonnummern von einem Amazon Chime Voice Connector getrennt.

```
aws chime disassociate-phone-numbers-from-voice-connector \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --e164-phone-numbers "+12065550100" "+12065550101"
```

Ausgabe:

```
{
  "PhoneNumberErrors": []
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DisassociatePhoneNumbersFromVoiceConnector](#) in der AWS CLI Befehlsreferenz.

disassociate-signin-delegate-groups-from-account

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-signin-delegate-groups-from-account`.

AWS CLI

So trennen Sie die Zuordnung von Anmeldedelegiertengruppen

Im folgenden `disassociate-signin-delegate-groups-from-account` Beispiel wird die Verbindung zwischen der angegebenen Anmeldedelegiertengruppe und dem angegebenen Amazon Chime Chime-Konto getrennt.

```
aws chime disassociate-signin-delegate-groups-from-account \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --group-names "my_users"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerzugriff und Benutzerberechtigungen verwalten](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [DisassociateSigninDelegateGroupsFromAccount](#) in der AWS CLI Befehlsreferenz.

get-account-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-account-settings`.

AWS CLI

Um Einstellungen für ein Konto abzurufen

Im folgenden `get-account-settings` Beispiel werden die Kontoeinstellungen für das angegebene Konto abgerufen.

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{  
  "AccountSettings": {  
    "DisableRemoteControl": false,  
    "EnableDialOut": false  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Amazon Chime Chime-Konten im Amazon Chime](#) Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetAccountSettings](#) in der AWS CLI Befehlsreferenz.

get-account

Das folgende Codebeispiel zeigt, wie man es benutzt `get-account`.

AWS CLI

Um die Details für ein Konto abzurufen

Im folgenden `get-account` Beispiel werden die Details für das angegebene Amazon Chime Chime-Konto abgerufen.

```
aws chime get-account \
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{
  "Account": {
    "AwsAccountId": "111122223333",
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "Name": "EnterpriseDirectory",
    "AccountType": "EnterpriseDirectory",
    "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
    "DefaultLicense": "Pro",
    "SupportedLicenses": [
      "Basic",
      "Pro"
    ],
    "SigninDelegateGroups": [
      {
        "GroupName": "myGroup"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Amazon Chime Chime-Konten im Amazon Chime Chime-Administratorhandbuch](#).

- API-Einheiten finden Sie [GetAccount](#) in der AWS CLI Befehlsreferenz.

get-bot

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bot`.

AWS CLI

Um Details über einen Bot abzurufen

Im folgenden `get-bot` Beispiel werden die Details für den angegebenen Bot angezeigt.

```
aws chime get-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Ausgabe:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": false,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"  
  }  
}
```

Weitere Informationen finden Sie unter [Chat-Bots aktualisieren](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [GetBot](#) in der AWS CLI Befehlsreferenz.

get-global-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-global-settings`.

AWS CLI

Um globale Einstellungen abzurufen

Im folgenden `get-global-settings` Beispiel werden die S3-Bucket-Namen abgerufen, die zum Speichern von Anruferdetaildatensätzen für Amazon Chime Business Calling und Amazon Chime Voice Connectors verwendet werden, die dem Administratorkonto zugeordnet sind. AWS

```
aws chime get-global-settings
```

Ausgabe:

```
{
  "BusinessCalling": {
    "CdrBucket": "s3bucket"
  },
  "VoiceConnector": {
    "CdrBucket": "s3bucket"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung globaler Einstellungen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetGlobalSettings](#) in der AWS CLI Befehlsreferenz.

get-phone-number-order

Das folgende Codebeispiel zeigt, wie man es benutzt `get-phone-number-order`.

AWS CLI

Um Details für eine Bestellung mit einer Telefonnummer zu erhalten

Im folgenden `get-phone-number-order` Beispiel werden die Details der angegebenen Rufnummernbestellung angezeigt.

```
aws chime get-phone-number-order \
  --phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

Ausgabe:

```
{
  "PhoneNumberOrder": {
    "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
```

```
"ProductType": "VoiceConnector",
"Status": "Partial",
"OrderedPhoneNumbers": [
  {
    "E164PhoneNumber": "+12065550100",
    "Status": "Acquired"
  },
  {
    "E164PhoneNumber": "+12065550101",
    "Status": "Acquired"
  },
  {
    "E164PhoneNumber": "+12065550102",
    "Status": "Failed"
  }
],
"CreatedTimestamp": "2019-08-09T21:35:21.427Z",
"UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetPhoneNumberOrder](#) in der AWS CLI Befehlsreferenz.

get-phone-number-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-phone-number-settings`.

AWS CLI

Um den Namen eines ausgehenden Anrufs abzurufen

Im folgenden `get-phone-number-settings` Beispiel wird der Standardname für ausgehende Anrufe für das Konto des anrufenden Benutzers abgerufen. AWS

```
aws chime get-phone-number-settings
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
```

```
"CallingName": "myName",
"CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetPhoneNumberSettings](#) in der AWS CLI Befehlsreferenz.

get-phone-number

Das folgende Codebeispiel zeigt, wie man es benutzt `get-phone-number`.

AWS CLI

Um Details zur Telefonnummer zu erhalten

Im folgenden `get-phone-number` Beispiel werden die Details der angegebenen Telefonnummer angezeigt.

```
aws chime get-phone-number \
  --phone-number-id +12065550100
```

Ausgabe:

```
{
  "PhoneNumber": {
    "PhoneNumberId": "%2B12065550100",
    "E164PhoneNumber": "+12065550100",
    "Type": "Local",
    "ProductType": "VoiceConnector",
    "Status": "Unassigned",
    "Capabilities": {
      "InboundCall": true,
      "OutboundCall": true,
      "InboundSMS": true,
      "OutboundSMS": true,
      "InboundMMS": true,
      "OutboundMMS": true
    },
    "Associations": [
      {
```

```

        "Value": "abcdef1ghij2klmno3pqr4",
        "Name": "VoiceConnectorId",
        "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
      }
    ],
    "CallingNameStatus": "UpdateInProgress",
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
    "UpdatedTimestamp": "2019-08-09T21:35:31.745Z"
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetPhoneNumber](#) in der AWS CLI Befehlsreferenz.

get-proxy-session

Das folgende Codebeispiel zeigt, wie man es benutzt `get-proxy-session`.

AWS CLI

Um Details zur Proxy-Sitzung abzurufen

Das folgende `get-proxy-session` Beispiel listet die Details der angegebenen Proxysitzung auf.

```

aws chime get-proxy-session \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891

```

Ausgabe:

```

{
  "ProxySession": {
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",
    "Status": "Open",
    "ExpiryMinutes": 60,
    "Capabilities": [
      "SMS",
      "Voice"
    ],
  },
}

```

```

    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",
    "UpdatedTimestamp": "2020-04-15T16:10:10.288Z",
    "Participants": [
      {
        "PhoneNumber": "+12065550100",
        "ProxyPhoneNumber": "+19135550199"
      },
      {
        "PhoneNumber": "+14015550101",
        "ProxyPhoneNumber": "+19135550199"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [GetProxySession](#) in der AWS CLI Befehlsreferenz.

get-room

Das folgende Codebeispiel zeigt, wie man es benutzt `get-room`.

AWS CLI

Um die Details zu einem Chatroom zu erhalten

Im folgenden `get-room` Beispiel werden Details zum angegebenen Chatroom angezeigt.

```

aws chime get-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j

```

Ausgabe:

```

{
  "Room": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Name": "chatRoom",
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
  }
}

```

```
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",  
    "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [GetRoom](#) in der AWS CLI Befehlsreferenz.

get-user-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-user-settings`.

AWS CLI

Um Benutzereinstellungen abzurufen

Im folgenden `get-user-settings` Beispiel werden die angegebenen Benutzereinstellungen angezeigt.

```
aws chime get-user-settings \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Ausgabe:

```
{  
  "UserSettings": {  
    "Telephony": {  
      "InboundCalling": true,  
      "OutboundCalling": true,  
      "SMS": true  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Benutzertelefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetUserSettings](#) in der AWS CLI Befehlsreferenz.

get-user

Das folgende Codebeispiel zeigt, wie man es benutzt `get-user`.

AWS CLI

Um Details über einen Benutzer abzurufen

Im folgenden `get-user` Beispiel werden die Details für den angegebenen Benutzer abgerufen.

```
aws chime get-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Ausgabe:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "PrimaryEmail": "marthar@example.com",  
    "DisplayName": "Martha Rivera",  
    "LicenseType": "Pro",  
    "UserRegistrationStatus": "Registered",  
    "RegisteredOn": "2018-12-20T18:45:25.231Z",  
    "InvitedOn": "2018-12-20T18:45:25.231Z",  
    "AlexaForBusinessMetadata": {  
      "IsAlexaForBusinessEnabled": false,  
      "AlexaForBusinessRoomArn": "null"  
    },  
    "PersonalPIN": "XXXXXXXXXX"  
  }  
}
```

Weitere Informationen finden Sie unter [Benutzer verwalten](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetUser](#) in der AWS CLI Befehlsreferenz.

get-voice-connector-group

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-group`.

AWS CLI

Um Details für eine Amazon Chime Voice Connector-Gruppe abzurufen

Im folgenden `get-voice-connector-group` Beispiel werden Details für die angegebene Amazon Chime Voice Connector-Gruppe angezeigt.

```
aws chime get-voice-connector-group \  
--voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

Ausgabe:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "myGroup",  
    "VoiceConnectorItems": [],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnectorGroup](#) in der AWS CLI Befehlsreferenz.

get-voice-connector-logging-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-logging-configuration`.

AWS CLI

Um Details zur Protokollierungskonfiguration abzurufen

Im folgenden `get-voice-connector-logging-configuration` Beispiel werden die Details der Protokollierungskonfiguration für den angegebenen Amazon Chime Voice Connector abgerufen.

```
aws chime get-voice-connector-logging-configuration \  

```

```
--voice-connector-id abcdef1ghij2klmno3pqr4
```

Ausgabe:

```
{
  "LoggingConfiguration": {
    "EnableSIPLogs": true
  }
}
```

Weitere Informationen finden Sie unter [Streaming von Amazon Chime Voice Connector-Medien nach Kinesis](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnectorLoggingConfiguration](#) in AWS CLI der Befehlsreferenz.

get-voice-connector-origination

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-origination`.

AWS CLI

Um die Ursprungseinstellungen abzurufen

Im folgenden `get-voice-connector-origination` Beispiel werden der ursprüngliche Host, der Port, das Protokoll, die Priorität und das Gewicht für den angegebenen Amazon Chime Voice Connector abgerufen.

```
aws chime get-voice-connector-origination \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Ausgabe:

```
{
  "Origination": {
    "Routes": [
      {
        "Host": "10.24.34.0",
        "Port": 1234,
        "Protocol": "TCP",
```

```
        "Priority": 1,
        "Weight": 5
      }
    ],
    "Disabled": false
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnectorOrigination](#) in der AWS CLI Befehlsreferenz.

get-voice-connector-proxy

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-proxy`.

AWS CLI

Um Details zur Proxy-Konfiguration abzurufen

Im folgenden `get-voice-connector-proxy` Beispiel werden die Proxy-Konfigurationsdetails für Ihren Amazon Chime Voice Connector abgerufen.

```
aws chime get-voice-connector-proxy \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Ausgabe:

```
{
  "Proxy": {
    "DefaultSessionExpiryMinutes": 60,
    "Disabled": false,
    "PhoneNumberCountries": [
      "US"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [GetVoiceConnectorProxy](#) in der AWS CLI Befehlsreferenz.

get-voice-connector-streaming-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-streaming-configuration`.

AWS CLI

Um Details zur Streaming-Konfiguration abzurufen

Im folgenden `get-voice-connector-streaming-configuration` Beispiel werden die Streaming-Konfigurationsdetails für den angegebenen Amazon Chime Voice Connector abgerufen.

```
aws chime get-voice-connector-streaming-configuration \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Ausgabe:

```
{
  "StreamingConfiguration": {
    "DataRetentionInHours": 24,
    "Disabled": false
  }
}
```

Weitere Informationen finden Sie unter [Streaming von Amazon Chime Voice Connector-Daten nach Kinesis](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnectorStreamingConfiguration](#) in AWS CLI der Befehlsreferenz.

get-voice-connector-termination-health

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-termination-health`.

AWS CLI

Um Informationen zum Status der Kündigung abzurufen

Im folgenden `get-voice-connector-termination-health` Beispiel werden die Informationen zum Status der Kündigung für den angegebenen Amazon Chime Voice Connector abgerufen.

```
aws chime get-voice-connector-termination-health \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Ausgabe:

```
{  
  "TerminationHealth": {  
    "Timestamp": "Fri Aug 23 16:45:55 UTC 2019",  
    "Source": "10.24.34.0"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnectorTerminationHealth](#) in der AWS CLI Befehlsreferenz.

get-voice-connector-termination

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector-termination`.

AWS CLI

Um die Terminierungseinstellungen abzurufen

Im folgenden `get-voice-connector-termination` Beispiel werden die Terminierungseinstellungen für den angegebenen Amazon Chime Voice Connector abgerufen.

```
aws chime get-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{  
  "Termination": {
```

```
    "CpsLimit": 1,
    "DefaultPhoneNumber": "+12065550100",
    "CallingRegions": [
      "US"
    ],
    "CidrAllowedList": [
      "10.24.34.0/23"
    ],
    "Disabled": false
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnectorTermination](#) in der AWS CLI Befehlsreferenz.

get-voice-connector

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-connector`.

AWS CLI

So rufen Sie Details für einen Amazon Chime Voice Connector ab

Im folgenden `get-voice-connector` Beispiel werden die Details des angegebenen Amazon Chime Voice Connectors angezeigt.

```
aws chime get-voice-connector \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Ausgabe:

```
{
  "VoiceConnector": {
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
    "AwsRegion": "us-west-2",
    "Name": "newVoiceConnector",
    "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",
    "RequireEncryption": true,
    "CreatedTimestamp": "2019-09-18T20:34:01.352Z",
    "UpdatedTimestamp": "2019-09-18T20:34:01.352Z"
  }
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [GetVoiceConnector](#) in der AWS CLI Befehlsreferenz.

invite-users

Das folgende Codebeispiel zeigt, wie man es benutzt `invite-users`.

AWS CLI

So laden Sie Benutzer ein, Amazon Chime beizutreten

Im folgenden `invite-users` Beispiel wird eine E-Mail gesendet, um einen Benutzer zu dem angegebenen Amazon Chime Chime-Konto einzuladen.

```
aws chime invite-users \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-email-list "alejandror@example.com" "janed@example.com"
```

Ausgabe:

```
{  
  "Invites": [  
    {  
      "InviteId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
      "Status": "Pending",  
      "EmailAddress": "alejandror@example.com",  
      "EmailStatus": "Sent"  
    },  
    {  
      "InviteId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
      "Status": "Pending",  
      "EmailAddress": "janed@example.com",  
      "EmailStatus": "Sent"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Benutzer einladen und sperren](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [InviteUsers](#) in der AWS CLI Befehlsreferenz.

list-accounts

Das folgende Codebeispiel zeigt, wie man es benutzt `list-accounts`.

AWS CLI

Um eine Liste von Konten zu erhalten

Im folgenden `list-accounts` Beispiel wird eine Liste der Amazon Chime Chime-Konten im Administratorkonto abgerufen. AWS

```
aws chime list-accounts
```

Ausgabe:

```
{
  "Accounts": [
    {
      "AwsAccountId": "111122223333",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "Name": "First Chime Account",
      "AccountType": "EnterpriseDirectory",
      "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
      "DefaultLicense": "Pro",
      "SupportedLicenses": [
        "Basic",
        "Pro"
      ],
      "SigninDelegateGroups": [
        {
          "GroupName": "myGroup"
        }
      ]
    },
    {
      "AwsAccountId": "111122223333",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
```



```

    "Name": "Second Chime Account",
    "AccountType": "Team",
    "CreatedTimestamp": "2018-09-04T21:44:22.292Z",
    "DefaultLicense": "Pro",
    "SupportedLicenses": [
      "Basic",
      "Pro"
    ],
    "SigninDelegateGroups": [
      {
        "GroupName": "myGroup"
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Amazon Chime Chime-Konten im Amazon Chime Chime-Administratorhandbuch](#).

- API-Einheiten finden Sie [ListAccounts](#) in der AWS CLI Befehlsreferenz.

list-bots

Das folgende Codebeispiel zeigt, wie man es benutzt `list-bots`.

AWS CLI

Um eine Liste von Bots abzurufen

Das folgende `list-bots` Beispiel listet die Bots auf, die mit dem angegebenen Amazon Chime Enterprise-Konto verknüpft sind.

```

aws chime list-bots \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45

```

Ausgabe:

```

{
  "Bot": {
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
  }
}

```

```
"DisplayName": "myBot (Bot)",
"BotType": "ChatBot",
"Disabled": false,
"CreatedTimestamp": "2019-09-09T18:05:56.749Z",
"UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
"BotEmail": "myBot-chimebot@example.com",
"SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
}
}
```

Weitere Informationen finden Sie unter [Verwenden von Chat-Bots mit Amazon Chime](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [ListBots](#) in der AWS CLI Befehlsreferenz.

list-phone-number-orders

Das folgende Codebeispiel zeigt, wie man es benutzt `list-phone-number-orders`.

AWS CLI

Um Bestellungen mit Telefonnummern aufzulisten

Das folgende `list-phone-number-orders` Beispiel listet die Telefonnummernbestellungen auf, die mit dem Konto des Amazon Chime Chime-Administrators verknüpft sind.

```
aws chime list-phone-number-orders
```

Ausgabe:

```
{
  "PhoneNumberOrders": [
    {
      "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
      "ProductType": "VoiceConnector",
      "Status": "Partial",
      "OrderedPhoneNumbers": [
        {
          "E164PhoneNumber": "+12065550100",
          "Status": "Acquired"
        },
        {
```

```

        "E164PhoneNumber": "+12065550101",
        "Status": "Acquired"
    },
    {
        "E164PhoneNumber": "+12065550102",
        "Status": "Failed"
    }
],
"CreatedTimestamp": "2019-08-09T21:35:21.427Z",
"UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
}
{
    "PhoneNumberOrderId": "cba54321-ed76-09f5-321g-h54i876j2109",
    "ProductType": "BusinessCalling",
    "Status": "Partial",
    "OrderedPhoneNumbers": [
        {
            "E164PhoneNumber": "+12065550103",
            "Status": "Acquired"
        },
        {
            "E164PhoneNumber": "+12065550104",
            "Status": "Acquired"
        },
        {
            "E164PhoneNumber": "+12065550105",
            "Status": "Failed"
        }
    ],
    "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
    "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
}
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ListPhoneNumberOrders](#) in der AWS CLI Befehlsreferenz.

list-phone-numbers

Das folgende Codebeispiel zeigt, wie man es benutzt `list-phone-numbers`.

AWS CLI

Um Telefonnummern für ein Amazon Chime Chime-Konto aufzulisten

Das folgende `list-phone-numbers` Beispiel listet die Telefonnummern auf, die mit dem Amazon Chime Chime-Konto des Administrators verknüpft sind.

```
aws chime list-phone-numbers
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "PhoneNumbers": [
    {
      "PhoneNumberId": "%2B12065550100",
      "E164PhoneNumber": "+12065550100",
      "Type": "Local",
      "ProductType": "VoiceConnector",
      "Status": "Assigned",
      "Capabilities": {
        "InboundCall": true,
        "OutboundCall": true,
        "InboundSMS": true,
        "OutboundSMS": true,
        "InboundMMS": true,
        "OutboundMMS": true
      },
      "Associations": [
        {
          "Value": "abcdef1ghij2klmno3pqr4",
          "Name": "VoiceConnectorId",
          "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
        }
      ],
      "CallingNameStatus": "UpdateInProgress",
      "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
      "UpdatedTimestamp": "2019-10-28T18:42:07.964Z"
    },
    {
      "PhoneNumberId": "%2B12065550101",
      "E164PhoneNumber": "+12065550101",
      "Type": "Local",
      "ProductType": "VoiceConnector",
```

```

    "Status": "Assigned",
    "Capabilities": {
      "InboundCall": true,
      "OutboundCall": true,
      "InboundSMS": true,
      "OutboundSMS": true,
      "InboundMMS": true,
      "OutboundMMS": true
    },
    "Associations": [
      {
        "Value": "abcdef1ghij2klmno3pqr4",
        "Name": "VoiceConnectorId",
        "AssociatedTimestamp": "2019-10-28T18:40:37.511Z"
      }
    ],
    "CallingNameStatus": "UpdateInProgress",
    "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
    "UpdatedTimestamp": "2019-10-28T18:42:07.960Z"
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ListPhoneNumbers](#) in der AWS CLI Befehlsreferenz.

list-proxy-sessions

Das folgende Codebeispiel zeigt die Verwendung `list-proxy-sessions`.

AWS CLI

Um Proxy-Sitzungen aufzulisten

Das folgende `list-proxy-sessions` Beispiel listet die Proxy-Sitzungen für Ihren Amazon Chime Voice Connector auf.

```

aws chime list-proxy-sessions \
  --voice-connector-id abcdef1ghij2klmno3pqr4

```

Ausgabe:

```
{
  "ProxySession": {
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",
    "Status": "Open",
    "ExpiryMinutes": 60,
    "Capabilities": [
      "SMS",
      "Voice"
    ],
    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",
    "UpdatedTimestamp": "2020-04-15T16:10:10.288Z",
    "Participants": [
      {
        "PhoneNumber": "+12065550100",
        "ProxyPhoneNumber": "+19135550199"
      },
      {
        "PhoneNumber": "+14015550101",
        "ProxyPhoneNumber": "+19135550199"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [ListProxySessions](#) in der AWS CLI Befehlsreferenz.

list-room-memberships

Das folgende Codebeispiel zeigt die Verwendung `list-room-memberships`.

AWS CLI

Um Raummitgliedschaften aufzulisten

Im folgenden `list-room-memberships` Beispiel wird eine Liste der Mitgliedschaftsdetails für den angegebenen Chatroom angezeigt.

```
aws chime list-room-memberships \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
```

```
--room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Ausgabe:

```
{
  "RoomMemberships": [
    {
      "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
      "Member": {
        "MemberId": "2ab2345c-67de-8901-f23g-45h678901j2k",
        "MemberType": "User",
        "Email": "zhangw@example.com",
        "FullName": "Zhang Wei",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
      },
      "Role": "Member",
      "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
      "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
    },
    {
      "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
      "Member": {
        "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
        "MemberType": "User",
        "Email": "janed@example.com",
        "FullName": "Jane Doe",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
      },
      "Role": "Administrator",
      "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
      "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRoomMemberships](#) in der AWS CLI Befehlsreferenz.

list-rooms

Das folgende Codebeispiel zeigt die Verwendung `list-rooms`.

AWS CLI

Um Chatrooms aufzulisten

Im folgenden `list-rooms` Beispiel wird eine Liste der Chatrooms im angegebenen Konto angezeigt. Die Liste wird nur nach den Chatrooms gefiltert, denen das angegebene Mitglied angehört.

```
aws chime list-rooms \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Ausgabe:

```
{
  "Room": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Name": "teamRoom",
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
    "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRooms](#) in der AWS CLI Befehlsreferenz.

list-users

Das folgende Codebeispiel zeigt die Verwendung `list-users`.

AWS CLI

Um die Benutzer in einem Konto aufzulisten

Das folgende `list-users` Beispiel listet die Benutzer für das angegebene Amazon Chime Chime-Konto auf.

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```


Ausgabe:

```
{
  "Users": [
    {
      "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "PrimaryEmail": "mariag@example.com",
      "DisplayName": "Maria Garcia",
      "LicenseType": "Pro",
      "UserType": "PrivateUser",
      "UserRegistrationStatus": "Registered",
      "RegisteredOn": "2018-12-20T18:45:25.231Z"
      "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
      }
    },
    {
      "UserId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "PrimaryEmail": "richardr@example.com",
      "DisplayName": "Richard Roe",
      "LicenseType": "Pro",
      "UserType": "PrivateUser",
      "UserRegistrationStatus": "Registered",
      "RegisteredOn": "2018-12-20T18:45:45.415Z"
      "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
      }
    },
    {
      "UserId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "PrimaryEmail": "saanvis@example.com",
      "DisplayName": "Saanvi Sarkar",
      "LicenseType": "Basic",
      "UserType": "PrivateUser",
      "UserRegistrationStatus": "Registered",
      "RegisteredOn": "2018-12-20T18:46:57.747Z"
      "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
      }
    },
    {
```

```

    "UserId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "PrimaryEmail": "wxiulan@example.com",
    "DisplayName": "Wang Xiulan",
    "LicenseType": "Basic",
    "UserType": "PrivateUser",
    "UserRegistrationStatus": "Registered",
    "RegisteredOn": "2018-12-20T18:47:15.390Z"
    "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
    }
  }
]
}

```

Weitere Informationen finden Sie unter [Benutzer verwalten](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ListUsers](#) in der AWS CLI Befehlsreferenz.

list-voice-connector-groups

Das folgende Codebeispiel zeigt die Verwendung `list-voice-connector-groups`.

AWS CLI

Um Amazon Chime Voice Connector-Gruppen für ein Amazon Chime Chime-Konto aufzulisten

Das folgende `list-voice-connector-groups` Beispiel listet die Amazon Chime Voice Connector-Gruppen auf, die dem Amazon Chime Chime-Konto des Administrators zugeordnet sind.

```
aws chime list-voice-connector-groups
```

Ausgabe:

```

{
  "VoiceConnectorGroups": [
    {
      "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl18901",
      "Name": "myGroup",
      "VoiceConnectorItems": [],
      "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
    }
  ]
}

```

```
        "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ListVoiceConnectorGroups](#) in der AWS CLI Befehlsreferenz.

list-voice-connector-termination-credentials

Das folgende Codebeispiel zeigt die Verwendung `list-voice-connector-termination-credentials`.

AWS CLI

Um eine Liste mit Kündigungsdaten abzurufen

Im folgenden `list-voice-connector-termination-credentials` Beispiel wird eine Liste der Kündigungsdaten für den angegebenen Amazon Chime Voice Connector abgerufen.

```
aws chime list-voice-connector-termination-credentials \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "Usernames": [
    "jdoe"
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ListVoiceConnectorTerminationCredentials](#) in der AWS CLI Befehlsreferenz.

list-voice-connectors

Das folgende Codebeispiel zeigt die Verwendung `list-voice-connectors`.

AWS CLI

Um Amazon Chime Voice Connectors für ein Konto aufzulisten

Das folgende `list-voice-connectors` Beispiel listet die Amazon Chime Voice Connectors auf, die dem Konto des Anrufers zugeordnet sind.

```
aws chime list-voice-connectors
```

Ausgabe:

```
{
  "VoiceConnectors": [
    {
      "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
      "AwsRegion": "us-east-1",
      "Name": "MyVoiceConnector",
      "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",
      "RequireEncryption": true,
      "CreatedTimestamp": "2019-06-04T18:46:56.508Z",
      "UpdatedTimestamp": "2019-09-18T16:33:00.806Z"
    },
    {
      "VoiceConnectorId": "cbadef1ghij2klmno3pqr5",
      "AwsRegion": "us-west-2",
      "Name": "newVoiceConnector",
      "OutboundHostName": "cbadef1ghij2klmno3pqr5.voiceconnector.chime.aws",
      "RequireEncryption": true,
      "CreatedTimestamp": "2019-09-18T20:34:01.352Z",
      "UpdatedTimestamp": "2019-09-18T20:34:01.352Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ListVoiceConnectors](#) in der AWS CLI Befehlsreferenz.

logout-user

Das folgende Codebeispiel zeigt die Verwendung `logout-user`.

AWS CLI

Um einen Benutzer abzumelden

Im folgenden `logout-user` Beispiel wird der angegebene Benutzer abgemeldet.

```
aws chime logout-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [LogoutUser](#) in der AWS CLI Befehlsreferenz.

put-voice-connector-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-voice-connector-logging-configuration`.

AWS CLI

So fügen Sie eine Protokollierungskonfiguration für einen Amazon Chime Voice Connector hinzu

Das folgende `put-voice-connector-logging-configuration` Beispiel aktiviert die SIP Protokollierungskonfiguration für den angegebenen Amazon Chime Voice Connector.

```
aws chime put-voice-connector-logging-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --logging-configuration EnableSIPLogs=true
```

Ausgabe:

```
{  
  "LoggingConfiguration": {  
    "EnableSIPLogs": true  
  }  
}
```

Weitere Informationen finden Sie unter [Streaming von Amazon Chime Voice Connector-Medien nach Kinesis](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [PutVoiceConnectorLoggingConfiguration](#) in AWS CLI der Befehlsreferenz.

put-voice-connector-origination

Das folgende Codebeispiel zeigt die Verwendung `put-voice-connector-origination`.

AWS CLI

So richten Sie die Ursprungseinstellungen ein

Im folgenden `put-voice-connector-origination` Beispiel werden der Ursprungshost, der Port, das Protokoll, die Priorität und die Gewichtung für den angegebenen Amazon Chime Voice Connector eingerichtet.

```
aws chime put-voice-connector-origination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --origination  
  Routes=[{Host="10.24.34.0",Port=1234,Protocol="TCP",Priority=1,Weight=5}],Disabled=false
```

Ausgabe:

```
{  
  "Origination": {  
    "Routes": [  
      {  
        "Host": "10.24.34.0",  
        "Port": 1234,  
        "Protocol": "TCP",  
        "Priority": 1,  
        "Weight": 5  
      }  
    ],  
    "Disabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [PutVoiceConnectorOrigination](#) in der AWS CLI Befehlsreferenz.

put-voice-connector-proxy

Das folgende Codebeispiel zeigt die Verwendung `put-voice-connector-proxy`.

AWS CLI

Um eine Proxykonfiguration einzurichten

Das folgende `put-voice-connector-proxy` Beispiel legt eine Proxykonfiguration für Ihren Amazon Chime Voice Connector fest.

```
aws chime put-voice-connector-proxy \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --default-session-expiry-minutes 60 \
  --phone-number-pool-countries "US"
```

Ausgabe:

```
{
  "Proxy": {
    "DefaultSessionExpiryMinutes": 60,
    "Disabled": false,
    "PhoneNumberCountries": [
      "US"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [PutVoiceConnectorProxy](#) in der AWS CLI Befehlsreferenz.

put-voice-connector-streaming-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-voice-connector-streaming-configuration`.

AWS CLI

Um eine Streaming-Konfiguration zu erstellen

Das folgende `put-voice-connector-streaming-configuration` Beispiel erstellt eine Streaming-Konfiguration für den angegebenen Amazon Chime Voice Connector. Es ermöglicht Medienstreaming vom Amazon Chime Voice Connector zu Amazon Kinesis und legt die Datenaufbewahrungsdauer auf 24 Stunden fest.

```
aws chime put-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --streaming-configuration DataRetentionInHours=24,Disabled=false
```

Ausgabe:

```
{  
  "StreamingConfiguration": {  
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [Streaming von Amazon Chime Voice Connector-Daten nach Kinesis](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [PutVoiceConnectorStreamingConfiguration](#) in AWS CLI der Befehlsreferenz.

put-voice-connector-termination-credentials

Das folgende Codebeispiel zeigt die Verwendung `put-voice-connector-termination-credentials`.

AWS CLI

Um Anmeldeinformationen für die Kündigung einzurichten

Im folgenden `put-voice-connector-termination-credentials` Beispiel werden die Kündigungsinformationen für den angegebenen Amazon Chime Voice Connector festgelegt.

```
aws chime put-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --credentials Username="jdoe",Password="XXXXXXXXX"
```


Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [PutVoiceConnectorTerminationCredentials](#) in der AWS CLI Befehlsreferenz.

put-voice-connector-termination

Das folgende Codebeispiel zeigt die Verwendung `put-voice-connector-termination`.

AWS CLI

Um die Einstellungen für die Kündigung einzurichten

Im folgenden `put-voice-connector-termination` Beispiel werden die Anrufregionen und die zulässigen IP-Host-Terminierungseinstellungen für den angegebenen Amazon Chime Voice Connector festgelegt.

```
aws chime put-voice-connector-termination \  
  --voice-connector-id abcdefghijklmno3pqr4 \  
  --termination CallingRegions="US",CidrAllowedList="10.24.34.0/23",Disabled=false
```

Ausgabe:

```
{  
  "Termination": {  
    "CpsLimit": 0,  
    "CallingRegions": [  
      "US"  
    ],  
    "CidrAllowedList": [  
      "10.24.34.0/23"  
    ],  
    "Disabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [PutVoiceConnectorTermination](#) in der AWS CLI Befehlsreferenz.

regenerate-security-token

Das folgende Codebeispiel zeigt die Verwendung `regenerate-security-token`.

AWS CLI

Um ein Sicherheitstoken neu zu generieren

Im folgenden `regenerate-security-token` Beispiel wird das Sicherheitstoken für den angegebenen Bot regeneriert.

```
aws chime regenerate-security-token \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Ausgabe:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": false,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"  
  }  
}
```

Weitere Informationen finden Sie unter [Chat-Bot-Anfragen authentifizieren](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [RegenerateSecurityToken](#) in der AWS CLI Befehlsreferenz.

reset-personal-pin

Das folgende Codebeispiel zeigt die Verwendung `reset-personal-pin`.

AWS CLI

Um das persönliche Meeting eines Benutzers zurückzusetzen PIN

Im folgenden `reset-personal-pin` Beispiel wird die persönliche Besprechung PIN des angegebenen Benutzers zurückgesetzt.

```
aws chime reset-personal-pin \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Ausgabe:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "PrimaryEmail": "mateo@example.com",  
    "DisplayName": "Mateo Jackson",  
    "LicenseType": "Pro",  
    "UserType": "PrivateUser",  
    "UserRegistrationStatus": "Registered",  
    "RegisteredOn": "2018-12-20T18:45:25.231Z",  
    "AlexaForBusinessMetadata": {  
      "IsAlexaForBusinessEnabled": False,  
      "AlexaForBusinessRoomArn": "null"  
    },  
    "PersonalPIN": "XXXXXXXXXX"  
  }  
}
```

Weitere Informationen finden Sie unter [Persönliche Besprechung ändern PINs](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [ResetPersonalPin](#) in der AWS CLI Befehlsreferenz.

restore-phone-number

Das folgende Codebeispiel zeigt die Verwendung `restore-phone-number`.

AWS CLI

Um eine Telefonnummer wiederherzustellen

Im folgenden `restore-phone-number` Beispiel wird die angegebene Telefonnummer aus der Löschwarteschlange wiederhergestellt.

```
aws chime restore-phone-number \
  --phone-number-id "+12065550100"
```

Ausgabe:

```
{
  "PhoneNumber": {
    "PhoneNumberId": "%2B12065550100",
    "E164PhoneNumber": "+12065550100",
    "Type": "Local",
    "ProductType": "BusinessCalling",
    "Status": "Unassigned",
    "Capabilities": {
      "InboundCall": true,
      "OutboundCall": true,
      "InboundSMS": true,
      "OutboundSMS": true,
      "InboundMMS": true,
      "OutboundMMS": true
    },
    "Associations": [],
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
    "UpdatedTimestamp": "2019-08-12T22:06:36.355Z"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [RestorePhoneNumber](#) in der AWS CLI Befehlsreferenz.

search-available-phone-numbers

Das folgende Codebeispiel zeigt die Verwendung `search-available-phone-numbers`.

AWS CLI

Um nach verfügbaren Telefonnummern zu suchen

Im folgenden `search-available-phone-numbers` Beispiel werden verfügbare Telefonnummern anhand der Vorwahl durchsucht.

```
aws chime search-available-phone-numbers \  
  --area-code "206"
```

Ausgabe:

```
{  
  "E164PhoneNumbers": [  
    "+12065550100",  
    "+12065550101",  
    "+12065550102",  
    "+12065550103",  
    "+12065550104",  
    "+12065550105",  
    "+12065550106",  
    "+12065550107",  
    "+12065550108",  
    "+12065550109",  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [SearchAvailablePhoneNumbers](#) in der AWS CLI Befehlsreferenz.

update-account-settings

Das folgende Codebeispiel zeigt die Verwendung `update-account-settings`.

AWS CLI

Um die Einstellungen für Ihr Konto zu aktualisieren

Das folgende `update-account-settings` Beispiel deaktiviert die Fernsteuerung von geteilten Bildschirmen für das angegebene Amazon Chime Chime-Konto.

```
aws chime update-account-settings \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --enable-remote-control false
```

```
--account-settings DisableRemoteControl=true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [UpdateAccountSettings AWS CLI-Befehlsreferenz](#).

update-account

Das folgende Codebeispiel zeigt die Verwendung `update-account`.

AWS CLI

Um ein Konto zu aktualisieren

Im folgenden `update-account` Beispiel wird der angegebene Kontoname aktualisiert.

```
aws chime update-account \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --name MyAccountName
```

Ausgabe:

```
{  
  "Account": {  
    "AwsAccountId": "111122223333",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "Name": "MyAccountName",  
    "AccountType": "Team",  
    "CreatedTimestamp": "2018-09-04T21:44:22.292Z",  
    "DefaultLicense": "Pro",  
    "SupportedLicenses": [  
      "Basic",  
      "Pro"  
    ],  
    "SigninDelegateGroups": [  
      {  
        "GroupName": "myGroup"  
      },  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Umbenennen Ihres Kontos](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateAccount](#) in der AWS CLI Befehlsreferenz.

update-bot

Das folgende Codebeispiel zeigt die Verwendung `update-bot`.

AWS CLI

Um einen Bot zu aktualisieren

Im folgenden `update-bot` Beispiel wird der Status des angegebenen Bots aktualisiert, sodass er nicht mehr ausgeführt wird.

```
aws chime update-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \  
  --disabled
```

Ausgabe:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": true,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"  
  }  
}
```

Weitere Informationen finden Sie unter [Chat-Bots aktualisieren](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [UpdateBot](#) in der AWS CLI Befehlsreferenz.

update-global-settings

Das folgende Codebeispiel zeigt die Verwendung `update-global-settings`.

AWS CLI

Um globale Einstellungen zu aktualisieren

Das folgende `update-global-settings` Beispiel aktualisiert den S3-Bucket, der zum Speichern von Anrufdetailaufzeichnungen für Amazon Chime Business Calling und Amazon Chime Voice Connectors verwendet wird, die dem Administratorkonto zugeordnet sind. AWS

```
aws chime update-global-settings \  
  --business-calling CdrBucket="s3bucket" \  
  --voice-connector CdrBucket="s3bucket"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung globaler Einstellungen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateGlobalSettings](#) in der AWS CLI Befehlsreferenz.

update-phone-number-settings

Das folgende Codebeispiel zeigt die Verwendung `update-phone-number-settings`.

AWS CLI

Um den Namen eines ausgehenden Anrufs zu aktualisieren

Im folgenden `update-phone-number-settings` Beispiel wird der Standardname für ausgehende Anrufe für das Administratorkonto aktualisiert. AWS

```
aws chime update-phone-number-settings \  
  --calling-name "myName"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdatePhoneNumberSettings](#) in der AWS CLI Befehlsreferenz.

update-phone-number

Das folgende Codebeispiel zeigt die Verwendung `update-phone-number`.

AWS CLI

Beispiel 1: Um den Produkttyp für eine Telefonnummer zu aktualisieren

Im folgenden `update-phone-number` Beispiel wird der Produkttyp der angegebenen Telefonnummer aktualisiert.

```
aws chime update-phone-number \
  --phone-number-id "+12065550100" \
  --product-type "BusinessCalling"
```

Ausgabe:

```
{
  "PhoneNumber": {
    "PhoneNumberId": "%2B12065550100",
    "E164PhoneNumber": "+12065550100",
    "Type": "Local",
    "ProductType": "BusinessCalling",
    "Status": "Unassigned",
    "Capabilities": {
      "InboundCall": true,
      "OutboundCall": true,
      "InboundSMS": true,
      "OutboundSMS": true,
      "InboundMMS": true,
      "OutboundMMS": true
    },
    "Associations": [],
    "CallingName": "phonenummer1",
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
    "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
  }
}
```

Beispiel 2: Um den Namen eines ausgehenden Anrufs für eine Telefonnummer zu aktualisieren

Im folgenden `update-phone-number` Beispiel wird der Name für ausgehende Anrufe für die angegebene Telefonnummer aktualisiert.

```
aws-Glockenspiel update-phone-number -- phone-number-id „+1206550100“ --calling-name  
„Telefonnummer2“
```

Ausgabe:

```
{  
  "PhoneNumber": {  
    "PhoneNumberId": "%2B12065550100",  
    "E164PhoneNumber": "+12065550100",  
    "Type": "Local",  
    "ProductType": "BusinessCalling",  
    "Status": "Unassigned",  
    "Capabilities": {  
      "InboundCall": true,  
      "OutboundCall": true,  
      "InboundSMS": true,  
      "OutboundSMS": true,  
      "InboundMMS": true,  
      "OutboundMMS": true  
    },  
    "Associations": [],  
    "CallingName": "phonenumber2",  
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",  
    "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Telefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdatePhoneNumber](#) in der AWS CLI Befehlsreferenz.

update-proxy-session

Das folgende Codebeispiel zeigt die Verwendung `update-proxy-session`.

AWS CLI

Um eine Proxysitzung zu aktualisieren

Im folgenden `update-proxy-session` Beispiel werden die Funktionen der Proxysitzung aktualisiert.

```
aws chime update-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891 \  
  --capabilities "Voice"
```

Ausgabe:

```
{  
  "ProxySession": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",  
    "Status": "Open",  
    "ExpiryMinutes": 60,  
    "Capabilities": [  
      "Voice"  
    ],  
    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "UpdatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "Participants": [  
      {  
        "PhoneNumber": "+12065550100",  
        "ProxyPhoneNumber": "+19135550199"  
      },  
      {  
        "PhoneNumber": "+14015550101",  
        "ProxyPhoneNumber": "+19135550199"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Proxy-Telefonsitzungen](#) im Amazon Chime Developer Guide.

- API-Einheiten finden Sie [UpdateProxySession](#) in der AWS CLI Befehlsreferenz.

update-room-membership

Das folgende Codebeispiel zeigt die Verwendung `update-room-membership`.

AWS CLI

Um eine Raummemberschaft zu aktualisieren

Im folgenden `update-room-membership` Beispiel wird die Rolle des angegebenen Chatroom-Mitglieds auf `Administrator` geändert.

```
aws chime update-room-membership \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k \  
  --role Administrator
```

Ausgabe:

```
{  
  "RoomMembership": {  
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",  
    "Member": {  
      "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",  
      "MemberType": "User",  
      "Email": "sofiamartinez@example.com",  
      "FullName": "Sofia Martinez",  
      "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"  
    },  
    "Role": "Administrator",  
    "InvitedBy": "arn:aws:iam::111122223333:user/admin",  
    "UpdatedTimestamp": "2019-12-02T22:40:22.931Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRoomMembership](#) in der AWS CLI Befehlsreferenz.

update-room

Das folgende Codebeispiel zeigt die Verwendung `update-room`.

AWS CLI

Um einen Chatroom zu aktualisieren

Im folgenden `update-room` Beispiel wird der Name des angegebenen Chatrooms geändert.

```
aws chime update-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --name teamRoom
```

Ausgabe:

```
{  
  "Room": {  
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",  
    "Name": "teamRoom",  
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",  
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",  
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",  
    "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Chat-Raums](#) im Amazon Chime Chime-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRoom](#) in der AWS CLI Befehlsreferenz.

update-user-settings

Das folgende Codebeispiel zeigt die Verwendung `update-user-settings`.

AWS CLI

Um Benutzereinstellungen zu aktualisieren

Das folgende `update-user-settings` Beispiel ermöglicht es dem angegebenen Benutzer, eingehende und ausgehende Anrufe zu tätigen und Nachrichten zu senden und zu empfangen SMS.

```
aws chime update-user-settings \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \  
  --user-settings "Telephony={InboundCalling=true,OutboundCalling=true,SMS=true}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Benutzertelefonnummern](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateUserSettings](#) in der AWS CLI Befehlsreferenz.

update-user

Das folgende Codebeispiel zeigt die Verwendung `update-user`.

AWS CLI

Um Benutzerdetails zu aktualisieren

In diesem Beispiel werden die angegebenen Details für den angegebenen Benutzer aktualisiert.

Befehl:

```
aws chime update-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \  
  --license-type "Basic"
```

Ausgabe:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"  
  }  
}
```

- API-Einheiten finden Sie [UpdateUser](#) in der AWS CLI Befehlsreferenz.

update-voice-connector-group

Das folgende Codebeispiel zeigt die Verwendung `update-voice-connector-group`.

AWS CLI

Um die Details für eine Amazon Chime Voice Connector-Gruppe zu aktualisieren

Das folgende `update-voice-connector-group` Beispiel aktualisiert die Details der angegebenen Amazon Chime Voice Connector-Gruppe.

```
aws chime update-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --name "newGroupName" \  
  --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=1
```

Ausgabe:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "newGroupName",  
    "VoiceConnectorItems": [  
      {  
        "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
        "Priority": 1  
      }  
    ],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-10-28T19:00:57.081Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connector-Gruppen](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateVoiceConnectorGroup](#) in der AWS CLI Befehlsreferenz.

update-voice-connector

Das folgende Codebeispiel zeigt die Verwendung `update-voice-connector`.

AWS CLI

Um die Details für einen Amazon Chime Voice Connector zu aktualisieren

Das folgende `update-voice-connector` Beispiel aktualisiert den Namen des angegebenen Amazon Chime Voice Connectors.

```
aws chime update-voice-connector \  
  --voice-connector-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --name "newConnectorName"
```

```
--voice-connector-id abcdef1ghij2klmno3pqr4 \  
--name newName \  
--require-encryption
```

Ausgabe:

```
{  
  "VoiceConnector": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "AwsRegion": "us-west-2",  
    "Name": "newName",  
    "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",  
    "RequireEncryption": true,  
    "CreatedTimestamp": "2019-09-18T20:34:01.352Z",  
    "UpdatedTimestamp": "2019-09-18T20:40:52.895Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Amazon Chime Voice Connectors](#) im Amazon Chime Chime-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateVoiceConnector](#) in der AWS CLI Befehlsreferenz.

Cloud API Control-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Cloud Control Aktionen ausführen und allgemeine Szenarien implementieren. API. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-resource

Das folgende Codebeispiel zeigt die Verwendung `create-resource`.

AWS CLI

Um eine Ressource zu erstellen

Im folgenden `create-resource` Beispiel wird eine benannte AWS ResourceExample: :Kinesis: :Stream-Ressource mit einer Aufbewahrungszeit von 168 Stunden und einer Shard-Anzahl von drei erstellt.

```
aws cloudcontrol create-resource \  
  --type-name AWS::Kinesis::Stream \  
  --desired-state "{\"Name\": \"ResourceExample\", \"RetentionPeriodHours\":168, \  
  \"ShardCount\":3}"
```

Ausgabe:

```
{  
  "ProgressEvent": {  
    "EventTime": 1632506656.706,  
    "TypeName": "AWS::Kinesis::Stream",  
    "OperationStatus": "IN_PROGRESS",  
    "Operation": "CREATE",  
    "Identifier": "ResourceExample",  
    "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Ressource](#) im Cloud Control-Benutzerhandbuch. API

- API-Einheiten finden Sie [CreateResource](#) in der AWS CLI Befehlsreferenz.

delete-resource

Das folgende Codebeispiel zeigt die Verwendung `delete-resource`.

AWS CLI

Um eine Ressource zu löschen

Im folgenden `delete-resource` Beispiel wird eine `AWS::Kinesis::Stream`-Ressource mit der ID `ResourceExample` aus Ihrem Konto gelöscht. AWS

```
aws cloudcontrol delete-resource \  
  --type-name AWS::Kinesis::Stream \  
  --identifier ResourceExample
```

Ausgabe:

```
{  
  "ProgressEvent": {  
    "TypeName": "AWS::Kinesis::Stream",  
    "Identifier": "ResourceExample",  
    "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",  
    "Operation": "DELETE",  
    "OperationStatus": "IN_PROGRESS",  
    "EventTime": 1632950300.14  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen einer Ressource](#) im Cloud Control-Benutzerhandbuch. API

- API-Einheiten finden Sie [DeleteResource](#) in der AWS CLI Befehlsreferenz.

get-resource-request-status

Das folgende Codebeispiel zeigt die Verwendung `get-resource-request-status`.

AWS CLI

Um die Statusinformationen einer Ressourcenanforderung abzurufen

Im folgenden `get-resource-request-status` Beispiel werden Statusinformationen zur angegebenen Ressourcenanforderung zurückgegeben.

```
aws cloudcontrol get-resource-request-status \  
  --request-id RequestExample
```

```
--request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

Ausgabe:

```
{
  "ProgressEvent": {
    "TypeName": "AWS::Kinesis::Stream",
    "Identifier": "Demo",
    "RequestToken": "e1a6b86e-46bd-41ac-bfba-001234567890",
    "Operation": "CREATE",
    "OperationStatus": "FAILED",
    "EventTime": 1632950268.481,
    "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with identifier
'Demo' already exists.",
    "ErrorCode": "AlreadyExists"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung von Ressourcenbetriebsanforderungen](#) im Cloud API Control-Benutzerhandbuch.

- API-Einheiten finden Sie [GetResourceRequestStatus](#) in der AWS CLI Befehlsreferenz.

get-resource

Das folgende Codebeispiel zeigt die Verwendung `get-resource`.

AWS CLI

Um den aktuellen Status einer Ressource abzurufen

Das folgende `get-resource` Beispiel gibt den aktuellen Status der benannten `AWS::Kinesis::Stream`-Ressource zurück. `ResourceExample`

```
aws cloudcontrol get-resource \
  --type-name AWS::Kinesis::Stream \
  --identifier ResourceExample
```

Ausgabe:

```
{
```

```

    "TypeName": "AWS::Kinesis::Stream",
    "ResourceDescription": {
      "Identifier": "ResourceExample",
      "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/ResourceExample\", \"RetentionPeriodHours\":168, \"Name\":\"ResourceExample\", \"ShardCount\":3}"
    }
  }
}

```

Weitere Informationen finden Sie unter [Lesen des aktuellen Status einer Ressource](#) im Cloud API Control-Benutzerhandbuch.

- API-Einheiten finden Sie [GetResource](#) in der AWS CLI Befehlsreferenz.

list-resource-requests

Das folgende Codebeispiel zeigt die Verwendung `list-resource-requests`.

AWS CLI

Um die aktiven Ressourcenbetriebsanforderungen aufzulisten

Im folgenden `list-resource-requests` Beispiel werden die Ressourcenanforderungen CREATE und die fehlgeschlagenen UPDATE Vorgänge in Ihrem AWS Konto aufgeführt.

```

aws cloudcontrol list-resource-requests \
  --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED

```

Ausgabe:

```

{
  "ResourceRequestStatusSummaries": [
    {
      "TypeName": "AWS::Kinesis::Stream",
      "Identifier": "Demo",
      "RequestToken": "e1a6b86e-46bd-41ac-bfba-633abcdfdbd7",
      "Operation": "CREATE",
      "OperationStatus": "FAILED",
      "EventTime": 1632950268.481,
      "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with
identifier 'Demo' already exists.",
      "ErrorCode": "AlreadyExists"
    }
  ]
}

```

```

    }
  ]
}

```

Weitere Informationen finden Sie unter [Verwaltung von Anfragen zu Ressourcenoperationen](#) im Cloud API Control-Benutzerhandbuch.

- API-Einheiten finden Sie [ListResourceRequests](#) in der AWS CLI Befehlsreferenz.

list-resources

Das folgende Codebeispiel zeigt die Verwendung `list-resources`.

AWS CLI

Um die Ressourcen eines bestimmten Typs aufzulisten

Das folgende `list-resources` Beispiel listet die `AWS::Kinesis::Stream`-Ressourcen auf, die in Ihrem Konto bereitgestellt werden. AWS

```

aws cloudcontrol list-resources \
  --type-name AWS::Kinesis::Stream

```

Ausgabe:

```

{
  "TypeName": "AWS::Kinesis::Stream",
  "ResourceDescriptions": [
    {
      "Identifier": "MyKinesisStream",
      "Properties": "{\"Name\":\"MyKinesisStream\"}"
    },
    {
      "Identifier": "AnotherStream",
      "Properties": "{\"Name\":\"AnotherStream\"}"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Ressourcen entdecken](#) im Cloud Control-Benutzerhandbuch. API

- API-Einheiten finden Sie [ListResources](#) in der AWS CLI Befehlsreferenz.

update-resource

Das folgende Codebeispiel zeigt die Verwendung `update-resource`.

AWS CLI

Um die Eigenschaften einer vorhandenen Ressource zu aktualisieren

Im folgenden `update-resource` Beispiel wird die Aufbewahrungsrichtlinie einer `AWS::Logs::LogGroup`-Ressource mit dem Namen `ExampleLogGroup` 90 Tage aktualisiert.

```
aws cloudcontrol update-resource \  
  --type-name AWS::Logs::LogGroup \  
  --identifier ExampleLogGroup \  
  --patch-document "[{\\"op\\":\\"replace\\",\\"path\\":\\"/RetentionInDays\\",\\"value\\":90}]"
```

Ausgabe:

```
{  
  "ProgressEvent": {  
    "EventTime": "2021-08-09T18:17:15.219Z",  
    "TypeName": "AWS::Logs::LogGroup",  
    "OperationStatus": "IN_PROGRESS",  
    "Operation": "UPDATE",  
    "Identifier": "ExampleLogGroup",  
    "RequestToken": "5f40c577-3534-4b20-9599-0b0123456789"  
  }  
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer Ressource](#) im Cloud API Control-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateResource](#) in der AWS CLI Befehlsreferenz.

AWS Cloud Map Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Cloud Map.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-private-dns-namespace

Das folgende Codebeispiel zeigt die Verwendung `create-private-dns-namespace`.

AWS CLI

Um einen privaten DNS Namespace zu erstellen

Das folgende `create-private-dns-namespace` Beispiel erstellt einen privaten DNS Namespace.

```
aws servicediscovery create-private-dns-namespace \  
  --name example.com \  
  --vpc vpc-1c56417b
```

Ausgabe:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"  
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen `get-operation`. Weitere Informationen finden Sie unter [get-operation](#).

Weitere Informationen finden Sie im AWS Cloud Map Developer Guide unter [Creating Namespaces](#).

- API-Einheiten finden Sie [CreatePrivateDnsNamespace](#) in AWS CLI der Befehlsreferenz.

create-service

Das folgende Codebeispiel zeigt die Verwendung `create-service`.

AWS CLI

Um einen Dienst zu erstellen

Das folgende `create-service` Beispiel erstellt einen Dienst.

```
aws servicediscovery create-service \  
  --name myservice \  
  --namespace-id ns-ylexjili4cdxy3xm \  
  --dns-config "NamespaceId=ns-  
ylexjili4cdxy3xm,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

Ausgabe:

```
{  
  "Service": {  
    "Id": "srv-p5zdwlg5uvvzjita",  
    "Arn": "arn:aws:servicediscovery:us-west-2:803642222207:service/srv-  
p5zdwlg5uvvzjita",  
    "Name": "myservice",  
    "NamespaceId": "ns-ylexjili4cdxy3xm",  
    "DnsConfig": {  
      "NamespaceId": "ns-ylexjili4cdxy3xm",  
      "RoutingPolicy": "MULTIVALUE",  
      "DnsRecords": [  
        {  
          "Type": "A",  
          "TTL": 60  
        }  
      ]  
    },  
    "CreateDate": 1587081768.334,  
    "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"  
  }  
}
```

Weitere Informationen finden Sie unter [Dienste erstellen](#) im AWS Cloud Map Developer Guide.

- API-Einheiten finden Sie [CreateService](#) in der AWS CLI Befehlsreferenz.

delete-namespace

Das folgende Codebeispiel zeigt die Verwendung `delete-namespace`.

AWS CLI

Um einen Namespace zu löschen

Im folgenden `delete-namespace` Beispiel wird ein Namespace gelöscht.

```
aws servicediscovery delete-namespace \  
  --id ns-ylexjili4cdxy3xm
```

Ausgabe:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6dtk"  
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen. `get-operation`
Weitere Informationen finden Sie unter [get-operation](#).

Weitere Informationen finden Sie unter [Löschen von Namespaces](#) im AWS Cloud Map Developer Guide.

- API-Einheiten finden Sie [DeleteNamespace](#) in AWS CLI der Befehlsreferenz.

delete-service

Das folgende Codebeispiel zeigt die Verwendung `delete-service`.

AWS CLI

Um einen Dienst zu löschen

Im folgenden `delete-service` Beispiel wird ein Dienst gelöscht.

```
aws servicediscovery delete-service \  
  --id srv-p5zdwlg5uvvzjita
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dienste löschen](#) im AWS Cloud Map Developer Guide.

- API-Einheiten finden Sie [DeleteService](#) in der AWS CLI Befehlsreferenz.

deregister-instance

Das folgende Codebeispiel zeigt die Verwendung `deregister-instance`.

AWS CLI

Um die Registrierung einer Dienstinstanz aufzuheben

Im folgenden `deregister-instance` Beispiel wird die Registrierung einer Dienstinstanz aufgehoben.

```
aws servicediscovery deregister-instance \  
  --service-id srv-p5zdwlg5uvvzjita \  
  --instance-id myservice-53
```

Ausgabe:

```
{  
  "OperationId": "4yejorelbukcjzpnr6t1mrghsjwpngf4-k98rnaiq"  
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen. `get-operation`
Weitere Informationen finden Sie unter [get-operation](#).

Weitere Informationen finden Sie unter [Deregistering Service Instances](#) im AWS Cloud Map Developer Guide.

- API-Einheiten finden Sie [DeregisterInstance](#) in der AWS CLI Befehlsreferenz.

discover-instances

Das folgende Codebeispiel zeigt die Verwendung `discover-instances`.

AWS CLI

Um registrierte Instanzen zu entdecken

Im folgenden `discover-instances` Beispiel werden registrierte Instanzen erkannt.

```
aws servicediscovery discover-instances \  
  --service-id srv-p5zdwlg5uvvzjita
```

```
--namespace-name example.com \  
--service-name myservice \  
--max-results 10 \  
--health-status ALL
```

Ausgabe:

```
{  
  "Instances": [  
    {  
      "InstanceId": "myservice-53",  
      "NamespaceName": "example.com",  
      "ServiceName": "myservice",  
      "HealthStatus": "UNKNOWN",  
      "Attributes": {  
        "AWS_INSTANCE_IPV4": "172.2.1.3",  
        "AWS_INSTANCE_PORT": "808"  
      }  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DiscoverInstances](#) in der AWS CLI Befehlsreferenz.

get-operation

Das folgende Codebeispiel zeigt die Verwendung der `get-operation`-Operation.

AWS CLI

Um das Ergebnis einer Operation zu erhalten

Das folgende `get-operation`-Beispiel ruft das Ergebnis einer Operation ab.

```
aws servicediscovery get-operation \  
--operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

Ausgabe:

```
{  
  "Operation": {  
    "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",
```

```

    "Type": "CREATE_NAMESPACE",
    "Status": "SUCCESS",
    "CreateDate": 1587055860.121,
    "UpdateDate": 1587055900.469,
    "Targets": {
      "NAMESPACE": "ns-ylexjili4cdxy3xm"
    }
  }
}

```

- API-Einheiten finden Sie [GetOperation](#) unter AWS CLI Befehlsreferenz.

list-instances

Das folgende Codebeispiel zeigt die Verwendung `list-instances`.

AWS CLI

Um Dienstinstanzen aufzulisten

Das folgende `list-instances` Beispiel listet Dienstinstanzen auf.

```

aws servicediscovery list-instances \
  --service-id srv-qzpwvt2tfqcegapy

```

Ausgabe:

```

{
  "Instances": [
    {
      "Id": "i-06bdabbae60f65a4e",
      "Attributes": {
        "AWS_INSTANCE_IPV4": "172.2.1.3",
        "AWS_INSTANCE_PORT": "808"
      }
    }
  ]
}

```

Weitere Informationen finden Sie im AWS Cloud Map Developer Guide unter [Eine Liste von Dienstinstanzen](#) anzeigen.

- API-Einheiten finden Sie [ListInstances](#) in der AWS CLI Befehlsreferenz.

list-namespaces

Das folgende Codebeispiel zeigt die Verwendung `list-namespaces`.

AWS CLI

Um Namespaces aufzulisten

Das folgende `list-namespaces` Beispiel listet Namespaces auf.

```
aws servicediscovery list-namespaces
```

Ausgabe:

```
{
  "Namespaces": [
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-a3ccy2e7e3a7rile",
      "CreateDate": 1585354387.357,
      "Id": "ns-a3ccy2e7e3a7rile",
      "Name": "local",
      "Properties": {
        "DnsProperties": {
          "HostedZoneId": "Z06752353VBUDTC32S84S"
        },
        "HttpProperties": {
          "HttpName": "local"
        }
      },
      "Type": "DNS_PRIVATE"
    },
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-pocfyjtrsmwtvcxx",
      "CreateDate": 1586468974.698,
      "Description": "My second namespace",
      "Id": "ns-pocfyjtrsmwtvcxx",
      "Name": "My-second-namespace",
      "Properties": {
        "DnsProperties": {},
        "HttpProperties": {
          "HttpName": "My-second-namespace"
        }
      }
    }
  ]
}
```

```
    }
  },
  "Type": "HTTP"
},
{
  "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
ylexjili4cdxy3xm",
  "CreateDate": 1587055896.798,
  "Id": "ns-ylexjili4cdxy3xm",
  "Name": "example.com",
  "Properties": {
    "DnsProperties": {
      "HostedZoneId": "Z09983722P0QME1B3KC8I"
    },
    "HttpProperties": {
      "HttpName": "example.com"
    }
  },
  "Type": "DNS_PRIVATE"
}
]
}
```

Weitere Informationen finden Sie im AWS Cloud Map Developer Guide unter [Eine Liste von Namespaces](#) anzeigen.

- API-Einheiten finden Sie unter Befehlsreferenz [ListNamespaces](#).AWS CLI

list-services

Das folgende Codebeispiel zeigt die Verwendung `list-services`.

AWS CLI

Um Dienste aufzulisten

Das folgende `list-services` Beispiel listet Dienste auf.

```
aws servicediscovery list-services
```

Ausgabe:

```
{
```

```

    "Services": [
      {
        "Id": "srv-p5zdwlg5uvvzjita",
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
p5zdwlg5uvvzjita",
        "Name": "myservice",
        "DnsConfig": {
          "RoutingPolicy": "MULTIVALUE",
          "DnsRecords": [
            {
              "Type": "A",
              "TTL": 60
            }
          ]
        },
        "CreateDate": 1587081768.334
      }
    ]
  }
}

```

Weitere Informationen finden Sie im AWS Cloud Map Developer Guide unter [Eine Liste von Diensten](#) anzeigen.

- API-Einheiten finden Sie [ListServices](#) in der AWS CLI Befehlsreferenz.

register-instance

Das folgende Codebeispiel zeigt die Verwendung `register-instance`.

AWS CLI

Um eine Dienstinstanz zu registrieren

Im folgenden `register-instance` Beispiel wird eine Dienstinstanz registriert.

```

aws servicediscovery register-instance \
  --service-id srv-p5zdwlg5uvvzjita \
  --instance-id myservice-53 \
  --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808

```

Ausgabe:

```
{
```

```
"OperationId": "4yejorelbukcjzpnr6t1mrghsjwpngf4-k95yg2u7"  
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen `get-operation`. Weitere Informationen finden Sie unter [get-operation](#).

Weitere Informationen finden Sie unter [Registrierung von Instanzen](#) im AWS Cloud Map Developer Guide.

- API-Einheiten finden Sie [RegisterInstance](#) in der AWS CLI Befehlsreferenz.

AWS Cloud9 Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Cloud9.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-environment-ec2

Das folgende Codebeispiel zeigt die Verwendung `create-environment-ec2`.

AWS CLI

Um eine AWS EC2 Cloud9-Entwicklungsumgebung zu erstellen

Das folgende `create-environment-ec2` Beispiel erstellt eine AWS Cloud9-Entwicklungsumgebung mit den angegebenen Einstellungen, startet eine Amazon Elastic Compute Cloud (AmazonEC2) -Instance und stellt dann eine Verbindung von der Instance zur Umgebung her.


```
aws cloud9 create-environment-ec2 \  
  --name my-demo-env \  
  --description "My demonstration development environment." \  
  --instance-type t2.micro --image-id amazonlinux-2023-x86_64 \  
  --subnet-id subnet-1fab8aEX \  
  --automatic-stop-time-minutes 60 \  
  --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

Ausgabe:

```
{  
  "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"  
}
```

Weitere Informationen finden Sie unter [Creating an EC2 Environment](#) im AWS Cloud9-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateEnvironmentEc2](#) in der AWS CLI Befehlsreferenz.

create-environment-membership

Das folgende Codebeispiel zeigt die Verwendung `create-environment-membership`.

AWS CLI

Um ein Umgebungsmitglied zu einer AWS Cloud9-Entwicklungsumgebung hinzuzufügen

In diesem Beispiel wird das angegebene Umgebungsmitglied zur angegebenen AWS Cloud9-Entwicklungsumgebung hinzugefügt.

Befehl:

```
aws cloud9 create-environment-membership --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/  
AnotherDemoUser --permissions read-write
```

Ausgabe:

```
{  
  "membership": {  
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",  
    "userId": "AIDAJ3LOROMOUCTBSU6EX",
```

```
"userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",  
"permissions": "read-write"  
}  
}
```

- API-Einheiten finden Sie [CreateEnvironmentMembership](#) in der AWS CLI Befehlsreferenz.

delete-environment-membership

Das folgende Codebeispiel zeigt die Verwendung `delete-environment-membership`.

AWS CLI

Um ein Umgebungsmitglied aus einer AWS Cloud9-Entwicklungsumgebung zu löschen

In diesem Beispiel wird das angegebene Umgebungsmitglied aus der angegebenen AWS Cloud9-Entwicklungsumgebung gelöscht.

Befehl:

```
aws cloud9 delete-environment-membership --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/  
AnotherDemoUser
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [DeleteEnvironmentMembership](#) in der AWS CLI Befehlsreferenz.

delete-environment

Das folgende Codebeispiel zeigt die Verwendung `delete-environment`.

AWS CLI

Um eine AWS Cloud9-Entwicklungsumgebung zu löschen

In diesem Beispiel wird die angegebene AWS Cloud9-Entwicklungsumgebung gelöscht. Wenn eine EC2 Amazon-Instance mit der Umgebung verbunden ist, wird auch die Instance beendet.

Befehl:

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [DeleteEnvironment](#) in der AWS CLI Befehlsreferenz.

describe-environment-memberships

Das folgende Codebeispiel zeigt die Verwendung `describe-environment-memberships`.

AWS CLI

Um Informationen über Umgebungsmitglieder für eine AWS Cloud9-Entwicklungsumgebung abzurufen

In diesem Beispiel werden Informationen über Umgebungsmitglieder für die angegebene AWS Cloud9-Entwicklungsumgebung abgerufen.

Befehl:

```
aws cloud9 describe-environment-memberships --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

Ausgabe:

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJ3LOROMOXTBSU6EX",
      "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
      "permissions": "read-write"
    },
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

```
]
}
```

Um Informationen über den Besitzer einer AWS Cloud9-Entwicklungsumgebung zu erhalten

In diesem Beispiel werden Informationen über den Besitzer der angegebenen AWS Cloud9-Entwicklungsumgebung abgerufen.

Befehl:

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

Ausgabe:

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

Um Informationen über ein Umgebungsmitglied für mehrere AWS Cloud9-Entwicklungsumgebungen zu erhalten

In diesem Beispiel werden Informationen über das angegebene Umgebungsmitglied für mehrere AWS Cloud9-Entwicklungsumgebungen abgerufen.

Befehl:

```
aws cloud9 describe-environment-memberships --user-
arn arn:aws:iam::123456789012:user/MyDemoUser
```

Ausgabe:

```
{
  "memberships": [
    {
```

```
"environmentId": "10a75714bd494714929e7f5ec4125aEX",
"lastAccess": 1516213427.0,
"userId": "AIDAJNUEDQAQWFELJDLEX",
"userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
"permissions": "owner"
},
{
"environmentId": "1980b80e5f584920801c09086667f0EX",
"lastAccess": 1516144884.0,
"userId": "AIDAJNUEDQAQWFELJDLEX",
"userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
"permissions": "owner"
}
]
}
```

- API-Einheiten finden Sie [DescribeEnvironmentMemberships](#) in der AWS CLI Befehlsreferenz.

describe-environment-status

Das folgende Codebeispiel zeigt die Verwendung `describe-environment-status`.

AWS CLI

Um Statusinformationen für eine AWS Cloud9-Entwicklungsumgebung abzurufen

In diesem Beispiel werden Statusinformationen für die angegebene AWS Cloud9-Entwicklungsumgebung abgerufen.

Befehl:

```
aws cloud9 describe-environment-status --environment-
id 685f892f431b45c2b28cb69eadcdb0EX
```

Ausgabe:

```
{
"status": "ready",
"message": "Environment is ready to use"
}
```

- API-Einheiten finden Sie [DescribeEnvironmentStatus](#) in der AWS CLI Befehlsreferenz.

describe-environments

Das folgende Codebeispiel zeigt die Verwendung `describe-environments`.

AWS CLI

Um Informationen über AWS Cloud9-Entwicklungsumgebungen zu erhalten

In diesem Beispiel werden Informationen zu den angegebenen AWS Cloud9-Entwicklungsumgebungen abgerufen.

Befehl:

```
aws cloud9 describe-environments --environment-ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

Ausgabe:

```
{
  "environments": [
    {
      "id": "685f892f431b45c2b28cb69eadcdb0EX",
      "name": "my-demo-ec2-env",
      "description": "Created from CodeStar.",
      "type": "ec2",
      "arn": "arn:aws:cloud9:us-east-1:123456789012:environment:685f892f431b45c2b28cb69eadcdb0EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    },
    {
      "id": "349c86d4579e4e7298d500ff57a6b2EX",
      "name": "my-demo-ssh-env",
      "description": "",
      "type": "ssh",
      "arn": "arn:aws:cloud9:us-east-1:123456789012:environment:349c86d4579e4e7298d500ff57a6b2EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    }
  ]
}
```

```
]
}
```

- API-Einheiten finden Sie [DescribeEnvironments](#) in der AWS CLI Befehlsreferenz.

list-environments

Das folgende Codebeispiel zeigt die Verwendung `list-environments`.

AWS CLI

Um eine Liste der verfügbaren AWS Cloud9-Entwicklungsumgebungskennungen zu erhalten

In diesem Beispiel wird eine Liste verfügbarer AWS Cloud9-Entwicklungsumgebungskennungen abgerufen.

Befehl:

```
aws cloud9 list-environments
```

Ausgabe:

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- API-Einheiten finden Sie [ListEnvironments](#) in der AWS CLI Befehlsreferenz.

update-environment-membership

Das folgende Codebeispiel zeigt die Verwendung `update-environment-membership`.

AWS CLI

Um die Einstellungen eines vorhandenen Umgebungsmitglieds für eine AWS Cloud9-Entwicklungsumgebung zu ändern

In diesem Beispiel werden die Einstellungen des angegebenen vorhandenen Umgebungsmitglieds für die angegebene AWS Cloud9-Entwicklungsumgebung geändert.

Befehl:

```
aws cloud9 update-environment-membership --environment-id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/AnotherDemoUser --permissions read-only
```

Ausgabe:

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUCTBSUGEX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
}
```

- API-Einheiten finden Sie unter [UpdateEnvironmentMembership AWS CLI](#) Befehlsreferenz.

update-environment

Das folgende Codebeispiel zeigt die Verwendung `update-environment`.

AWS CLI

Um die Einstellungen einer vorhandenen AWS Cloud9-Entwicklungsumgebung zu ändern

In diesem Beispiel werden die angegebenen Einstellungen der angegebenen vorhandenen AWS Cloud9-Entwicklungsumgebung geändert.

Befehl:

```
aws cloud9 update-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX --name my-changed-demo-env --description "My changed demonstration development environment."
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [UpdateEnvironment](#) in der AWS CLI Befehlsreferenz.

AWS CloudFormation Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS CloudFormation.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

activate-type

Das folgende Codebeispiel zeigt die Verwendung `activate-type`.

AWS CLI

Um einen Typ zu aktivieren

Im folgenden `activate-type` Beispiel wird eine öffentliche Erweiterung eines Drittanbieters aktiviert, sodass sie für die Verwendung in Stack-Vorlagen verfügbar ist.

```
aws cloudformation activate-type \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0 \  
  --type-name-alias Example::Test::Alias
```

Ausgabe:

```
{  
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-  
Test-Alias"  
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [ActivateType](#) unter AWS CLI Befehlsreferenz.

batch-describe-type-configurations

Das folgende Codebeispiel zeigt die Verwendung `batch-describe-type-configurations`.

AWS CLI

Um eine Typkonfiguration stapelweise zu beschreiben

Im folgenden `batch-describe-type-configurations` Beispiel werden die Daten für den Typ konfiguriert.

```
aws cloudformation batch-describe-type-configurations \
  --region us-west-2 \
  --type-configuration-identifiers TypeArn="arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Example-Test-
Type,TypeConfigurationAlias=MyConfiguration"
```

Ausgabe:

```
{
  "Errors": [],
  "UnprocessedTypeConfigurations": [],
  "TypeConfigurations": [
    {
      "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
Example-Test-Type",
      "Alias": "MyConfiguration",
      "Configuration": "{\n      \"Example\": {\n          \"ApiKey\":
\n\"examplekey\", \n          \"ApplicationKey\": \"examplekey1\", \n
\n\"ApiURL\": \"exampleurl\"\n      }\n}",
      "LastUpdated": "2021-10-01T15:25:46.210000+00:00",
      "TypeArn": "arn:aws:cloudformation:us-east-1:123456789012:type/resource/
Example-Test-Type"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [BatchDescribeTypeConfigurations](#) unter AWS CLI Befehlsreferenz.

cancel-update-stack

Das folgende Codebeispiel zeigt die Verwendung `cancel-update-stack`.

AWS CLI

Um ein laufendes Stack-Update abubrechen

Mit dem folgenden `cancel-update-stack` Befehl wird ein Stack-Update auf dem `myteststack` Stack abgebrochen:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- API-Einheiten finden Sie [CancelUpdateStack](#) in der AWS CLI Befehlsreferenz.

continue-update-rollback

Das folgende Codebeispiel zeigt die Verwendung `continue-update-rollback`.

AWS CLI

Um einen Update-Rollback erneut zu versuchen

Im folgenden `continue-update-rollback` Beispiel wird ein Rollback-Vorgang nach einem zuvor fehlgeschlagenen Stack-Update wieder aufgenommen.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [ContinueUpdateRollback](#) in der AWS CLI Befehlsreferenz.

create-change-set

Das folgende Codebeispiel zeigt die Verwendung `create-change-set`.

AWS CLI

Um einen Änderungssatz zu erstellen

Im folgenden `create-change-set` Beispiel wird ein Änderungssatz mit der `CAPABILITY_IAM` Fähigkeit erstellt. Die Datei `template.yaml` ist eine AWS CloudFormation Vorlage im aktuellen Ordner, die einen Stapel definiert, der IAM Ressourcen enthält.

```
aws cloudformation create-change-set \  
  --stack-name my-application \  
  --change-set-name my-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM
```

Ausgabe:

```
{  
  "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/  
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-application/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"  
}
```

- API-Einheiten finden Sie [CreateChangeSet](#) in der AWS CLI Befehlsreferenz.

`create-stack-instances`

Das folgende Codebeispiel zeigt die Verwendung `create-stack-instances`.

AWS CLI

Um Stack-Instanzen zu erstellen

Im folgenden `create-stack-instances` Beispiel werden Instanzen eines Stack-Sets in zwei Konten und in vier Regionen erstellt. Die Einstellung für die Fehlertoleranz stellt sicher, dass das Update in allen Konten und Regionen versucht wird, auch wenn einige Stacks nicht erstellt werden können.

```
aws cloudformation create-stack-instances \  
  --stack-set-name my-stack-set \  
  --template-body file://template.yaml
```

```
--accounts 123456789012 223456789012 \  
--regions us-east-1 us-east-2 us-west-1 us-west-2 \  
--operation-preferences FailureToleranceCount=7
```

Ausgabe:

```
{  
  "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"  
}
```

Verwenden Sie den `create-stack-set` Befehl, um ein Stack-Set zu erstellen.

- API-Einheiten finden Sie [CreateStackInstances](#) in der AWS CLI Befehlsreferenz.

create-stack-set

Das folgende Codebeispiel zeigt die Verwendung `create-stack-set`.

AWS CLI

Um ein Stack-Set zu erstellen

Im folgenden `create-stack-set` Beispiel wird ein Stack-Set unter Verwendung der angegebenen YAML-Dateivorlage erstellt. `template.yaml` ist eine AWS CloudFormation Vorlage im aktuellen Ordner, die einen Stapel definiert.

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stack-set \  
  --template-body file://template.yaml \  
  --description "SNS topic"
```

Ausgabe:

```
{  
  "StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"  
}
```

Verwenden Sie den `create-stack-instances` Befehl, um dem Stack-Set Stack-Instanzen hinzuzufügen.

- API-Einheiten finden Sie [CreateStackSet](#) in der AWS CLI Befehlsreferenz.

create-stack

Das folgende Codebeispiel zeigt die Verwendung `create-stack`.

AWS CLI

Um einen AWS CloudFormation Stapel zu erstellen

Der folgende `create-stacks` Befehl erstellt `myteststack` mithilfe der `sampltemplate.json` Vorlage einen Stack mit dem Namen:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://  
sampltemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Ausgabe:

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

Weitere Informationen finden Sie unter `Stacks` im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [CreateStack](#) in der AWS CLI Befehlsreferenz.

deactivate-type

Das folgende Codebeispiel zeigt die Verwendung `deactivate-type`.

AWS CLI

Um einen Typ zu deaktivieren

Im folgenden `deactivate-type` Beispiel wird eine öffentliche Erweiterung deaktiviert, die zuvor in diesem Konto und dieser Region aktiviert wurde.

```
aws cloudformation deactivate-type \  
  --region us-west-2 \  
  --type MODULE \  
  --type-name Example::Test::Type::MODULE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [DeactivateType](#) unter AWS CLI Befehlsreferenz.

delete-change-set

Das folgende Codebeispiel zeigt die Verwendung `delete-change-set`.

AWS CLI

Um einen Änderungssatz zu löschen

Im folgenden `delete-change-set` Beispiel wird ein Änderungssatz gelöscht, indem der Name des Änderungssatzes und der Stackname angegeben werden.

```
aws cloudformation delete-change-set \  
  --stack-name my-stack \  
  --change-set-name my-change-set
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Im folgenden `delete-change-set` Beispiel wird ein Änderungssatz gelöscht, indem der vollständige ARN Änderungssatz angegeben wird.

```
aws cloudformation delete-change-set \  
  --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteChangeSet AWS CLI](#) Befehlsreferenz.

delete-stack-instances

Das folgende Codebeispiel zeigt die Verwendung `delete-stack-instances`.

AWS CLI

Um Stack-Instances zu löschen

Das folgende `delete-stack-instances` Beispiel löscht Instanzen eines Stack-Sets in zwei Konten in zwei Regionen und beendet die Stacks.

```
aws cloudformation delete-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 567890123456 \  
  --regions us-east-1 us-west-1 \  
  --no-retain-stacks
```

Ausgabe:

```
{  
  "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"  
}
```

Verwenden Sie den Befehl, um ein leeres Stack-Set zu löschen. `delete-stack-set`

- API-Einheiten finden Sie [DeleteStackInstances](#) in der AWS CLI Befehlsreferenz.

delete-stack-set

Das folgende Codebeispiel zeigt die Verwendung `delete-stack-set`.

AWS CLI

Um ein Stack-Set zu löschen

Der folgende Befehl löscht das angegebene leere Stack-Set. Das Stack-Set muss leer sein.

```
aws cloudformation delete-stack-set \  
  --stack-set-name my-stack-set
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `delete-stack-instances` Befehl, um Instanzen aus dem Stack-Set zu löschen.

- API-Einheiten finden Sie [DeleteStackSet](#) in der AWS CLI Befehlsreferenz.

delete-stack

Das folgende Codebeispiel zeigt die Verwendung `delete-stack`.


```
--type-name My::Logs::LogGroup \  
--version-id 00000002
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der CloudFormation Registrierung](#).

- APIEinzelheiten finden Sie [DeregisterType](#) unter AWS CLI Befehlsreferenz.

describe-account-limits

Das folgende Codebeispiel zeigt die Verwendung `describe-account-limits`.

AWS CLI

Um Informationen über Ihre Kontolimits zu erhalten

Mit dem folgenden Befehl wird eine Liste der regionalen Beschränkungen für das aktuelle Konto abgerufen.

```
aws cloudformation describe-account-limits
```

Ausgabe:

```
{  
  "AccountLimits": [  
    {  
      "Name": "StackLimit",  
      "Value": 200  
    },  
    {  
      "Name": "StackOutputsLimit",  
      "Value": 60  
    },  
    {  
      "Name": "ConcurrentResourcesLimit",  
      "Value": 2500  
    }  
  ]  
}
```

- APIEinzelheiten finden Sie [DescribeAccountLimits](#) in der AWS CLI Befehlsreferenz.

describe-change-set

Das folgende Codebeispiel zeigt die Verwendung `describe-change-set`.

AWS CLI

Um Informationen über einen Änderungssatz zu erhalten

Im folgenden `describe-change-set` Beispiel werden die Details des Änderungssatzes angezeigt, der durch den Namen des Änderungssatzes und den Stacknamen angegeben ist.

```
aws cloudformation describe-change-set \  
  --change-set-name my-change-set \  
  --stack-name my-stack
```

Im folgenden `describe-change-set` Beispiel werden die Details des durch den gesamten Änderungssatz angegebenen ARN Änderungssatzes angezeigt:

```
aws cloudformation describe-change-set \  
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

Ausgabe:

```
{  
  "Changes": [  
    {  
      "Type": "Resource",  
      "ResourceChange": {  
        "Action": "Modify",  
        "LogicalResourceId": "function",  
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",  
        "ResourceType": "AWS::Lambda::Function",  
        "Replacement": "False",  
        "Scope": [  
          "Properties"  
        ],  
        "Details": [  
          {  
            "Target": {  
              "Attribute": "Properties",  
              "Name": "Timeout",
```

```

        "RequiresRecreation": "Never"
      },
      "Evaluation": "Static",
      "ChangeSource": "DirectModification"
    }
  ]
}
},
"ChangeSetName": "my-change-set",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
"StackName": "my-stack",
"Description": null,
"Parameters": null,
"CreationTime": "2019-10-02T05:20:56.651Z",
"ExecutionStatus": "AVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": null,
"NotificationARNs": [],
"RollbackConfiguration": {},
"Capabilities": [
  "CAPABILITY_IAM"
],
"Tags": null
}

```

- API-Einheiten finden Sie [DescribeChangeSet](#) unter AWS CLI Befehlsreferenz.

describe-publisher

Das folgende Codebeispiel zeigt die Verwendung `describe-publisher`.

AWS CLI

Um einen Herausgeber zu beschreiben

Im folgenden `describe-publisher` Beispiel werden die Informationen für einen Herausgeber konfiguriert.

```
aws cloudformation describe-publisher \
```

```
--region us-west-2 \  
--publisher-id 000q6TfUovXsEMmgKowxDZLLwqr2QUsh
```

Ausgabe:

```
{  
  "PublisherId": "000q6TfUovXsEMmgKowxDZLLwqr2QUshd2e75c8c",  
  "PublisherStatus": "VERIFIED",  
  "IdentityProvider": "AWS_Marketplace",  
  "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?  
id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"  
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [DescribePublisher](#) unter AWS CLI Befehlsreferenz.

describe-stack-drift-detection-status

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-drift-detection-status`.

AWS CLI

Um den Status eines Vorgangs zur Drifterkennung zu überprüfen

Das folgende `describe-stack-drift-detection-status` Beispiel zeigt den Status eines Drifterkennungsvorgangs an. Ruft die BY-ID ab, wenn der `detect-stack-drift` Befehl ausgeführt wird.

```
aws cloudformation describe-stack-drift-detection-status \  
--stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

Ausgabe:

```
{  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
  "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4",  
  "StackDriftStatus": "DRIFTED",  
  "DetectionStatus": "DETECTION_COMPLETE",  
  "DriftedStackResourceCount": 1,  
}
```

```
"Timestamp": "2019-10-02T05:54:30.902Z"
}
```

- API-Einheiten finden Sie [DescribeStackDriftDetectionStatus](#) in der AWS CLI Befehlsreferenz.

describe-stack-events

Das folgende Codebeispiel zeigt die Verwendung von `describe-stack-events`.

AWS CLI

Um Stack-Ereignisse zu beschreiben

Im folgenden `describe-stack-events` Beispiel werden die 2 neuesten Ereignisse für den angegebenen Stack angezeigt.

```
aws cloudformation describe-stack-events \
  --stack-name my-stack \
  --max-items 2

{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2019-10-02T05:34:29.556Z",
      "ResourceStatus": "UPDATE_COMPLETE"
    },
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "ResourceType": "AWS::CloudFormation::Stack",
```

```

    "Timestamp": "2019-10-02T05:34:29.127Z",
    "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
  },
  "NextToken": "eyJJOZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- API-Einheiten finden Sie [DescribeStackEvents](#) in der AWS CLI Befehlsreferenz.

describe-stack-instance

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-instance`.

AWS CLI

Um eine Stack-Instanz zu beschreiben

Der folgende Befehl beschreibt eine Instanz des angegebenen Stack-Sets im angegebenen Konto und in der angegebenen Region. Das Stack-Set befindet sich in der aktuellen Region und dem aktuellen Konto, und die Instanz befindet sich in der `us-west-2` Region im Konto `123456789012`.

```

aws cloudformation describe-stack-instance \
  --stack-set-name my-stack-set \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-west-2

```

Ausgabe:

```

{
  "StackInstance": {
    "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
    "Region": "us-west-2",
    "Account": "123456789012",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/4287f9a0-e615-xmpl-894a-12b31d3117be",
    "ParameterOverrides": [],
    "Status": "OUTDATED",
    "StatusReason": "ResourceLogicalId:ConfigBucket, Resource Type:AWS::S3::Bucket, ResourceStatusReason:You have attempted to create

```

```

more buckets than allowed (Service: Amazon S3; Status Code: 400; Error Code:
TooManyBuckets; Request ID: F7F21CXMPL580224; S3 Extended Request ID: egd/
Fdt89BXMPLyiqbMNljVvk55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj9671hriaG/dWMobS040o=)."
    }
}

```

- API-Einheiten finden Sie [DescribeStackInstance](#) in der AWS CLI Befehlsreferenz.

describe-stack-resource-drifts

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-resource-drifts`.

AWS CLI

Um Informationen über Ressourcen zu erhalten, die von der Stack-Definition abweichen

Mit dem folgenden Befehl werden Informationen zu den Ressourcen angezeigt, die für den angegebenen Stack verschoben wurden. Verwenden Sie den `detect-stack-drift` Befehl, um die Drifterkennung zu initiieren. :

```

aws cloudformation describe-stack-resource-drifts \
  --stack-name my-stack

```

Die Ausgabe zeigt eine AWS Lambda-Funktion, die geändert wurde: `out-of-band`

```

{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "function",
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
      "ResourceType": "AWS::Lambda::Function",
      "ExpectedProperties": "{\"Description\": \"Write a file to S3.\",
\\\"Environment\\\": {\\\"Variables\\\": {\\\"bucket\\\": \\\"my-stack-bucket-1vc62xmplgguf
\\\"}}, \\\"Handler\\\": \\\"index.handler\\\", \\\"MemorySize\\\": 128, \\\"Role\\\":
\\\"arn:aws:iam:123456789012:role/my-functionRole-HIZXMPLE0M9E\\\", \\\"Runtime\\\":
\\\"nodejs10.x\\\", \\\"Tags\\\": [ {\\\"Key\\\": \\\"lambda:createdBy\\\", \\\"Value\\\": \\\"SAM\\\"} ], \\\"Timeout
\\\": 900, \\\"TracingConfig\\\": {\\\"Mode\\\": \\\"Active\\\"} }",
      "ActualProperties": "{\"Description\": \"Write a file to S3.\",
\\\"Environment\\\": {\\\"Variables\\\": {\\\"bucket\\\": \\\"my-stack-bucket-1vc62xmplgguf

```



```

    \"}},\\"Handler\\":\\"index.handler\\",\\"MemorySize\\":256,\\"Role\\":
    \\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\\",\\"Runtime\\":
    \\"nodejs10.x\\",\\"Tags\\":[{\\"Key\\":\\"lambda:createdBy\\",\\"Value\\":\\"SAM\\"}],\\"Timeout
    \":22,\\"TracingConfig\\":{\\"Mode\\":\\"Active\\"}],
      \"PropertyDifferences\": [
        {
          \"PropertyPath\": \"/MemorySize\",
          \"ExpectedValue\": \"128\",
          \"ActualValue\": \"256\",
          \"DifferenceType\": \"NOT_EQUAL\"
        },
        {
          \"PropertyPath\": \"/Timeout\",
          \"ExpectedValue\": \"900\",
          \"ActualValue\": \"22\",
          \"DifferenceType\": \"NOT_EQUAL\"
        }
      ],
      \"StackResourceDriftStatus\": \"MODIFIED\",
      \"Timestamp\": \"2019-10-02T05:54:44.064Z\"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeStackResourceDrifts](#) in der AWS CLI Befehlsreferenz.

describe-stack-resource

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-resource`.

AWS CLI

Um Informationen über eine Stack-Ressource zu erhalten

Im folgenden `describe-stack-resource` Beispiel werden Details für die Ressource angezeigt, die `MyFunction` im angegebenen Stack benannt ist.

```

aws cloudformation describe-stack-resource \
  --stack-name MyStack \
  --logical-resource-id MyFunction

```

Ausgabe:

```
{
  "StackResourceDetail": {
    "StackName": "MyStack",
    "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "MyFunction",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "Metadata": "{}",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
}
```

- API-Einheiten finden Sie [DescribeStackResource](#) unter AWS CLI Befehlsreferenz.

describe-stack-resources

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-resources`.

AWS CLI

Um Informationen über eine Stack-Ressource zu erhalten

Im folgenden `describe-stack-resources` Beispiel werden Details zu den Ressourcen im angegebenen Stack angezeigt.

```
aws cloudformation describe-stack-resources \
  --stack-name my-stack
```

Ausgabe:

```
{
  "StackResources": [
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

    "LogicalResourceId": "bucket",
    "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
    "ResourceType": "AWS::S3::Bucket",
    "Timestamp": "2019-10-02T04:34:11.345Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "StackName": "my-stack",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "function",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "Timestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "StackName": "my-stack",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "functionRole",
    "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
    "ResourceType": "AWS::IAM::Role",
    "Timestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
}

```

- API-Einheiten finden Sie [DescribeStackResources](#) unter AWS CLI Befehlsreferenz.

describe-stack-set-operation

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-set-operation`.

AWS CLI

Um Informationen über einen Stack-Set-Vorgang zu erhalten

Das folgende `describe-stack-set-operation` -Beispiel zeigt Details für einen Aktualisierungsvorgang auf dem angegebenen Stack-Set an.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name enable-config \  
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

Ausgabe:

```
{  
  "StackSetOperation": {  
    "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",  
    "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",  
    "Action": "UPDATE",  
    "Status": "SUCCEEDED",  
    "OperationPreferences": {  
      "RegionOrder": [  
        "us-east-1",  
        "us-west-2",  
        "eu-west-1",  
        "us-west-1"  
      ],  
      "FailureToleranceCount": 7,  
      "MaxConcurrentCount": 2  
    },  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",  
    "CreationTimestamp": "2019-10-03T16:28:44.377Z",  
    "EndTimestamp": "2019-10-03T16:42:08.607Z"  
  }  
}
```

- API-Einheiten finden Sie [DescribeStackSetOperation](#) in der AWS CLI Befehlsreferenz.

`describe-stack-set`

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-set`.

AWS CLI

Um Informationen über ein Stack-Set zu erhalten

Das folgende describe-stack-set -Beispiel zeigt Details über das angegebene Stack-Set an.

```
aws cloudformation describe-stack-set \  
  --stack-set-name my-stack-set
```

Ausgabe:

```
{  
  "StackSet": {  
    "StackSetName": "my-stack-set",  
    "StackSetId": "my-stack-set:296a3360-xmpl-40af-be78-9341e95bf743",  
    "Description": "Create an Amazon SNS topic",  
    "Status": "ACTIVE",  
    "TemplateBody": "AWSTemplateFormatVersion: '2010-09-09'\nDescription: An AWS  
SNS topic\nResources:\n  topic:\n    Type: AWS::SNS::Topic",  
    "Parameters": [],  
    "Capabilities": [],  
    "Tags": [],  
    "StackSetARN": "arn:aws:cloudformation:us-west-2:123456789012:stackset/  
enable-config:296a3360-xmpl-40af-be78-9341e95bf743",  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"  
  }  
}
```

- API-Einheiten finden Sie [DescribeStackSet](#) in der AWS CLI Befehlsreferenz.

describe-stacks

Das folgende Codebeispiel zeigt die Verwendung describe-stacks.

AWS CLI

Um AWS CloudFormation Stapel zu beschreiben

Der folgende describe-stacks Befehl zeigt zusammenfassende Informationen für den myteststack Stack:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Ausgabe:

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
      "Tags": [],
      "Outputs": [
        {
          "Description": "Name of S3 bucket to hold website content",
          "OutputKey": "BucketName",
          "OutputValue": "myteststack-s3bucket-jssofilzie2w"
        }
      ],
      "StackStatusReason": null,
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "CREATE_COMPLETE",
      "DisableRollback": false
    }
  ]
}
```

Weitere Informationen finden Sie unter Stacks im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeStacks](#) in der AWS CLI Befehlsreferenz.

describe-type-registration

Das folgende Codebeispiel zeigt die Verwendung `describe-type-registration`.

AWS CLI

Geben Sie Registrierungsinformationen ein, um anzuzeigen

Im folgenden `describe-type-registration` Beispiel werden Informationen zur angegebenen Typregistrierung angezeigt, einschließlich des aktuellen Status, des Typs und der Version des Typs.

```
aws cloudformation describe-type-registration \  
  --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "ProgressStatus": "COMPLETE",  
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup",  
  "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",  
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup/00000001"  
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der CloudFormation Registrierung](#).

- API-Einheiten finden Sie [DescribeTypeRegistration](#) unter AWS CLI Befehlsreferenz.

describe-type

Das folgende Codebeispiel zeigt die Verwendung `describe-type`.

AWS CLI

Um Typinformationen anzuzeigen

Im folgenden `describe-type` Beispiel werden Informationen für den angegebenen Typ angezeigt.

```
aws cloudformation describe-type \  
  --type-name My::Logs::LogGroup \  
  --type RESOURCE
```

Ausgabe:

```
{
```

```
"SourceUrl": "https://github.com/aws-cloudformation/aws-cloudformation-resource-providers-logs.git",
  "Description": "Customized resource derived from AWS::Logs::LogGroup",
  "TimeCreated": "2019-12-03T23:29:33.321Z",
  "Visibility": "PRIVATE",
  "TypeName": "My::Logs::LogGroup",
  "LastUpdated": "2019-12-03T23:29:33.321Z",
  "DeprecatedStatus": "LIVE",
  "ProvisioningType": "FULLY_MUTABLE",
  "Type": "RESOURCE",
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup/00000001",
  "Schema": "[details omitted]"
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter [Using the CloudFormation Registry](#).

- API-Einheiten finden Sie [DescribeType](#) unter AWS CLI Befehlsreferenz.

detect-stack-drift

Das folgende Codebeispiel zeigt die Verwendung `detect-stack-drift`.

AWS CLI

Um verschwendete Ressourcen zu erkennen

Das folgende `detect-stack-drift` Beispiel initiiert die Drifterkennung für den angegebenen Stack.

```
aws cloudformation detect-stack-drift \
  --stack-name my-stack
```

Ausgabe:

```
{
  "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"
}
```

Sie können diese ID dann zusammen mit dem `describe-stack-resource-drifts` Befehl verwenden, um driftete Ressourcen zu beschreiben.

- API-Einheiten finden Sie [DetectStackDrift](#) in der AWS CLI Befehlsreferenz.

detect-stack-resource-drift

Das folgende Codebeispiel zeigt die Verwendung `detect-stack-resource-drift`.

AWS CLI

Um Abweichungen bei einer Ressource zu erkennen

Im folgenden `detect-stack-resource-drift` Beispiel wird eine Ressource geprüft, die `MyFunction` in einem nach Drift benannten Stapel benannt `MyStack` ist:

```
aws cloudformation detect-stack-resource-drift \
  --stack-name MyStack \
  --logical-resource-id MyFunction
```

Die Ausgabe zeigt eine AWS Lambda-Funktion, die geändert wurde: `out-of-band`

```
{
  "StackResourceDrift": {
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/MyStack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "MyFunction",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "ExpectedProperties": "{\"Description\": \"Write a file to S3.\",
  \"Environment\": {\"Variables\": {\"bucket\": \"my-stack-bucket-1vc62xmplgguf\"}},
  \"Handler\": \"index.handler\", \"MemorySize\": 128, \"Role\": \"arn:aws:iam:123456789012:role/my-functionRole-HIZXMPLE0M9E\", \"Runtime\": \"nodejs10.x\", \"Tags\": [{\"Key\": \"lambda:createdBy\", \"Value\": \"SAM\"}], \"Timeout\": 900, \"TracingConfig\": {\"Mode\": \"Active\"}}",
    "ActualProperties": "{\"Description\": \"Write a file to S3.\", \"Environment\": {\"Variables\": {\"bucket\": \"my-stack-bucket-1vc62xmplgguf\"}}, \"Handler\": \"index.handler\", \"MemorySize\": 256, \"Role\": \"arn:aws:iam:123456789012:role/my-functionRole-HIZXMPLE0M9E\", \"Runtime\": \"nodejs10.x\", \"Tags\": [{\"Key\": \"lambda:createdBy\", \"Value\": \"SAM\"}], \"Timeout\": 22, \"TracingConfig\": {\"Mode\": \"Active\"}}",
    "PropertyDifferences": [
      {
        "PropertyPath": "/MemorySize",
        "ExpectedValue": "128",
```

```

        "ActualValue": "256",
        "DifferenceType": "NOT_EQUAL"
    },
    {
        "PropertyPath": "/Timeout",
        "ExpectedValue": "900",
        "ActualValue": "22",
        "DifferenceType": "NOT_EQUAL"
    }
],
"StackResourceDriftStatus": "MODIFIED",
"Timestamp": "2019-10-02T05:58:47.433Z"
}
}

```

- API-Einheiten finden Sie [DetectStackResourceDrift](#) in der AWS CLI Befehlsreferenz.

detect-stack-set-drift

Das folgende Codebeispiel zeigt die Verwendung `detect-stack-set-drift`.

AWS CLI

Um Drift auf einem Stack-Set und allen zugehörigen Stack-Instances zu erkennen

Das folgende `detect-stack-set-drift` Beispiel initiiert Drift-Erkennungsoperationen auf dem angegebenen Stack-Set, einschließlich aller Stack-Instances, die diesem Stack-Set zugeordnet sind, und gibt eine Operations-ID zurück, mit der der Status des Drift-Vorgangs verfolgt werden kann.

```

aws cloudformation detect-stack-set-drift \
  --stack-set-name stack-set-drift-example

```

Ausgabe:

```

{
  "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}

```

Weitere Informationen finden Sie unter [Erkennen nicht verwalteter Konfigurationsänderungen in Stack-Sets](#) im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [DetectStackSetDrift](#) in der AWS CLI Befehlsreferenz.

estimate-template-cost

Das folgende Codebeispiel zeigt die Verwendung `estimate-template-cost`.

AWS CLI

Um die Kosten der Vorlage zu schätzen

Im folgenden `estimate-template-cost` Beispiel wird ein Kostenvoranschlag für eine Vorlage generiert, die `template.yaml` im aktuellen Ordner benannt ist.

```
aws cloudformation estimate-template-cost \  
  --template-body file://template.yaml
```

Ausgabe:

```
{  
  "Url": "http://calculator.s3.amazonaws.com/calc5.html?  
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"  
}
```

- API-Einheiten finden Sie [EstimateTemplateCost](#) unter AWS CLI Befehlsreferenz.

execute-change-set

Das folgende Codebeispiel zeigt die Verwendung `execute-change-set`.

AWS CLI

Um einen Änderungssatz auszuführen

Im folgenden `execute-change-set` Beispiel wird ein Änderungssatz ausgeführt, der durch den Namen des Änderungssatzes und den Stacknamen angegeben ist.

```
aws cloudformation execute-change-set \  
  --change-set-name my-change-set \  
  --stack-name my-stack
```

Im folgenden `execute-change-set` Beispiel wird ein durch den gesamten Änderungssatz spezifizierter ARN Änderungssatz ausgeführt.

```
aws cloudformation execute-change-set \
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
  change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- API-Einheiten finden Sie unter [ExecuteChangeSet AWS CLI](#) Befehlsreferenz.

get-stack-policy

Das folgende Codebeispiel zeigt die Verwendung `get-stack-policy`.

AWS CLI

Um eine Stack-Richtlinie anzuzeigen

Das folgende `get-stack-policy` Beispiel zeigt die Stack-Richtlinie für den angegebenen Stack. Verwenden Sie den `set-stack-policy` Befehl, um eine Richtlinie an einen Stack anzuhängen.

```
aws cloudformation get-stack-policy \
  --stack-name my-stack
```

Ausgabe:

```
{
  "StackPolicyBody": "{\n  \"Statement\" : [\n    {\n      \"Effect\" :\n  \"Allow\",\n      \"Action\" : \"Update:*\",\n      \"Principal\" : \"*\",\n      \"Resource\" : \"*\"\n    },\n    {\n      \"Effect\" : \"Deny\",\n      \"Action\" : \"Update:*\",\n      \"Principal\" : \"*\",\n      \"Resource\" :\n  \"LogicalResourceId/bucket\"\n    }\n  ]\n}"
```

- API-Einheiten finden Sie [GetStackPolicy](#) in der AWS CLI Befehlsreferenz.

get-template-summary

Das folgende Codebeispiel zeigt die Verwendung `get-template-summary`.

AWS CLI

Um eine Vorlagenzusammenfassung anzuzeigen

Der folgende Befehl zeigt zusammenfassende Informationen zu den Ressourcen und Metadaten für die angegebene Vorlagendatei an.

```
aws cloudformation get-template-summary \  
--template-body file://template.yaml
```

Ausgabe:

```
{  
  "Parameters": [],  
  "Description": "A VPC and subnets.",  
  "ResourceTypes": [  
    "AWS::EC2::VPC",  
    "AWS::EC2::Subnet",  
    "AWS::EC2::Subnet",  
    "AWS::EC2::RouteTable",  
    "AWS::EC2::VPCEndpoint",  
    "AWS::EC2::SubnetRouteTableAssociation",  
    "AWS::EC2::SubnetRouteTableAssociation",  
    "AWS::EC2::VPCEndpoint"  
  ],  
  "Version": "2010-09-09"  
}
```

- API-Einheiten finden Sie [GetTemplateSummary](#) unter AWS CLI Befehlsreferenz.

get-template

Das folgende Codebeispiel zeigt die Verwendung `get-template`.

AWS CLI

Um den Vorlagentext für einen AWS CloudFormation Stapel anzuzeigen

Der folgende `get-template` Befehl zeigt die Vorlage für den `myteststack` Stack:

```
aws cloudformation get-template --stack-name myteststack
```

Ausgabe:

```
{
  "TemplateBody": {
    "AWSTemplateFormatVersion": "2010-09-09",
    "Outputs": {
      "BucketName": {
        "Description": "Name of S3 bucket to hold website content",
        "Value": {
          "Ref": "S3Bucket"
        }
      }
    },
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
    "Resources": {
      "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
          "AccessControl": "PublicRead"
        }
      }
    }
  }
}
```

- API-Einheiten finden Sie [GetTemplate](#) in der AWS CLI Befehlsreferenz.

list-change-sets

Das folgende Codebeispiel zeigt die Verwendung `list-change-sets`.

AWS CLI

Um Änderungssätze aufzulisten

Im folgenden `list-change-sets` Beispiel wird eine Liste der ausstehenden Änderungssätze für den angegebenen Stack angezeigt.

```
aws cloudformation list-change-sets \
```

```
--stack-name my-stack
```

Ausgabe:

```
{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "StackName": "my-stack",
      "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/
my-change-set/70160340-7914-xmpl-bcbf-128a1fa78b5d",
      "ChangeSetName": "my-change-set",
      "ExecutionStatus": "AVAILABLE",
      "Status": "CREATE_COMPLETE",
      "CreationTime": "2019-10-02T05:38:54.297Z"
    }
  ]
}
```

- API-Einheiten finden Sie [ListChangeSets](#) unter AWS CLI Befehlsreferenz.

list-exports

Das folgende Codebeispiel zeigt die Verwendung `list-exports`.

AWS CLI

Um Exporte aufzulisten

Im folgenden `list-exports` Beispiel wird eine Liste der Exporte aus Stapeln in der aktuellen Region angezeigt.

```
aws cloudformation list-exports
```

Ausgabe:

```
{
  "Exports": [
    {
      "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
```

```

        "Name": "private-vpc-subnet-a",
        "Value": "subnet-07b410xmplddcfa03"
    },
    {
        "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
        "Name": "private-vpc-subnet-b",
        "Value": "subnet-075ed3xmplabd2fb1"
    },
    {
        "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
        "Name": "private-vpc-vpcid",
        "Value": "vpc-011d7xmpl1100e9841"
    }
]
}

```

- API-Einheiten finden Sie unter [ListExports AWS CLI Befehlsreferenz](#).

list-imports

Das folgende Codebeispiel zeigt die Verwendung `list-imports`.

AWS CLI

Um Importe aufzulisten

Das folgende `list-imports` Beispiel listet die Stapel auf, die den angegebenen Export importieren. Verwenden Sie den `list-exports` Befehl, um die Liste der verfügbaren Exporte abzurufen.

```

aws cloudformation list-imports \
  --export-name private-vpc-vpcid

```

Ausgabe:

```

{
  "Imports": [
    "my-database-stack"
  ]
}

```



```
}

```

- API-Einheiten finden Sie [ListImports](#) in der AWS CLI Befehlsreferenz.

list-stack-instances

Das folgende Codebeispiel zeigt die Verwendung `list-stack-instances`.

AWS CLI

Um Instanzen für einen Stack aufzulisten

Das folgende `list-stack-instances` Beispiel listet die Instanzen auf, die aus dem angegebenen Stack-Set erstellt wurden.

```
aws cloudformation list-stack-instances \
  --stack-set-name enable-config
```

Die Beispielausgabe enthält Details zu einem Stack, der aufgrund eines Fehlers nicht aktualisiert werden konnte:

```
{
  "Summaries": [
    {
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
      "Region": "us-west-2",
      "Account": "123456789012",
      "StackId": "arn:aws:cloudformation:ap-northeast-1:123456789012:stack/StackSet-enable-config-35a6ac50-d9f8-4084-86e4-7da34d5de4c4/a1631cd0-e5fb-xmpl-b474-0aa20f14f06e",
      "Status": "CURRENT"
    },
    {
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
      "Region": "us-west-2",
      "Account": "123456789012",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/eab53680-e5fa-xmpl-ba14-0a522351f81e",
      "Status": "OUTDATED",
      "StatusReason": "ResourceLogicalId:ConfigDeliveryChannel, ResourceLogicalId:ConfigDeliveryChannel, ResourceType:AWS::Config::DeliveryChannel, ResourceStatusReason:Failed to put"
    }
  ]
}
```

```

delivery channel 'StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532-
ConfigDeliveryChannel-10JWJ7XD59WR0' because the maximum number of delivery
channels: 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code:
MaxNumberOfDeliveryChannelsExceededException; Request ID: d14b34a0-ef7c-xmpl-
acf8-8a864370ae56)."
```

```

    }
  ]
}
```

- API-Einheiten finden Sie [ListStackInstances](#) in der AWS CLI Befehlsreferenz.

list-stack-resources

Das folgende Codebeispiel zeigt die Verwendung `list-stack-resources`.

AWS CLI

Um Ressourcen in einem Stapel aufzulisten

Der folgende Befehl zeigt die Liste der Ressourcen im angegebenen Stack an.

```
aws cloudformation list-stack-resources \
  --stack-name my-stack
```

Ausgabe:

```
{
  "StackResourceSummaries": [
    {
      "LogicalResourceId": "bucket",
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
      "ResourceType": "AWS::S3::Bucket",
      "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    },
    {
      "LogicalResourceId": "function",
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
      "ResourceType": "AWS::Lambda::Function",
```

```

    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "LogicalResourceId": "functionRole",
    "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
    "ResourceType": "AWS::IAM::Role",
    "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
}

```

- API-Einheiten finden Sie [ListStackResources](#) in der AWS CLI Befehlsreferenz.

list-stack-set-operation-results

Das folgende Codebeispiel zeigt die Verwendung `list-stack-set-operation-results`.

AWS CLI

Um die Ergebnisse von Stack-Set-Operationen aufzulisten

Der folgende Befehl zeigt die Ergebnisse eines Aktualisierungsvorgangs für Instanzen im angegebenen Stack-Set an.

```

aws cloudformation list-stack-set-operation-results \
  --stack-set-name enable-config \
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0

```

Ausgabe:

```

{
  "Summaries": [
    {
      "Account": "223456789012",

```

```

    "Region": "us-west-2",
    "Status": "SUCCEEDED",
    "AccountGateResult": {
      "Status": "SKIPPED",
      "StatusReason": "Function not found: arn:aws:lambda:eu-
west-1:223456789012:function:AWSCloudFormationStackSetAccountGate"
    }
  },
  {
    "Account": "223456789012",
    "Region": "ap-south-1",
    "Status": "CANCELLED",
    "StatusReason": "Cancelled since failure tolerance has exceeded"
  }
]
}

```

Hinweis: Der SKIPPED Status für AccountGateResult wird für erfolgreiche Operationen erwartet, sofern Sie keine Account-Gate-Funktion erstellen.

- API-Einheiten finden Sie [ListStackSetOperationResults](#) in der AWS CLI Befehlsreferenz.

list-stack-set-operations

Das folgende Codebeispiel zeigt die Verwendung `list-stack-set-operations`.

AWS CLI

Um Stack-Set-Operationen aufzulisten

Im folgenden `list-stack-set-operations` Beispiel wird die Liste der letzten Operationen auf dem angegebenen Stack-Set angezeigt.

```

aws cloudformation list-stack-set-operations \
  --stack-set-name my-stack-set

```

Ausgabe:

```

{
  "Summaries": [
    {
      "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",

```

```

    "Action": "UPDATE",
    "Status": "SUCCEEDED",
    "CreationTimestamp": "2019-10-03T16:28:44.377Z",
    "EndTimestamp": "2019-10-03T16:42:08.607Z"
  },
  {
    "OperationId": "891aa98f-7118-xmpl-00b2-00954d1dd0d6",
    "Action": "UPDATE",
    "Status": "FAILED",
    "CreationTimestamp": "2019-10-03T15:43:53.916Z",
    "EndTimestamp": "2019-10-03T15:45:58.925Z"
  }
]
}

```

- API-Einheiten finden Sie [ListStackSetOperations](#) unter AWS CLI Befehlsreferenz.

list-stack-sets

Das folgende Codebeispiel zeigt die Verwendung `list-stack-sets`.

AWS CLI

Um Stack-Sets aufzulisten

Im folgenden `list-stack-sets` Beispiel wird die Liste der Stack-Sets in der aktuellen Region und im aktuellen Konto angezeigt.

```
aws cloudformation list-stack-sets
```

Ausgabe:

```

{
  "Summaries": [
    {
      "StackSetName": "enable-config",
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
      "Description": "Enable AWS Config",
      "Status": "ACTIVE"
    }
  ]
}

```

- API-Einheiten finden Sie [ListStackSets](#) in der AWS CLI Befehlsreferenz.

list-stacks

Das folgende Codebeispiel zeigt die Verwendung `list-stacks`.

AWS CLI

Um AWS CloudFormation Stapel aufzulisten

Der folgende `list-stacks` Befehl zeigt eine Zusammenfassung aller Stapel mit dem Status: `CREATE_COMPLETE`

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Ausgabe:

```
[
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
    "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
    "StackStatusReason": null,
    "CreationTime": "2013-08-26T03:27:10.190Z",
    "StackName": "myteststack",
    "StackStatus": "CREATE_COMPLETE"
  }
]
```

- API-Einheiten finden Sie [ListStacks](#) in der AWS CLI Befehlsreferenz.

list-type-registrations

Das folgende Codebeispiel zeigt die Verwendung `list-type-registrations`.

AWS CLI

Um die abgeschlossenen Registrierungen eines Typs aufzulisten

Im folgenden `list-type-registrations` Beispiel wird eine Liste der abgeschlossenen Typregistrierungen für den angegebenen Typ angezeigt.

```
aws cloudformation list-type-registrations \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --registration-status-filter COMPLETE
```

Ausgabe:

```
{  
  "RegistrationTokenList": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
  ]  
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der CloudFormation Registrierung](#).

- API-Einheiten finden Sie [ListTypeRegistrations](#) unter AWS CLI Befehlsreferenz.

list-type-versions

Das folgende Codebeispiel zeigt die Verwendung `list-type-versions`.

AWS CLI

Um die Version einer Erweiterung aufzulisten

Im folgenden `list-type-versions` Beispiel werden zusammenfassende Informationen zu den Versionen einer Erweiterung zurückgegeben.

```
aws cloudformation list-type-versions \  
  --endpoint https://example.com \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name My::Resource::Example \  
  --publisher-id 123456789012
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [ListTypeVersions](#) unter AWS CLI Befehlsreferenz.

list-types

Das folgende Codebeispiel zeigt die Verwendung `list-types`.

AWS CLI

Um die privaten Ressourcentypen in einem Konto aufzulisten

Im folgenden `list-types` Beispiel wird eine Liste der privaten Ressourcentypen angezeigt, die derzeit im aktuellen AWS Konto registriert sind.

```
aws cloudformation list-types
```

Ausgabe:

```
{
  "TypeSummaries": [
    {
      "Description": "WordPress blog resource for internal use",
      "LastUpdated": "2019-12-04T18:28:15.059Z",
      "TypeName": "My::WordPress::BlogExample",
      "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-WordPress-BlogExample",
      "DefaultVersionId": "00000005",
      "Type": "RESOURCE"
    },
    {
      "Description": "Customized resource derived from AWS::Logs::LogGroup",
      "LastUpdated": "2019-12-04T18:28:15.059Z",
      "TypeName": "My::Logs::LogGroup",
      "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup",
      "DefaultVersionId": "00000003",
      "Type": "RESOURCE"
    }
  ]
}
```



```
]
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der CloudFormation Registrierung](#).

- API-Einheiten finden Sie [ListTypes](#) unter AWS CLI Befehlsreferenz.

package

Das folgende Codebeispiel zeigt die Verwendung `package`.

AWS CLI

Der folgende Befehl exportiert eine Vorlage, die `template.json` nach dem Hochladen lokaler Artefakte in den S3-Bucket benannt wurde, `bucket-name` und schreibt die exportierte Vorlage in: `packaged-template.json`

```
aws cloudformation package --template-file /path_to_template/template.json --s3-bucket bucket-name --output-template-file packaged-template.json --use-json
```

- API-Einheiten finden Sie unter [Package](#) in der AWS CLI Befehlsreferenz.

publish-type

Das folgende Codebeispiel zeigt die Verwendung `publish-type`.

AWS CLI

Um eine Erweiterung zu veröffentlichen

Im folgenden `publish-type` Beispiel wird die angegebene Erweiterung in der CloudFormation Registrierung als öffentliche Erweiterung in dieser Region veröffentlicht.

```
aws cloudformation publish-type \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0
```

Ausgabe:

```
{
  "PublicTypeArn": "arn:aws:cloudformation:us-west-2::type/
resource/000q6TfUovXsEMmgKowxDZLLwqr2QUshd2e75c8c/Example-
Test-1234567890abcdef0/1.0.0"
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [PublishType](#) unter AWS CLI Befehlsreferenz.

register-publisher

Das folgende Codebeispiel zeigt die Verwendung `register-publisher`.

AWS CLI

Um einen Herausgeber zu registrieren

Im folgenden `register-publisher` Beispiel wird ein Herausgeber registriert und der Parameter Terms and Condition akzeptiert.

```
aws cloudformation register-publisher \
  --region us-west-2 \
  --accept-terms-and-conditions
```

Ausgabe:

```
{
  "PublisherId": "000q6TfUovXsEMmgKowxDZLLwqr2QUshd2e75c8c"
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [RegisterPublisher](#) unter AWS CLI Befehlsreferenz.

register-type

Das folgende Codebeispiel zeigt die Verwendung `register-type`.

AWS CLI

Um einen Ressourcentyp zu registrieren

Im folgenden `register-type` Beispiel wird der angegebene Ressourcentyp als privater Ressourcentyp im Benutzerkonto registriert.

```
aws cloudformation register-type \  
  --type-name My::Organization::ResourceName \  
  --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \  
  --type RESOURCE
```

Ausgabe:

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"  
}
```

Weitere Informationen finden Sie unter [Registrierung von Ressourcenanbietern](#) im Benutzerhandbuch für die CloudFormation Befehlszeilenschnittstelle zur Typentwicklung.

- API-Einheiten finden Sie [RegisterType](#) unter AWS CLI Befehlsreferenz.

set-stack-policy

Das folgende Codebeispiel zeigt die Verwendung `set-stack-policy`.

AWS CLI

Um eine Stack-Richtlinie anzuwenden

Im folgenden `set-stack-policy` Beispiel werden Updates für die angegebene Ressource im angegebenen Stack deaktiviert. `stack-policy.json` ist ein JSON-Dokument, das die Operationen definiert, die für Ressourcen im Stapel zulässig sind.

```
aws cloudformation set-stack-policy \  
  --stack-name my-stack \  
  --stack-policy-body file://stack-policy.json
```

Ausgabe:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/bucket"
    }
  ]
}
```

- API-Einheiten finden Sie [SetStackPolicy](#) in der AWS CLI Befehlsreferenz.

set-type-configuration

Das folgende Codebeispiel zeigt die Verwendung `set-type-configuration`.

AWS CLI

Um Daten zu konfigurieren

Im folgenden `set-type-configuration` Beispiel werden die Konfigurationsdaten für eine registrierte CloudFormation Erweiterung im angegebenen Konto und in der angegebenen Region angegeben.

```
aws cloudformation set-type-configuration \
  --region us-west-2 \
  --type RESOURCE \
  --type-name Example::Test::Type \
  --configuration-alias default \
  --configuration "{\"CredentialKey\": \"testUserCredential\"}"
```

Ausgabe:

```
{
```

```
"ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-configuration/resource/Example-Test-Type/default"
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- APIEinzelheiten finden Sie [SetTypeConfiguration](#) unter AWS CLI Befehlsreferenz.

set-type-default-version

Das folgende Codebeispiel zeigt die Verwendung `set-type-default-version`.

AWS CLI

Um die Standardversion eines Typs festzulegen

Im folgenden `set-type-default-version` Beispiel wird festgelegt, dass die angegebene Typversion als Standard für diesen Typ verwendet wird.

```
aws cloudformation set-type-default-version \
  --type RESOURCE \
  --type-name My::Logs::LogGroup \
  --version-id 00000003
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter [Using the CloudFormation Registry](#).

- APIEinzelheiten finden Sie [SetTypeDefaultVersion](#) unter AWS CLI Befehlsreferenz.

signal-resource

Das folgende Codebeispiel zeigt die Verwendung `signal-resource`.

AWS CLI

Um eine Ressource zu signalisieren

Das folgende `signal-resource` Beispiel signalisiert `success`, dass die `MyWaitCondition` im genannten Stapel angegebene Wartebedingung erfüllt werden soll `my-stack`.

```
aws cloudformation signal-resource \  
  --stack-name my-stack \  
  --logical-resource-id MyWaitCondition \  
  --unique-id 1234 \  
  --status SUCCESS
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SignalResource](#) in der AWS CLI Befehlsreferenz.

stop-stack-set-operation

Das folgende Codebeispiel zeigt die Verwendung `stop-stack-set-operation`.

AWS CLI

Um einen Stack-Set-Vorgang zu beenden

Das folgende `stop-stack-set-operation` Beispiel stoppt einen laufenden Aktualisierungsvorgang für das angegebene Stack-Set.

```
aws cloudformation stop-stack-set-operation \  
  --stack-set-name my-stack-set \  
  --operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [StopStackSetOperation](#) in AWS CLI der Befehlsreferenz.

test-type

Das folgende Codebeispiel zeigt die Verwendung `test-type`.

AWS CLI

Um eine Erweiterung zu testen

Im folgenden `test-type` Beispiel wird eine registrierte Erweiterung getestet, um sicherzustellen, dass sie alle erforderlichen Anforderungen für die Veröffentlichung in der CloudFormation Registrierung erfüllt.

```
aws cloudformation test-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001
```

Ausgabe:

```
{  
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/  
Sample-Test-Resource123/00000001"  
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch [unter Verwenden der AWS CloudFormation Registrierung](#).

- API-Einheiten finden Sie [TestType](#) unter AWS CLI Befehlsreferenz.

update-stack-instances

Das folgende Codebeispiel zeigt die Verwendung `update-stack-instances`.

AWS CLI

Um Stack-Instances zu aktualisieren

Im folgenden `update-stack-instances` Beispiel wird erneut versucht, Stack-Instances in zwei Konten in zwei Regionen mit den neuesten Einstellungen zu aktualisieren. Die angegebene Einstellung für die Fehlertoleranz stellt sicher, dass das Update in allen Konten und Regionen versucht wird, auch wenn einige Stacks nicht aktualisiert werden können.

```
aws cloudformation update-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 567890123456 \  
  --regions us-east-1 us-west-2 \  
  --operation-preferences FailureToleranceCount=3
```

Ausgabe:

```
{  
  "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"  
}
```

- API-Einheiten finden Sie [UpdateStackInstances](#) in der AWS CLI Befehlsreferenz.

update-stack-set

Das folgende Codebeispiel zeigt die Verwendung `update-stack-set`.

AWS CLI

Um ein Stack-Set zu aktualisieren

Im folgenden `update-stack-set` Beispiel wird den Stack-Instances im angegebenen Stack-Set ein Tag mit `IT` dem Schlüsselnamen `Owner` und dem Wert `IT` hinzugefügt.

```
aws cloudformation update-stack-set \  
  --stack-set-name my-stack-set \  
  --use-previous-template \  
  --tags Key=Owner,Value=IT
```

Ausgabe:

```
{  
  "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"  
}
```

- API-Einheiten finden Sie [UpdateStackSet](#) in der AWS CLI Befehlsreferenz.

update-stack

Das folgende Codebeispiel zeigt die Verwendung `update-stack`.

AWS CLI

Um AWS CloudFormation Stacks zu aktualisieren

Der folgende `update-stack` Befehl aktualisiert die Vorlage und die Eingabeparameter für den `mystack` Stack:

```
aws cloudformation update-stack --stack-name mystack --  
  template-url https://s3.amazonaws.com/sample/updated.template --
```



```
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

Der folgende `update-stack` Befehl aktualisiert nur den SubnetIDs Parameterwert für den `mystack` Stack. Wenn Sie keinen Parameterwert angeben, wird der in der Vorlage angegebene Standardwert verwendet:

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template  
--parameters ParameterKey=KeyPairName,UsePreviousValue=true  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

Mit dem folgenden `update-stack` Befehl werden dem Stack zwei Themen für `mystack` Stack-Benachrichtigungen hinzugefügt:

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --  
notification-arns "arn:aws:sns:use-east-1:123456789012:mytopic1" "arn:aws:sns:us-  
east-1:123456789012:mytopic2"
```

Weitere Informationen finden Sie unter [AWS CloudFormation Stack-Updates](#) im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateStack](#) in der AWS CLI Befehlsreferenz.

update-termination-protection

Das folgende Codebeispiel zeigt die Verwendung `update-termination-protection`.

AWS CLI

Um den Kündigungsschutz zu aktivieren

Im folgenden `update-termination-protection` Beispiel wird der Terminierungsschutz für den angegebenen Stack aktiviert.

```
aws cloudformation update-termination-protection \  
--stack-name my-stack \  
--enable-termination-protection
```

Ausgabe:

```
{
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- API-Einheiten finden Sie [UpdateTerminationProtection](#) in der AWS CLI Befehlsreferenz.

validate-template

Das folgende Codebeispiel zeigt die Verwendung `validate-template`.

AWS CLI

Um eine AWS CloudFormation Vorlage zu validieren

Der folgende `validate-template` Befehl validiert die `sampletemplate.json` Vorlage:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Ausgabe:

```
{
  "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
showing how to create a publicly accessible S3 bucket. **WARNING** This template
creates an S3 bucket. You will be billed for the AWS resources used if you create a
stack from this template.",
  "Parameters": [],
  "Capabilities": []
}
```

Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter [Arbeiten mit AWS CloudFormation Vorlagen](#).

- API-Einheiten finden Sie [ValidateTemplate](#) in der AWS CLI Befehlsreferenz.

CloudFront Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CloudFront.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-cloud-front-origin-access-identity

Das folgende Codebeispiel zeigt die Verwendung `create-cloud-front-origin-access-identity`.

AWS CLI

Um eine CloudFront ursprüngliche Zugriffsidentität zu erstellen

Im folgenden Beispiel wird eine ursprüngliche CloudFront Zugriffsidentität (OAI) erstellt, indem die OAI Konfiguration als Befehlszeilenargument bereitgestellt wird:

```
aws cloudfront create-cloud-front-origin-access-identity \  
  --cloud-front-origin-access-identity-config \  
    CallerReference="cli-example",Comment="Example OAI"
```

Sie können dasselbe erreichen, indem Sie die OAI Konfiguration in einer JSON Datei angeben, wie im folgenden Beispiel gezeigt:

```
aws cloudfront create-cloud-front-origin-access-identity \  
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

Die Datei `OAI-config.json` ist ein JSON Dokument im aktuellen Verzeichnis, das Folgendes enthält:

```
{  
  "CallerReference": "cli-example",
```

```
"Comment": "Example OAI"
}
```

Unabhängig davon, ob Sie die OAI Konfiguration mit einem Befehlszeilenargument oder einer JSON Datei angeben, ist die Ausgabe dieselbe:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/origin-access-identity/
cloudfront/E74FTE3AEXAMPLE",
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI"
    }
  }
}
```

- API Einheiten finden Sie [CreateCloudFrontOriginAccessIdentity](#) unter AWS CLI Befehlsreferenz.

create-distribution-with-tags

Das folgende Codebeispiel zeigt, wie man es benutzt `create-distribution-with-tags`.

AWS CLI

Um eine CloudFront Distribution mit Tags zu erstellen

Im folgenden Beispiel wird eine Distribution mit zwei Tags erstellt, indem die Verteilungskonfiguration und die Tags in einer JSON Datei mit dem Namen bereitgestellt werden `dist-config-with-tags.json`:

```
aws cloudfront create-distribution-with-tags \
  --distribution-config-with-tags file://dist-config-with-tags.json
```

Die Datei `dist-config-with-tags.json` ist ein JSON Dokument im aktuellen Ordner, das Folgendes enthält. Beachten Sie das Tags Objekt oben in der Datei, das zwei Tags enthält:

Name = ExampleDistributionProject = ExampleProject

```
{
  "Tags": {
    "Items": [
      {
        "Key": "Name",
        "Value": "ExampleDistribution"
      },
      {
        "Key": "Project",
        "Value": "ExampleProject"
      }
    ]
  },
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
          "DomainName": "awsexamplebucket.s3.amazonaws.com",
          "OriginPath": "",
          "CustomHeaders": {
            "Quantity": 0
          },
          "S3OriginConfig": {
            "OriginAccessIdentity": ""
          }
        }
      ]
    },
    "OriginGroups": {
      "Quantity": 0
    },
    "DefaultCacheBehavior": {
      "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
      "ForwardedValues": {
        "QueryString": false,

```

```
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
}
```

```

    },
    "Comment": "",
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}

```

Ausgabe:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/EDFDVBD6EXAMPLE",
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  }
}

```

```
  },
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
          "DomainName": "awsexamplebucket.s3.amazonaws.com",
          "OriginPath": "",
          "CustomHeaders": {
            "Quantity": 0
          },
          "S3OriginConfig": {
            "OriginAccessIdentity": ""
          }
        }
      ]
    },
    "OriginGroups": {
      "Quantity": 0
    },
    "DefaultCacheBehavior": {
      "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
      "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
          "Forward": "none"
        },
        "Headers": {
          "Quantity": 0
        },
        "QueryStringCacheKeys": {
          "Quantity": 0
        }
      },
      "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
      }
    },
  },
}
```



```
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ],
      "CachedMethods": {
        "Quantity": 2,
        "Items": [
          "HEAD",
          "GET"
        ]
      }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
      "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
  "Enabled": true,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
```

```

    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}

```

- API-Einheiten finden Sie [CreateDistributionWithTags](#) in der AWS CLI Befehlsreferenz.

create-distribution

Das folgende Codebeispiel zeigt, wie man es benutzt `create-distribution`.

AWS CLI

Um eine CloudFront Distribution zu erstellen

Im folgenden Beispiel wird eine Distribution für einen S3-Bucket mit dem Namen `awsexamplebucket` erstellt und außerdem mithilfe von Befehlszeilenargumenten `index.html` als Standard-Root-Objekt angegeben:

```

aws cloudfront create-distribution \
  --origin-domain-name awsexamplebucket.s3.amazonaws.com \
  --default-root-object index.html

```

Anstatt Befehlszeilenargumente zu verwenden, können Sie die Distributionskonfiguration in einer JSON Datei angeben, wie im folgenden Beispiel gezeigt:

```

aws cloudfront create-distribution \
  --distribution-config file://dist-config.json

```

Die Datei `dist-config.json` ist ein JSON Dokument im aktuellen Ordner, das Folgendes enthält:

```
{
  "CallerReference": "cli-example",
  "Aliases": {
    "Quantity": 0
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      },
      "Headers": {
        "Quantity": 0
      },
      "QueryStringCacheKeys": {
        "Quantity": 0
      }
    },
    "TrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
}
```

```
"ViewerProtocolPolicy": "allow-all",
"MinTTL": 0,
"AllowedMethods": {
  "Quantity": 2,
  "Items": [
    "HEAD",
    "GET"
  ],
  "CachedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ]
  }
},
"SmoothStreaming": false,
"DefaultTTL": 86400,
"MaxTTL": 31536000,
"Compress": false,
"LambdaFunctionAssociations": {
  "Quantity": 0
},
"FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
  "CloudFrontDefaultCertificate": true,
  "MinimumProtocolVersion": "TLSv1",
  "CertificateSource": "cloudfront"
```

```

},
"Restrictions": {
  "GeoRestriction": {
    "RestrictionType": "none",
    "Quantity": 0
  }
},
"WebACLId": "",
"HttpVersion": "http2",
"IsIPV6Enabled": true
}

```

Unabhängig davon, ob Sie die Verteilungsinformationen mit einem Befehlszeilenargument oder einer JSON Datei angeben, ist die Ausgabe dieselbe:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EMLARXS9EXAMPLE",
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-11-22T00:55:15.705Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    }
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",

```

```
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    }
]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
        "Headers": {
            "Quantity": 0
        },
    },
    "QueryStringCacheKeys": {
        "Quantity": 0
    }
},
"TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
},
"ViewerProtocolPolicy": "allow-all",
"MinTTL": 0,
"AllowedMethods": {
    "Quantity": 2,
    "Items": [
        "HEAD",
        "GET"
    ],
    "CachedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ]
    }
}
```

```
    }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
      "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
  "Enabled": true,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLId": "",
  "HttpVersion": "http2",
  "IsIPV6Enabled": true
}
}
```

- API-Einheiten finden Sie [CreateDistribution](#) unter AWS CLI Befehlsreferenz.

create-field-level-encryption-config

Das folgende Codebeispiel zeigt, wie man es benutzt `create-field-level-encryption-config`.

AWS CLI

Um eine Verschlüsselungskonfiguration auf CloudFront Feldebene zu erstellen

Im folgenden Beispiel wird eine Verschlüsselungskonfiguration auf Feldebene erstellt, indem die Konfigurationsparameter in einer Datei mit dem Namen bereitgestellt werden. JSON `fle-config.json` Bevor Sie eine Verschlüsselungskonfiguration auf Feldebene erstellen können, müssen Sie über ein Verschlüsselungsprofil auf Feldebene verfügen. Informationen zum Erstellen eines Profils finden Sie unter dem Befehl `-profile`. `create-field-level-encryption`

Weitere Informationen zur Verschlüsselung auf CloudFront Feldebene finden Sie unter Verschlüsselung auf [Feldebene zum Schutz vertraulicher Daten verwenden](#) im Amazon Developer Guide. CloudFront

```
aws cloudfront create-field-level-encryption-config \
  --field-level-encryption-config file://fle-config.json
```

Die Datei `fle-config.json` ist ein JSON Dokument im aktuellen Ordner, das Folgendes enthält:

```
{
  "CallerReference": "cli-example",
  "Comment": "Example FLE configuration",
  "QueryArgProfileConfig": {
    "ForwardWhenQueryArgProfileIsUnknown": true,
    "QueryArgProfiles": {
      "Quantity": 0
    }
  },
  "ContentTypeProfileConfig": {
    "ForwardWhenContentTypeIsUnknown": true,
    "ContentTypeProfiles": {
      "Quantity": 1,
      "Items": [
```



```

        {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSYOCVU",
            "ContentType": "application/x-www-form-urlencoded"
        }
    ]
}

```

Ausgabe:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption/
C3KM2WVD605UAY",
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryption": {
    "Id": "C3KM2WVD605UAY",
    "LastModifiedTime": "2019-12-10T21:30:18.974Z",
    "FieldLevelEncryptionConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example FLE configuration",
      "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
          "Quantity": 0,
          "Items": []
        }
      }
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSYOCVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}

```

```
}
}
```

- API-Einheiten finden Sie [CreateFieldLevelEncryptionConfig](#) in der AWS CLI Befehlsreferenz.

create-field-level-encryption-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `create-field-level-encryption-profile`.

AWS CLI

Um ein Verschlüsselungsprofil auf CloudFront Feldebene zu erstellen

Im folgenden Beispiel wird ein Verschlüsselungsprofil auf Feldebene erstellt, indem die Parameter in einer Datei mit dem Namen bereitgestellt werden. JSON `fle-profile-config.json` Bevor Sie ein Verschlüsselungsprofil auf Feldebene erstellen können, benötigen Sie einen öffentlichen Schlüssel. CloudFront Informationen zum Erstellen eines CloudFront öffentlichen Schlüssels finden Sie im `create-public-key` Befehl.

Weitere Informationen zur Verschlüsselung auf CloudFront Feldebene finden Sie unter Verschlüsselung auf [Feldebene zum Schutz vertraulicher Daten verwenden](#) im Amazon Developer Guide. CloudFront

```
aws cloudfront create-field-level-encryption-profile \
  --field-level-encryption-profile-config file://fle-profile-config.json
```

Die Datei `fle-profile-config.json` ist ein JSON Dokument im aktuellen Ordner, das Folgendes enthält:

```
{
  "Name": "ExampleFLEProfile",
  "CallerReference": "cli-example",
  "Comment": "FLE profile for AWS CLI example",
  "EncryptionEntities": {
    "Quantity": 1,
    "Items": [
      {
        "PublicKeyId": "K2K8NC4HVFE3M0",
        "ProviderId": "ExampleFLEProvider",
        "FieldPatterns": {
```

```

        "Quantity": 1,
        "Items": [
            "ExampleSensitiveField"
        ]
    }
}

```

Ausgabe:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption-
profile/PPK0UOSIF5WSV",
  "ETag": "E2QWRUHEXAMPLE",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0UOSIF5WSV",
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfile",
      "CallerReference": "cli-example",
      "Comment": "FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {
            "PublicKeyId": "K2K8NC4HVFE3M0",
            "ProviderId": "ExampleFLEProvider",
            "FieldPatterns": {
              "Quantity": 1,
              "Items": [
                "ExampleSensitiveField"
              ]
            }
          }
        ]
      }
    }
  }
}

```

- API-Einheiten finden Sie [CreateFieldLevelEncryptionProfile](#) in der AWS CLI Befehlsreferenz.

create-invalidation

Das folgende Codebeispiel zeigt, wie man es benutzt `create-invalidation`.

AWS CLI

Um eine Invalidierung für eine CloudFront Distribution zu erstellen

Das folgende `create-invalidation` Beispiel erstellt eine Invalidierung für die angegebenen Dateien in der angegebenen CloudFront Distribution:

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

Ausgabe:

```
{  
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/  
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",  
  "Invalidation": {  
    "Id": "I1JLWSDAP8FU89",  
    "Status": "InProgress",  
    "CreateTime": "2019-12-05T18:24:51.407Z",  
    "InvalidationBatch": {  
      "Paths": {  
        "Quantity": 2,  
        "Items": [  
          "/example-path/example-file2.png",  
          "/example-path/example-file.jpg"  
        ]  
      },  
      "CallerReference": "cli-1575570291-670203"  
    }  
  }  
}
```

Im vorherigen Beispiel wurde AWS CLI automatisch ein `CallerReference` Zufallscode generiert. Um eigene Parameter anzugeben oder um zu vermeiden `CallerReference`, dass die Invalidierungsparameter als Befehlszeilenargumente übergeben werden, können Sie eine JSON Datei verwenden. Im folgenden Beispiel wird eine Invalidierung für zwei Dateien erstellt,

indem die Invalidierungsparameter in einer JSON Datei mit dem Namen angegeben werden:
`inv-batch.json`

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --invalidation-batch file://inv-batch.json
```

Inhalt von `inv-batch.json`:

```
{  
  "Paths": {  
    "Quantity": 2,  
    "Items": [  
      "/example-path/example-file.jpg",  
      "/example-path/example-file2.png"  
    ]  
  },  
  "CallerReference": "cli-example"  
}
```

Ausgabe:

```
{  
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/  
EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUY0IK",  
  "Invalidation": {  
    "Id": "I2J0I21PCUY0IK",  
    "Status": "InProgress",  
    "CreateTime": "2019-12-05T18:40:49.413Z",  
    "InvalidationBatch": {  
      "Paths": {  
        "Quantity": 2,  
        "Items": [  
          "/example-path/example-file.jpg",  
          "/example-path/example-file2.png"  
        ]  
      },  
      "CallerReference": "cli-example"  
    }  
  }  
}
```

- API-Einheiten finden Sie unter [CreateInvalidation AWS CLI-Befehlsreferenz](#).

create-public-key

Das folgende Codebeispiel zeigt, wie man es benutzt `create-public-key`.

AWS CLI

Um einen CloudFront öffentlichen Schlüssel zu erstellen

Im folgenden Beispiel wird ein CloudFront öffentlicher Schlüssel erstellt, indem die Parameter in einer JSON-Datei mit dem Namen `pub-key-config.json` bereitgestellt werden. Bevor Sie diesen Befehl verwenden können, benötigen Sie einen mit PEM-codierten öffentlichen Schlüssel. Weitere Informationen finden Sie unter [Create an RSA Key Pair](#) im Amazon CloudFront Developer Guide.

```
aws cloudfront create-public-key \
  --public-key-config file://pub-key-config.json
```

Die Datei `pub-key-config.json` ist ein JSON-Dokument im aktuellen Ordner, das Folgendes enthält. Beachten Sie, dass der öffentliche Schlüssel im PEM-Format kodiert ist.

```
{
  "CallerReference": "cli-example",
  "Name": "ExampleKey",
  "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBcGKAQEAAxPmbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwXQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
  "Comment": "example public key"
}
```

Ausgabe:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
```

```

    "ETag": "E2QWRUHEXAMPLE",
    "PublicKey": {
      "Id": "KDFB19YGCR002",
      "CreatedTime": "2019-12-05T18:51:43.781Z",
      "PublicKeyConfig": {
        "CallerReference": "cli-example",
        "Name": "ExampleKey",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
        "Comment": "example public key"
      }
    }
  }
}

```

- API-Einheiten finden Sie [CreatePublicKey](#) in der AWS CLI Befehlsreferenz.

delete-cloud-front-origin-access-identity

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-cloud-front-origin-access-identity`.

AWS CLI

Um eine CloudFront ursprüngliche Zugriffsidentität zu löschen

Im folgenden Beispiel wird die ursprüngliche Zugriffsidentität (OAI) mit der ID `E74FTE3AEXAMPLE` gelöscht. Um eine zu löschen OAI, benötigen Sie die OAI s-ID und ETag. Die OAI ID wird in der Ausgabe der Befehle `create-cloud-front-origin-access-identity` und `list-cloud-front-origin-access-identities` zurückgegeben. Verwenden Sie den Befehl `-access-identity` oder `-ETag`, um das abzurufen. `get-cloud-front-origin-access-identity-config` Verwenden Sie die `--if-match` Option, um die s anzugeben. OAI ETag

```

aws cloudfront delete-cloud-front-origin-access-identity \
  --id E74FTE3AEXAMPLE \
  --if-match E2QWRUHEXAMPLE

```

Bei Erfolg hat dieser Befehl keine Ausgabe.

- API-Einheiten finden Sie [DeleteCloudFrontOriginAccessIdentity](#) in der AWS CLI Befehlsreferenz.

delete-distribution

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-distribution`.

AWS CLI

Um eine CloudFront Distribution zu löschen

Im folgenden Beispiel wird die CloudFront Distribution mit der ID `EDFDVBD6EXAMPLE` gelöscht. Bevor Sie eine Distribution löschen können, müssen Sie sie deaktivieren. Verwenden Sie den Befehl `update-distribution`, um eine Distribution zu deaktivieren. Weitere Informationen finden Sie in den Beispielen für die Update-Distribution.

Wenn eine Distribution deaktiviert ist, können Sie sie löschen. Um eine Distribution zu löschen, müssen Sie die `--if-match` Option zum Bereitstellen der Distribution verwenden `ETag`. Um die `ETag` abzurufen, verwenden Sie den `get-distribution-config` Befehl `get-distribution or`.

```
aws cloudfront delete-distribution \  
  --id EDFDVBD6EXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

Wenn dieser Befehl erfolgreich ist, hat er keine Ausgabe.

- API-Einheiten finden Sie [DeleteDistribution](#) in der AWS CLI Befehlsreferenz.

delete-field-level-encryption-config

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-field-level-encryption-config`.

AWS CLI

Um eine Verschlüsselungskonfiguration auf CloudFront Feldebene zu löschen

Im folgenden Beispiel wird die Verschlüsselungskonfiguration auf CloudFront Feldebene mit der ID `C3KM2WVD605UAY` gelöscht. Um eine Verschlüsselungskonfiguration auf Feldebene zu löschen, benötigen Sie die zugehörige ID und `ETag`. Die ID wird in der Ausgabe der Befehle

`create-field-level-encryption -config` und `list-field-level-encryption -configs` zurückgegeben. Verwenden Sie den Befehl `get-field-level-encryption` oder `ETag get-field-level-encryption -config`, um das abzurufen. Verwenden Sie die `--if-match` Option, um die Konfiguration bereitzustellen. ETag

```
aws cloudfront delete-field-level-encryption-config \  
  --id C3KM2WVD605UAY \  
  --if-match E26M4BIAV81ZF6
```

Bei Erfolg hat dieser Befehl keine Ausgabe.

- API-Einheiten finden Sie [DeleteFieldLevelEncryptionConfig](#) in der AWS CLI Befehlsreferenz.

delete-field-level-encryption-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-field-level-encryption-profile`.

AWS CLI

Um ein Verschlüsselungsprofil auf CloudFront Feldebene zu löschen

Im folgenden Beispiel wird das Verschlüsselungsprofil auf CloudFront Feldebene mit der ID `PPK0UOSIF5WSV` gelöscht. Um ein Verschlüsselungsprofil auf Feldebene zu löschen, benötigen Sie die zugehörige ID und ETag. Die ID wird in der Ausgabe der Befehle `create-field-level-encryption -profile` und `list-field-level-encryption -profiles` zurückgegeben. Verwenden Sie den Befehl `get-field-level-encryption -profile` oder `get-field-level-encryption -profile-config ETag`, um das abzurufen. Verwenden Sie die `--if-match` Option, um die Profile bereitzustellen. ETag

```
aws cloudfront delete-field-level-encryption-profile \  
  --id PPK0UOSIF5WSV \  
  --if-match EJETYFJ9CL66D
```

Bei Erfolg hat dieser Befehl keine Ausgabe.

- API-Einheiten finden Sie [DeleteFieldLevelEncryptionProfile](#) in der AWS CLI Befehlsreferenz.

delete-public-key

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-public-key`.

AWS CLI

Um einen CloudFront öffentlichen Schlüssel zu löschen

Im folgenden Beispiel wird der CloudFront öffentliche Schlüssel mit der ID `KDFB19YGCR002` gelöscht. Um einen öffentlichen Schlüssel zu löschen, benötigen Sie seine ID und ETag. Die ID wird in der Ausgabe der `list-public-keys` Befehle `create-public-key` und zurückgegeben. Verwenden Sie den `get-public-key-config` Befehl `get-public-key` oder ETag, um das abzurufen. Verwenden Sie die `--if-match` Option, um die öffentlichen Schlüssel bereitzustellen ETag.

```
aws cloudfront delete-public-key \  
  --id KDFB19YGCR002 \  
  --if-match E2QWRUHEXAMPLE
```

Bei Erfolg hat dieser Befehl keine Ausgabe.

- API Einzelheiten finden Sie [DeletePublicKey](#) in der AWS CLI Befehlsreferenz.

get-cloud-front-origin-access-identity-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-cloud-front-origin-access-identity-config`.

AWS CLI

Um eine Konfiguration der CloudFront Origin-Zugriffsidentität zu erhalten

Im folgenden Beispiel werden Metadaten zur ursprünglichen CloudFront Zugriffsidentität (OAI) mit der ID `E74FTE3AEXAMPLE` abgerufen, einschließlich ihrer ETag. Die OAI ID wird in der Ausgabe der Befehle `create-cloud-front-origin -access-identity` und `list-cloud-front-origin -access-identities` zurückgegeben.

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

Ausgabe:

```
{  
  "ETag": "E2QWRUHEXAMPLE",  
  "CloudFrontOriginAccessIdentityConfig": {  
    "CallerReference": "cli-example",  
    "Comment": "Example OAI"  
  }  
}
```

```
}

```

- Einzelheiten API finden Sie in der Befehlsreferenz. [GetCloudFrontOriginAccessIdentityConfig](#) AWS CLI

get-cloud-front-origin-access-identity

Das folgende Codebeispiel zeigt, wie man es benutzt `get-cloud-front-origin-access-identity`.

AWS CLI

Um eine CloudFront ursprüngliche Zugriffsidentität zu erhalten

Im folgenden Beispiel wird die ursprüngliche CloudFront Zugriffsidentität (OAI) mit der ID `E74FTE3AEXAMPLE`, einschließlich ihrer ETag und der zugehörigen kanonischen S3-ID. Die OAI ID wird in der Ausgabe der Befehle `-access-identity` und `create-cloud-front-origin-access-identities` zurückgegeben. `list-cloud-front-origin`

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

Ausgabe:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI"
    }
  }
}
```

- Einzelheiten API finden Sie in der Befehlsreferenz. [GetCloudFrontOriginAccessIdentity](#) AWS CLI

get-distribution-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-distribution-config`.

AWS CLI

Um eine CloudFront Distributionskonfiguration zu erhalten

Im folgenden Beispiel werden Metadaten über die CloudFront Distribution mit der ID abgerufen `EDFDVBD6EXAMPLE`, einschließlich ihrer ETag. Die Distributions-ID wird in den Befehlen `create-distribution` und `list-distributions` zurückgegeben.

```
aws cloudfront get-distribution-config --id EDFDVBD6EXAMPLE
```

Ausgabe:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
          "DomainName": "awsexamplebucket.s3.amazonaws.com",
          "OriginPath": "",
          "CustomHeaders": {
            "Quantity": 0
          },
          "S3OriginConfig": {
            "OriginAccessIdentity": ""
          }
        }
      ]
    },
    "OriginGroups": {
      "Quantity": 0
    },
    "DefaultCacheBehavior": {
      "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
      "ForwardedValues": {
        "QueryString": false,

```

```
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
}
```

```

    },
    "Comment": "",
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}

```

- API-Einheiten finden Sie in der Befehlsreferenz. [GetDistributionConfig](#) AWS CLI

get-distribution

Das folgende Codebeispiel zeigt, wie man es benutzt `get-distribution`.

AWS CLI

Um eine CloudFront Distribution zu erhalten

Im folgenden Beispiel wird die CloudFront Distribution mit der ID `EDFDVBD6EXAMPLE`, einschließlich ihrer ETag, abgerufen. Die Distributions-ID wird in den Befehlen `create-distribution` und `list-distributions` zurückgegeben.

```
aws cloudfront get-distribution --id EDFDVBD6EXAMPLE
```

Ausgabe:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "Deployed",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
  },
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
      "QueryString": false,

```

```
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
```



```

    },
    "Comment": "",
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}
}
}

```

- API-Einheiten finden Sie in der Befehlsreferenz. [GetDistribution](#) AWS CLI

get-field-level-encryption-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-field-level-encryption-config`.

AWS CLI

Um Metadaten zu einer Verschlüsselungskonfiguration auf CloudFront Feldebene abzurufen

Im folgenden Beispiel werden Metadaten über die Verschlüsselungskonfiguration auf CloudFront Feldebene mit der ID `C3KM2WVD605UAY` abgerufen, einschließlich ihrer ETag

```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

Ausgabe:

```
{
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryptionConfig": {
    "CallerReference": "cli-example",
    "Comment": "Example FLE configuration",
    "QueryArgProfileConfig": {
      "ForwardWhenQueryArgProfileIsUnknown": true,
      "QueryArgProfiles": {
        "Quantity": 0,
        "Items": []
      }
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSY0CVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}
```

- API-Einheiten finden Sie unter [GetFieldLevelEncryptionConfig AWS CLI Befehlsreferenz](#).

get-field-level-encryption-profile-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-field-level-encryption-profile-config`.

AWS CLI

Um eine Konfiguration für ein Verschlüsselungsprofil CloudFront auf Feldebene zu erhalten

Im folgenden Beispiel werden Metadaten über das Verschlüsselungsprofil auf CloudFront Feldebene mit ID PPK0U0SIF5WSV abgerufen, einschließlich seiner: ETag

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0U0SIF5WSV
```

Ausgabe:

```
{
  "ETag": "E1QQG65FS2L2GC",
  "FieldLevelEncryptionProfileConfig": {
    "Name": "ExampleFLEProfile",
    "CallerReference": "cli-example",
    "Comment": "FLE profile for AWS CLI example",
    "EncryptionEntities": {
      "Quantity": 1,
      "Items": [
        {
          "PublicKeyId": "K2K8NC4HVFE3M0",
          "ProviderId": "ExampleFLEProvider",
          "FieldPatterns": {
            "Quantity": 1,
            "Items": [
              "ExampleSensitiveField"
            ]
          }
        }
      ]
    }
  }
}
```

- API-Einheiten finden Sie unter [GetFieldLevelEncryptionProfileConfig AWS CLI-Befehlsreferenz](#).

get-field-level-encryption-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `get-field-level-encryption-profile`.

AWS CLI

Um ein Verschlüsselungsprofil auf CloudFront Feldebene zu erhalten

Im folgenden Beispiel wird das Verschlüsselungsprofil auf CloudFront Feldebene mit der ID `PPK0U0SIF5WSV` abgerufen, einschließlich seiner: ETag

```
aws cloudfront get-field-level-encryption-profile --id PPK0U0SIF5WSV
```

Ausgabe:

```
{
  "ETag": "E1QQG65FS2L2GC",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0U0SIF5WSV",
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfile",
      "CallerReference": "cli-example",
      "Comment": "FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {
            "PublicKeyId": "K2K8NC4HVFE3M0",
            "ProviderId": "ExampleFLEProvider",
            "FieldPatterns": {
              "Quantity": 1,
              "Items": [
                "ExampleSensitiveField"
              ]
            }
          }
        ]
      }
    }
  }
}
```

- API-Einheiten finden Sie [GetFieldLevelEncryptionProfile](#) in der AWS CLI Befehlsreferenz.

get-field-level-encryption

Das folgende Codebeispiel zeigt, wie man es benutzt `get-field-level-encryption`.

AWS CLI

Um eine Verschlüsselungskonfiguration auf CloudFront Feldebene zu erhalten

Im folgenden Beispiel wird die Verschlüsselungskonfiguration auf CloudFront Feldebene mit der ID C3KM2WVD605UAY abgerufen, einschließlich ihrer: ETag

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

Ausgabe:

```
{
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryption": {
    "Id": "C3KM2WVD605UAY",
    "LastModifiedTime": "2019-12-10T21:30:18.974Z",
    "FieldLevelEncryptionConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example FLE configuration",
      "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
          "Quantity": 0,
          "Items": []
        }
      },
      "ContentTypeProfileConfig": {
        "ForwardWhenContentTypeIsUnknown": true,
        "ContentTypeProfiles": {
          "Quantity": 1,
          "Items": [
            {
              "Format": "URLEncoded",
              "ProfileId": "P280MFCLSYOCVU",
              "ContentType": "application/x-www-form-urlencoded"
            }
          ]
        }
      }
    }
  }
}
```

- API-Einheiten finden Sie [GetFieldLevelEncryption](#) in der AWS CLI Befehlsreferenz.

get-invalidation

Das folgende Codebeispiel zeigt, wie man es benutzt `get-invalidation`.

AWS CLI

Um eine CloudFront Ungültigerklärung zu erhalten

Im folgenden Beispiel wird die Invalidierung mit der ID `I2J0I21PCUY0IK` für die CloudFront Distribution mit der ID abgerufen: `EDFDVBD6EXAMPLE`

```
aws cloudfront get-invalidation --id I2J0I21PCUY0IK --distribution-id EDFDVBD6EXAMPLE
```

Ausgabe:

```
{
  "Invalidation": {
    "Status": "Completed",
    "InvalidationBatch": {
      "Paths": {
        "Items": [
          "/example-path/example-file.jpg",
          "/example-path/example-file-2.jpg"
        ],
        "Quantity": 2
      },
      "CallerReference": "cli-example"
    },
    "Id": "I2J0I21PCUY0IK",
    "CreateTime": "2019-12-05T18:40:49.413Z"
  }
}
```

- API-Einheiten finden Sie [GetInvalidation](#) in der AWS CLI Befehlsreferenz.

get-public-key-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-public-key-config`.

AWS CLI

Um eine Konfiguration mit CloudFront öffentlichen Schlüsseln zu erhalten

Im folgenden Beispiel werden Metadaten über den CloudFront öffentlichen Schlüssel mit der ID abgerufen `KDFB19YGCR002`, einschließlich seiner ETag. Die ID des öffentlichen Schlüssels wird in den `list-public-keys` Befehlen `create-public-key` und zurückgegeben.

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

Ausgabe:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKeyConfig": {
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAPmbCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLumore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
    "Comment": "example public key"
  }
}
```

- API Einzelheiten finden Sie [GetPublicKeyConfig](#) in der AWS CLI Befehlsreferenz.

get-public-key

Das folgende Codebeispiel zeigt, wie man es benutzt `get-public-key`.

AWS CLI

Um einen CloudFront öffentlichen Schlüssel zu erhalten

Das folgende Beispiel ruft den CloudFront öffentlichen Schlüssel mit der ID ab `KDFB19YGCR002`, einschließlich seiner ETag. Die ID des öffentlichen Schlüssels wird in den `list-public-keys` Befehlen `create-public-key` und zurückgegeben.

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

Ausgabe:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKey": {
    "Id": "KDFB19YGCR002",
    "CreatedTime": "2019-12-05T18:51:43.781Z",
    "PublicKeyConfig": {
      "CallerReference": "cli-example",
      "Name": "ExampleKey",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPmBCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUMore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnStb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
      "Comment": "example public key"
    }
  }
}
```

- API-Einheiten finden Sie [GetPublicKey](#) in der AWS CLI Befehlsreferenz.

list-cloud-front-origin-access-identities

Das folgende Codebeispiel zeigt, wie man es benutzt `list-cloud-front-origin-access-identities`.

AWS CLI

Um die ursprünglichen CloudFront Zugriffsidentitäten aufzulisten

Im folgenden Beispiel wird eine Liste der ursprünglichen CloudFront Zugriffsidentitäten (OAs) in Ihrem AWS Konto abgerufen:

```
aws cloudfront list-cloud-front-origin-access-identities
```

Ausgabe:

```
{
```



```

    "CloudFrontOriginAccessIdentityList": {
      "Items": [
        {
          "Id": "E74FTE3AEXAMPLE",
          "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
          "Comment": "Example OAI"
        },
        {
          "Id": "EH1HDMBEXAMPLE",
          "S3CanonicalUserId":
"1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
          "Comment": "Test OAI"
        },
        {
          "Id": "E2X2C9TEXAMPLE",
          "S3CanonicalUserId":
"cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
          "Comment": "Example OAI #2"
        }
      ]
    }
  }
}

```

- API-Einheiten finden Sie [ListCloudFrontOriginAccessIdentities](#) in der AWS CLI Befehlsreferenz.

list-distributions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-distributions`.

AWS CLI

Um CloudFront Distributionen aufzulisten

Im folgenden Beispiel wird eine Liste der CloudFront Verteilungen in Ihrem AWS Konto abgerufen:

```
aws cloudfront list-distributions
```

Ausgabe:

```
{
```

```

"DistributionList": {
  "Items": [
    {
      "Id": "EMLARXS9EXAMPLE",
      "ARN": "arn:aws:cloudfront::123456789012:distribution/
EMLARXS9EXAMPLE",
      "Status": "InProgress",
      "LastModifiedTime": "2019-11-22T00:55:15.705Z",
      "InProgressInvalidationBatches": 0,
      "DomainName": "d111111abcdef8.cloudfront.net",
      "ActiveTrustedSigners": {
        "Enabled": false,
        "Quantity": 0
      },
      "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
          "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
          "Quantity": 1,
          "Items": [
            {
              "Id": "awsexamplebucket.s3.amazonaws.com-cli-
example",
              "DomainName": "awsexamplebucket.s3.amazonaws.com",
              "OriginPath": "",
              "CustomHeaders": {
                "Quantity": 0
              },
              "S3OriginConfig": {
                "OriginAccessIdentity": ""
              }
            }
          ]
        },
        "OriginGroups": {
          "Quantity": 0
        },
        "DefaultCacheBehavior": {
          "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-
example",
          "ForwardedValues": {

```

```
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
```

```

        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
},
{
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d930174dauwrn8.cloudfront.net",
    "ActiveTrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        }
    }
},

```

```

"DefaultRootObject": "index.html",
"Origins": {
  "Quantity": 1,
  "Items": [
    {
      "Id": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
      "DomainName": "awsexamplebucket1.s3.amazonaws.com",
      "OriginPath": "",
      "CustomHeaders": {
        "Quantity": 0
      },
      "S3OriginConfig": {
        "OriginAccessIdentity": ""
      }
    }
  ]
},
"OriginGroups": {
  "Quantity": 0
},
"DefaultCacheBehavior": {
  "TargetOriginId": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
  "ForwardedValues": {
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,

```

```
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
},
"Restrictions": {
    "GeoRestriction": {
        "RestrictionType": "none",
```

```

        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
},
{
  "Id": "E1X5IZQEXAMPLE",
  "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1X5IZQEXAMPLE",
  "Status": "Deployed",
  "LastModifiedTime": "2019-11-06T21:31:48.864Z",
  "DomainName": "d2e04y12345678.cloudfront.net",
  "Aliases": {
    "Quantity": 0
  },
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket2",
        "DomainName": "awsexamplebucket2.s3.us-
west-2.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket2",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      }
    }
  }
}

```

```
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
  },
  "CachedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ]
  }
},
"SmoothStreaming": false,
"DefaultTTL": 86400,
"MaxTTL": 31536000,
"Compress": false,
"LambdaFunctionAssociations": {
  "Quantity": 0
},
"FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponse": {
  "Quantity": 0
},
"Comment": "",
```



```

    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "HTTP1_1",
    "IsIPV6Enabled": true
  }
]
}

```

- API-Einheiten finden Sie [ListDistributions](#) in der AWS CLI Befehlsreferenz.

list-field-level-encryption-configs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-field-level-encryption-configs`.

AWS CLI

Um Verschlüsselungskonfigurationen auf CloudFront Feldebene aufzulisten

Im folgenden Beispiel wird eine Liste der Verschlüsselungskonfigurationen auf CloudFront Feldebene in Ihrem Konto abgerufen: AWS

```
aws cloudfront list-field-level-encryption-configs
```

Ausgabe:

```

{
  "FieldLevelEncryptionList": {
    "MaxItems": 100,
    "Quantity": 1,

```

```

    "Items": [
      {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "Comment": "Example FLE configuration",
        "QueryArgProfileConfig": {
          "ForwardWhenQueryArgProfileIsUnknown": true,
          "QueryArgProfiles": {
            "Quantity": 0,
            "Items": []
          }
        },
        "ContentTypeProfileConfig": {
          "ForwardWhenContentTypeIsUnknown": true,
          "ContentTypeProfiles": {
            "Quantity": 1,
            "Items": [
              {
                "Format": "URLEncoded",
                "ProfileId": "P280MFCLSY0CVU",
                "ContentType": "application/x-www-form-urlencoded"
              }
            ]
          }
        }
      }
    ]
  }
}

```

- API-Einheiten finden Sie [ListFieldLevelEncryptionConfigs](#) in der AWS CLI Befehlsreferenz.

list-field-level-encryption-profiles

Das folgende Codebeispiel zeigt, wie man es benutzt `list-field-level-encryption-profiles`.

AWS CLI

Um Verschlüsselungsprofile auf CloudFront Feldebene aufzulisten

Im folgenden Beispiel wird eine Liste der Verschlüsselungsprofile auf CloudFront Feldebene in Ihrem Konto abgerufen: AWS

aws cloudfront list-field-level-encryption-profiles

Ausgabe:

```
{
  "FieldLevelEncryptionProfileList": {
    "MaxItems": 100,
    "Quantity": 2,
    "Items": [
      {
        "Id": "P280MFCLSY0CVU",
        "LastModifiedTime": "2019-12-05T01:05:39.896Z",
        "Name": "ExampleFLEProfile",
        "EncryptionEntities": {
          "Quantity": 1,
          "Items": [
            {
              "PublicKeyId": "K2K8NC4HVFE3M0",
              "ProviderId": "ExampleFLEProvider",
              "FieldPatterns": {
                "Quantity": 1,
                "Items": [
                  "ExampleSensitiveField"
                ]
              }
            }
          ]
        },
        "Comment": "FLE profile for AWS CLI example"
      },
      {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T01:03:16.537Z",
        "Name": "ExampleFLEProfile2",
        "EncryptionEntities": {
          "Quantity": 1,
          "Items": [
            {
              "PublicKeyId": "K2ABC10EXAMPLE",
              "ProviderId": "ExampleFLEProvider2",
              "FieldPatterns": {
                "Quantity": 1,
                "Items": [

```

```

    "ExampleSensitiveField2"
  ]
}
},
"Comment": "FLE profile #2 for AWS CLI example"
}
]
}
}

```

- API-Einheiten finden Sie [ListFieldLevelEncryptionProfiles](#) in der AWS CLI Befehlsreferenz.

list-invalidations

Das folgende Codebeispiel zeigt, wie man es benutzt `list-invalidations`.

AWS CLI

Um CloudFront Ungültigkeiten aufzulisten

Im folgenden Beispiel wird eine Liste der Ungültigerklärungen für die CloudFront Distribution mit der ID abgerufen: `EDFDVBD6EXAMPLE`

```
aws cloudfront list-invalidations --distribution-id EDFDVBD6EXAMPLE
```

Ausgabe:

```

{
  "InvalidationList": {
    "Marker": "",
    "Items": [
      {
        "Status": "Completed",
        "Id": "YNY2LI2BVJ4NJU",
        "CreateTime": "2019-08-31T21:15:52.042Z"
      }
    ],
    "IsTruncated": false,
    "MaxItems": 100,
    "Quantity": 1
  }
}

```

```
}
}
```

- API-Einheiten finden Sie unter [ListInvalidations AWS CLI-Befehlsreferenz](#).

list-public-keys

Das folgende Codebeispiel zeigt, wie man es benutzt `list-public-keys`.

AWS CLI

Um CloudFront öffentliche Schlüssel aufzulisten

Im folgenden Beispiel wird eine Liste der CloudFront öffentlichen Schlüssel in Ihrem AWS Konto abgerufen:

```
aws cloudfront list-public-keys
```

Ausgabe:

```
{
  "PublicKeyList": {
    "MaxItems": 100,
    "Quantity": 2,
    "Items": [
      {
        "Id": "K2K8NC4HVFE3M0",
        "Name": "ExampleKey",
        "CreatedTime": "2019-12-05T01:04:28.818Z",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPmBCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAE/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBaz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWNwe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
        "Comment": "example public key"
      },
      {
        "Id": "K1S0LWQ2L5HTBU",
        "Name": "ExampleKey2",
```

```

        "CreatedTime": "2019-12-09T23:28:11.110Z",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAAOCAQ8AMIIBCgKCAQEAp0CAg88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1go17FV6pYN178iJg3E08JBbwt1H
+cR9\nLGSf60NDeVhm760c39Np/vWg0dsGQcRbi9WmKZeS0DqjQGzVZWqPmito3FzWV6b
\nfVY5N36U/RdbVAJm95Km+qaMY1bIdF40t72bi3IkKYV5h1B2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmmgIVnk0QM37NYeoDnka0uTpu\nha/
+3b8t0b2z3LBVHPkp85zJRA0XacSwf5rZtPYKBNFsixTa2n55k2r218m0kMC4\nUwIDAQAB\n-----END
PUBLIC KEY-----",
        "Comment": "example public key #2"
    }
  ]
}

```

- API-Einheiten finden Sie [ListPublicKeys](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um Tags für eine CloudFront Distribution aufzulisten

Im folgenden Beispiel wird eine Liste der Tags für eine CloudFront Distribution abgerufen:

```

aws cloudfront list-tags-for-resource \
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE

```

Ausgabe:

```

{
  "Tags": {
    "Items": [
      {
        "Key": "DateCreated",
        "Value": "2019-12-04"
      },
      {
        "Key": "Name",
        "Value": "Example name"
      }
    ]
  }
}

```

```

    },
    {
      "Key": "Project",
      "Value": "Example project"
    }
  ]
}

```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

sign

Das folgende Codebeispiel zeigt, wie man es benutzt.

AWS CLI

Um ein zu signieren CloudFront URL

Das folgende Beispiel signiert CloudFront URL a. Um eine zu signieren URL, benötigen Sie die Schlüsselpaar-ID (in der AWS Management Console als Access Key-ID bezeichnet) und den privaten Schlüssel des CloudFront key pair des vertrauenswürdigen Unterzeichners. Weitere Informationen zu signierten URLs finden Sie unter [Bereitstellung privater Inhalte mit signierten URLs und signierten Cookies](#) im Amazon CloudFront Developer Guide.

```

aws cloudfront sign \
  --url https://d111111abcdef8.cloudfront.net/private-content/private-file.html \
  --key-pair-id APKAEIBAERJR2EXAMPLE \
  --private-key file://cf-signer-priv-key.pem \
  --date-less-than 2020-01-01

```

Ausgabe:

```

https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKvc6pwkif6oZc-
JWSpDkH0UH7EBGGqvgurkeCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoenH0GFteRf9ELK-
Bs3nljTLjtbgzIUt7QJNKXcWr8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GGl6l2kgCpXUB6Jx6K~Y3kpV0dzUP0IqFLHAnJobjbhxqrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FK0qNEaWDIB
GNvjRJxqWf93uMobeM0iVYahb-e0KIItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE

```

- API-Einheiten finden Sie unter AWS CLI Befehlsreferenz zum [Anmelden](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine CloudFront Distribution zu taggen

Im folgenden `tag-resource` Beispiel werden der angegebenen CloudFront Verteilung zwei Tags hinzugefügt.

```
aws cloudfront tag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example project"}]'
```

Anstatt Befehlszeilenargumente zu verwenden, können Sie die Tags in einer JSON Datei angeben, wie im folgenden Beispiel gezeigt:

```
aws cloudfront tag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tags file://tags.json
```

Inhalt von `tags.json`:

```
{  
  "Items": [  
    {  
      "Key": "Name",  
      "Value": "Example name"  
    },  
    {  
      "Key": "Project",  
      "Value": "Example project"  
    }  
  ]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um Tags aus einer CloudFront Distribution zu entfernen

Im folgenden Beispiel werden mithilfe von Befehlszeilenargumenten zwei Tags aus einer CloudFront Distribution entfernt:

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys Items=Name,Project
```

Anstatt Befehlszeilenargumente zu verwenden, können Sie die Tag-Schlüssel in einer JSON Datei angeben, wie im folgenden Beispiel gezeigt:

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys file://tag-keys.json
```

Die Datei `tag-keys.json` ist ein JSON Dokument im aktuellen Ordner, das Folgendes enthält:

```
{  
  "Items": [  
    "Name",  
    "Project"  
  ]  
}
```

Bei erfolgreicher Ausführung hat dieser Befehl keine Ausgabe.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-cloud-front-origin-access-identity

Das folgende Codebeispiel zeigt, wie man es benutzt `update-cloud-front-origin-access-identity`.

AWS CLI

Um eine CloudFront ursprüngliche Zugriffsidentität zu aktualisieren

Im folgenden Beispiel wird die ursprüngliche Zugriffsidentität (OAI) mit der ID aktualisiert `E74FTE3AEXAMPLE`. Das einzige Feld, das Sie aktualisieren können, OAI ist das Feld `Comment`.

Um eine zu aktualisieren OAI, benötigen Sie die OAI s-ID und ETag. Die OAI ID wird in der Ausgabe der Befehle `create-cloud-front-origin -access-identity` und `list-cloud-front-origin -access-identities` zurückgegeben. Verwenden Sie den Befehl `-access-identity` oder `-ETag`, um das abzurufen. `get-cloud-front-origin get-cloud-front-origin access-identity-config` Verwenden Sie die `--if-match` Option, um die s anzugeben. OAI ETag

```
aws cloudfront update-cloud-front-origin-access-identity \
  --id E74FTE3AEXAMPLE \
  --if-match E2QWRUHEXAMPLE \
  --cloud-front-origin-access-identity-config \
    CallerReference=cli-example,Comment="Example OAI Updated"
```

Sie können dasselbe erreichen, indem Sie die OAI Konfiguration in einer JSON Datei angeben, wie im folgenden Beispiel gezeigt:

```
aws cloudfront update-cloud-front-origin-access-identity \
  --id E74FTE3AEXAMPLE \
  --if-match E2QWRUHEXAMPLE \
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

Die Datei `OAI-config.json` ist ein JSON Dokument im aktuellen Verzeichnis, das Folgendes enthält:

```
{
  "CallerReference": "cli-example",
  "Comment": "Example OAI Updated"
}
```

Unabhängig davon, ob Sie die OAI Konfiguration mit einem Befehlszeilenargument oder einer JSON Datei angeben, ist die Ausgabe dieselbe:

```
{
  "ETag": "E9LHASXEXAMPLE",
```

```

    "CloudFrontOriginAccessIdentity": {
      "Id": "E74FTE3AEXAMPLE",
      "S3CanonicalUserId":
"cd13868f797c227f7bea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
      "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI Updated"
      }
    }
  }
}

```

- API-Einheiten finden Sie [UpdateCloudFrontOriginAccessIdentity](#) unter AWS CLI Befehlsreferenz.

update-distribution

Das folgende Codebeispiel zeigt, wie man es benutzt `update-distribution`.

AWS CLI

Um das Standard-Root-Objekt einer CloudFront Distribution zu aktualisieren

Im folgenden Beispiel wird das Standard-Stammobjekt `index.html` für die CloudFront Distribution mit der ID `EDFDVBD6EXAMPLE` aktualisiert:

```

aws cloudfront update-distribution --id EDFDVBD6EXAMPLE \
  --default-root-object index.html

```

Ausgabe:

```

{
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:55:39.870Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  }
}

```

```
  },
  "DistributionConfig": {
    "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
    "Aliases": {
      "Quantity": 0
    },
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "example-website",
        "DomainName": "www.example.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
      },
      "CustomOriginConfig": {
        "HTTPPort": 80,
        "HTTPSPort": 443,
        "OriginProtocolPolicy": "match-viewer",
        "OriginSslProtocols": {
          "Quantity": 2,
          "Items": [
            "SSLv3",
            "TLSv1"
          ]
        },
        "OriginReadTimeout": 30,
        "OriginKeepaliveTimeout": 5
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "example-website",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      }
    },
  },
}
```

```
    "Headers": {
      "Quantity": 1,
      "Items": [
        "*"
      ]
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ]
  },
  "CachedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ]
  }
},
"SmoothStreaming": false,
"DefaultTTL": 86400,
"MaxTTL": 31536000,
"Compress": false,
"LambdaFunctionAssociations": {
  "Quantity": 0
},
"FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
```

```

    },
    "Comment": "",
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http1.1",
    "IsIPV6Enabled": true
  }
}
}
}

```

Um eine CloudFront Distribution zu aktualisieren

Im folgenden Beispiel wird die CloudFront Distribution mit der ID deaktiviert, EMLARXS9EXAMPLE indem die Verteilungskonfiguration in einer JSON Datei mit dem Namen `dist-config-disable.json` bereitgestellt wird. Um eine Distribution zu aktualisieren, müssen Sie die `--if-match` Option zur Bereitstellung der Distribution verwenden. ETag Um die abzurufenETag, verwenden Sie den `get-distribution-config` Befehl `get-distribution` or.

Nachdem Sie das folgende Beispiel verwendet haben, um eine Distribution zu deaktivieren, können Sie sie mit dem Befehl `delete-distribution` löschen.

```

aws cloudfront update-distribution \
  --id EMLARXS9EXAMPLE \
  --if-match E2QWRUHEXAMPLE \

```

```
--distribution-config file://dist-config-disable.json
```

Die Datei `dist-config-disable.json` ist ein JSON Dokument im aktuellen Ordner, das Folgendes enthält. Beachten Sie, dass das `Enabled` Feld wie folgt gesetzt ist `false`:

```
{
  "CallerReference": "cli-1574382155-496510",
  "Aliases": {
    "Quantity": 0
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      },
      "Headers": {
        "Quantity": 0
      },
      "QueryStringCacheKeys": {
        "Quantity": 0
      }
    }
  }
}
```

```
    },
    "TrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ],
      "CachedMethods": {
        "Quantity": 2,
        "Items": [
          "HEAD",
          "GET"
        ]
      }
    },
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
      "Quantity": 0
    },
    },
    "FieldLevelEncryptionId": ""
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
```



```

    "Enabled": false,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}

```

Ausgabe:

```

{
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:32:35.553Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
  "DistributionConfig": {
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",

```

```
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    ]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId":
"awsexamplebucket.s3.amazonaws.com-1574382155-273939",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
        "Headers": {
            "Quantity": 0
        },
        "QueryStringCacheKeys": {
            "Quantity": 0
        }
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
```

```
        "HEAD",
        "GET"
    ]
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": false,
"ViewerCertificate": {
  "CloudFrontDefaultCertificate": true,
  "MinimumProtocolVersion": "TLSv1",
  "CertificateSource": "cloudfront"
},
"Restrictions": {
  "GeoRestriction": {
    "RestrictionType": "none",
    "Quantity": 0
  }
},
"WebACLId": "",
"HttpVersion": "http2",
"IsIPV6Enabled": true
}
```

```
}
}
```

- API-Einheiten finden Sie [UpdateDistribution](#) in der AWS CLI Befehlsreferenz.

update-field-level-encryption-config

Das folgende Codebeispiel zeigt, wie man es benutzt `update-field-level-encryption-config`.

AWS CLI

Um eine Verschlüsselungskonfiguration auf CloudFront Feldebene zu aktualisieren

Im folgenden Beispiel wird das `Comment` Feld der Konfiguration für die Verschlüsselung auf Feldebene mit der ID aktualisiert, `C3KM2WVD605UAY` indem die Parameter in einer Datei bereitgestellt werden. JSON

Um eine Verschlüsselungskonfiguration auf Feldebene zu aktualisieren, benötigen Sie die ID und der Konfiguration. ETag Die ID wird in der Ausgabe der Befehle `create-field-level-encryption-config` und `list-field-level-encryption-configs` zurückgegeben. Verwenden Sie den Befehl `get-field-level-encryption` oder ETag `get-field-level-encryption-config`, um das abzurufen. Verwenden Sie die `--if-match` Option, um die Konfiguration bereitzustellen. ETag

```
aws cloudfront update-field-level-encryption-config \
  --id C3KM2WVD605UAY \
  --if-match E2P4Z4VU7TY5SG \
  --field-level-encryption-config file://fle-config.json
```

Die Datei `fle-config.json` ist ein JSON Dokument im aktuellen Verzeichnis, das Folgendes enthält:

```
{
  "CallerReference": "cli-example",
  "Comment": "Updated example FLE configuration",
  "QueryArgProfileConfig": {
    "ForwardWhenQueryArgProfileIsUnknown": true,
    "QueryArgProfiles": {
      "Quantity": 0
    }
  }
}
```

```

},
"ContentTypeProfileConfig": {
  "ForwardWhenContentTypeIsUnknown": true,
  "ContentTypeProfiles": {
    "Quantity": 1,
    "Items": [
      {
        "Format": "URLEncoded",
        "ProfileId": "P280MFCLSYOCVU",
        "ContentType": "application/x-www-form-urlencoded"
      }
    ]
  }
}
}

```

Ausgabe:

```

{
  "ETag": "E26M4BIAV81ZF6",
  "FieldLevelEncryption": {
    "Id": "C3KM2WVD605UAY",
    "LastModifiedTime": "2019-12-10T22:26:26.170Z",
    "FieldLevelEncryptionConfig": {
      "CallerReference": "cli-example",
      "Comment": "Updated example FLE configuration",
      "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
          "Quantity": 0,
          "Items": []
        }
      }
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSYOCVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}

```



```

"Comment": "Updated FLE profile for AWS CLI example",
"EncryptionEntities": {
  "Quantity": 1,
  "Items": [
    {
      "PublicKeyId": "K2K8NC4HVFE3M0",
      "ProviderId": "ExampleFLEProvider",
      "FieldPatterns": {
        "Quantity": 2,
        "Items": [
          "ExampleSensitiveField",
          "SecondExampleSensitiveField"
        ]
      }
    }
  ]
}
}
}
}

```

Ausgabe:

```

{
  "ETag": "EJETYFJ9CL66D",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0UOSIF5WSV",
    "LastModifiedTime": "2019-12-10T19:05:58.296Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfileUpdated",
      "CallerReference": "cli-example",
      "Comment": "Updated FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {
            "PublicKeyId": "K2K8NC4HVFE3M0",
            "ProviderId": "ExampleFLEProvider",
            "FieldPatterns": {
              "Quantity": 2,
              "Items": [
                "ExampleSensitiveField",
                "SecondExampleSensitiveField"
              ]
            }
          }
        ]
      }
    }
  }
}

```

```
}  
  }  
    }  
      ]  
        }  
          }  
            }
```

- API-Einheiten finden Sie [UpdateFieldLevelEncryptionProfile](#) in der AWS CLI Befehlsreferenz.

CloudSearch Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren CloudSearch.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

upload-documents

Das folgende Codebeispiel zeigt die Verwendung `upload-documents`.

AWS CLI

Der folgende `upload-documents` Befehl lädt einen Stapel von JSON Dokumenten in eine CloudSearch Amazon-Domain hoch:

```
aws cloudsearchdomain upload-documents --endpoint-url https://doc-my-  
domain.us-west-1.cloudsearch.amazonaws.com --content-type application/json --  
documents document-batch.json
```

Ausgabe:


```
{
  "status": "success",
  "adds": 5000,
  "deletes": 0
}
```

- API-Einheiten finden Sie [UploadDocuments](#) in der AWS CLI Befehlsreferenz.

CloudTrail Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CloudTrail.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags

Das folgende Codebeispiel zeigt die Verwendung `add-tags`.

AWS CLI

Um dem Trail Tags hinzuzufügen

Der folgende `add-tags` Befehl fügt Tags hinzu für `Trail1`:

```
aws cloudtrail add-tags --resource-id arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --tags-list Key=name,Value=Alice Key=location,Value=us
```

- API-Einheiten finden Sie [AddTags](#) in der AWS CLI Befehlsreferenz.

create-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-subscription`.

AWS CLI

Um AWS Ressourcen für einen Trail zu erstellen und zu konfigurieren

Der folgende `create-subscription` Befehl erstellt einen neuen S3-Bucket und ein neues SNS Thema für `Trail1`:

```
aws cloudtrail create-subscription --name Trail1 --s3-new-bucket my-bucket --sns-new-topic my-topic
```

Ausgabe:

```
Setting up new S3 bucket my-bucket...
Setting up new SNS topic my-topic...
Creating/updating CloudTrail configuration...
CloudTrail configuration:
{
  "trailList": [
    {
      "IncludeGlobalServiceEvents": true,
      "Name": "Trail1",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
      "LogFileValidationEnabled": false,
      "IsMultiRegionTrail": false,
      "S3BucketName": "my-bucket",
      "SnsTopicName": "my-topic",
      "HomeRegion": "us-east-1"
    }
  ],
  "ResponseMetadata": {
    "HTTPStatusCode": 200,
    "RequestId": "f39e51f6-c615-11e5-85bd-d35ca21ee3e2"
  }
}
Starting CloudTrail service...
Logs will be delivered to my-bucket
```

- API Einheiten finden Sie [CreateSubscription](#) in der AWS CLI Befehlsreferenz.

create-trail

Das folgende Codebeispiel zeigt die Verwendung `create-trail`.

AWS CLI

Um einen Trail zu erstellen

Der folgende `create-trail` Befehl erstellt einen Trail mit mehreren Regionen mit dem Namen `Trail1` und der Angabe eines S3-Buckets:

```
aws cloudtrail create-trail --name Trail1 --s3-bucket-name my-bucket --is-multi-region-trail
```

Ausgabe:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- API-Einheiten finden Sie [CreateTrail](#) in der AWS CLI Befehlsreferenz.

delete-trail

Das folgende Codebeispiel zeigt die Verwendung `delete-trail`.

AWS CLI

Um einen Trail zu löschen

Der folgende `delete-trail` Befehl löscht einen Trail mit dem Namen `Trail1`:

```
aws cloudtrail delete-trail --name Trail1
```

- API-Einheiten finden Sie [DeleteTrail](#) in der AWS CLI Befehlsreferenz.

describe-trails

Das folgende Codebeispiel zeigt die Verwendung `describe-trails`.

AWS CLI

Um einen Trail zu beschreiben

Der folgende `describe-trails` Befehl gibt die Einstellungen für `Trail1` und zurück `Trail2`:

```
aws cloudtrail describe-trails --trail-name-list Trail1 Trail2
```

Ausgabe:

```
{
  "trailList": [
    {
      "IncludeGlobalServiceEvents": true,
      "Name": "Trail1",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
      "LogFileValidationEnabled": false,
      "IsMultiRegionTrail": false,
      "S3BucketName": "my-bucket",
      "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/
CloudTrail_CloudWatchLogs_Role",
      "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-
group:CloudTrail:*",
      "SnsTopicName": "my-topic",
      "HomeRegion": "us-east-1"
    },
    {
      "IncludeGlobalServiceEvents": true,
      "Name": "Trail2",
      "S3KeyPrefix": "my-prefix",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
      "LogFileValidationEnabled": false,
      "IsMultiRegionTrail": false,
      "S3BucketName": "my-bucket",
      "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/4c5ae5ac-3c13-421e-8335-c7868ef6a769",
      "HomeRegion": "us-east-1"
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie [DescribeTrails](#) in der AWS CLI Befehlsreferenz.

get-event-selectors

Das folgende Codebeispiel zeigt die Verwendung `get-event-selectors`.

AWS CLI

Um die Event-Selector-Einstellungen für einen Trail anzuzeigen

Der folgende `get-event-selectors` Befehl gibt die Einstellungen für `Trail1` zurück:

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

Ausgabe:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [],
      "ReadWriteType": "All"
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
}
```

- API-Einheiten finden Sie [GetEventSelectors](#) in der AWS CLI Befehlsreferenz.

get-trail-status

Das folgende Codebeispiel zeigt die Verwendung `get-trail-status`.

AWS CLI

Um den Status einer Spur abzurufen

Der folgende `get-trail-status` Befehl gibt die Liefer- und Protokollierungsdetails für `Trail1` zurück:

```
aws cloudtrail get-trail-status --name Trail1
```

Ausgabe:

```
{
  "LatestNotificationTime": 1454022144.869,
  "LatestNotificationAttemptSucceeded": "2016-01-28T23:02:24Z",
  "LatestDeliveryAttemptTime": "2016-01-28T23:02:24Z",
  "LatestDeliveryTime": 1454022144.869,
  "TimeLoggingStarted": "2015-11-06T18:36:38Z",
  "LatestDeliveryAttemptSucceeded": "2016-01-28T23:02:24Z",
  "IsLogging": true,
  "LatestCloudWatchLogsDeliveryTime": 1454022144.918,
  "StartLoggingTime": 1446834998.695,
  "StopLoggingTime": 1446834996.933,
  "LatestNotificationAttemptTime": "2016-01-28T23:02:24Z",
  "TimeLoggingStopped": "2015-11-06T18:36:36Z"
}
```

- API-Einheiten finden Sie [GetTrailStatus](#) in der AWS CLI Befehlsreferenz.

list-public-keys

Das folgende Codebeispiel zeigt die Verwendung `list-public-keys`.

AWS CLI

Um alle öffentlichen Schlüssel für einen Trail aufzulisten

Der folgende `list-public-keys` Befehl gibt alle öffentlichen Schlüssel zurück, deren private Schlüssel innerhalb des angegebenen Zeitraums zum Signieren der Digest-Dateien verwendet wurden:

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

Ausgabe:

```
{
  "PublicKeyList": [
    {
```

```

    "ValidityStartTime": 1453076702.0,
    "ValidityEndTime": 1455668702.0,
    "Value": "MIIBCgKCAQEA1SS3c192HDycr/MTj0mo0has8habjrraXw+Kz1WF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kU11xboEcEPHYoIYMbgfSw7KGnuDKwkLzsQWhUJ0cIb0HASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmU18/P7XpfpWQuk4h8g3pXZ0rNXr081bh4d39svj7Uqdhv0XoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",
    "Fingerprint": "7f3f401420072e50a65a141430817ab3"
  }
]
}

```

- API-Einheiten finden Sie [ListPublicKeys](#) in der AWS CLI Befehlsreferenz.

list-tags

Das folgende Codebeispiel zeigt die Verwendung `list-tags`.

AWS CLI

Um die Tags für einen Trail aufzulisten

Der folgende `list-tags` Befehl listet die Tags für `Trail1` und auf `Trail2`:

```

aws cloudtrail list-tags --resource-id-list arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2

```

Ausgabe:

```

{
  "ResourceTagList": [
    {
      "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
      "TagsList": [
        {
          "Value": "Alice",
          "Key": "name"
        },
        {
          "Value": "us",
          "Key": "location"
        }
      ]
    }
  ]
}

```

```

    }
  ]
},
{
  "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
  "TagsList": [
    {
      "Value": "Bob",
      "Key": "name"
    }
  ]
}
]
}

```

- API-Einheiten finden Sie [ListTags](#) in der AWS CLI Befehlsreferenz.

lookup-events

Das folgende Codebeispiel zeigt die Verwendung `lookup-events`.

AWS CLI

Um nach Ereignissen für einen Trail zu suchen

Der folgende `lookup-events` Befehl sucht nach API Aktivitätsereignissen anhand des Attributs `EventName`:

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

Ausgabe:

```

{
  "Events": [
    {
      "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
      "Username": "my-session-name",
      "EventTime": "2021-11-18T09:41:02-08:00",
      "CloudTrailEvent": "{\"eventVersion\":\"1.02\", \"userIdentity\": {\"type\": \"AssumedRole\", \"principalId\": \"AR0AJIKPFTA72SWU4L7T4:my-session-name\", \"arn\": \"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\", \"accountId\":

```



```

{"123456789012",{"sessionContext":{"attributes":{"mfaAuthenticated":"false"},
"creationDate":"2016-01-26T21:42:12Z"},"sessionIssuer":{"type":"Role",
"principalId":"AROAJIKPFTA72SWU4L7T4","arn":"arn:aws:iam::123456789012:role/
my-role","accountId":"123456789012","userName":"my-role"}}},"eventTime
":"2016-01-26T21:42:12Z","eventSource":"signin.amazonaws.com","eventName":
"ConsoleLogin","awsRegion":"us-east-1","sourceIPAddress":"72.21.198.70",
"userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36","requestParameters
":null,"responseElements":{"ConsoleLogin":{"Success":true},"additionalEventData":
{"MobileVersion":"No","MFAUsed":"No"},"eventID":"654ccbc0-ba0d-486a-9076-
dbf7274677a7","eventType":"AwsConsoleSignIn","recipientAccountId":
"123456789012"}},
  "eventName": "ConsoleLogin",
  "resources": []
}
]
}

```

- API-Einheiten finden Sie [LookupEvents](#) in der AWS CLI Befehlsreferenz.

put-event-selectors

Das folgende Codebeispiel zeigt die Verwendung von `put-event-selectors`.

AWS CLI

Beispiel 1: Konfigurieren Sie einen Trail, um Verwaltungsereignisse und Datenereignisse mithilfe erweiterter Ereignisselektoren zu protokollieren

Sie können erweiterte Event-Selektoren und Bedingungen für Ihre erweiterten Event-Selektoren hinzufügen, bis zu einem Maximum von 500 Werten für alle Bedingungen und Selektoren in einem Trail. Sie können erweiterte Event-Selektoren verwenden, um alle verfügbaren Datenereignistypen zu protokollieren. Sie können entweder erweiterte Ereignisselektoren oder einfache Ereignisselektoren verwenden, aber nicht beide. Wenn Sie erweiterte Ereignisselektoren auf einen Trail anwenden, werden alle vorhandenen grundlegenden Ereignisselektoren überschrieben.

Im folgenden Beispiel wird eine erweiterte Ereignisauswahl für einen Trail mit dem Namen `myTrail` erstellt, um alle Verwaltungsereignisse `myTrail` zu protokollieren, S3 `PutObject` und `DeleteObject` API Aufrufe für alle S3-Buckets bis auf einen zu protokollieren, API Datenaufrufe für eine Lambda-Funktion mit dem Namen `myFunction` zu protokollieren und API `Publish`-Aufrufe zu einem SNS Thema mit dem Namen `myTopic` zu protokollieren.

```
aws cloudtrail put-event-selectors \
  --trail-name myTrail \
  --advanced-event-selectors '[{"Name": "Log all management events",
  "FieldSelectors": [{ "Field": "eventCategory", "Equals": ["Management"] } ] },
  {"Name": "Log PutObject and DeleteObject events for all but one
  bucket", "FieldSelectors": [{ "Field": "eventCategory", "Equals": ["Data"] },
  { "Field": "resources.type", "Equals": ["AWS::S3::Object"] }, { "Field":
  "eventName", "Equals": ["PutObject", "DeleteObject"] }, { "Field": "resources.ARN",
  "NotStartsWith": ["arn:aws:s3:::sample_bucket_name/"] } ]}, {"Name": "Log
  data events for a specific Lambda function", "FieldSelectors": [{ "Field":
  "eventCategory", "Equals": ["Data"] }, { "Field": "resources.type",
  "Equals": ["AWS::Lambda::Function"] }, { "Field": "resources.ARN", "Equals":
  ["arn:aws:lambda:us-east-1:123456789012:function:myFunction"] } ]}, {"Name":
  "Log all Publish API calls on a specific SNS topic", "FieldSelectors":
  [{ "Field": "eventCategory", "Equals": ["Data"] }, { "Field": "resources.type",
  "Equals": ["AWS::SNS::Topic"] }, { "Field": "eventName", "Equals":
  ["Publish"] }, { "Field": "resources.ARN", "Equals": ["arn:aws:sns:us-
  east-1:123456789012:myTopic.fifo"] } ]}]'
```

Ausgabe:

```
{
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/myTrail",
  "AdvancedEventSelectors": [
    {
      "Name": "Log all management events",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [
            "Management"
          ]
        }
      ]
    },
    {
      "Name": "Log PutObject and DeleteObject events for all but one bucket",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [
            "Data"
          ]
        }
      ]
    }
  ]
}
```

```
    },
    {
      "Field": "resources.type",
      "Equals": [
        "AWS::S3::Object"
      ]
    },
    {
      "Field": "eventName",
      "Equals": [
        "PutObject",
        "DeleteObject"
      ]
    },
    {
      "Field": "resources.ARN",
      "NotStartsWith": [
        "arn:aws:s3:::sample_bucket_name/"
      ]
    }
  ]
},
{
  "Name": "Log data events for a specific Lambda function",
  "FieldSelectors": [
    {
      "Field": "eventCategory",
      "Equals": [
        "Data"
      ]
    },
    {
      "Field": "resources.type",
      "Equals": [
        "AWS::Lambda::Function"
      ]
    },
    {
      "Field": "resources.ARN",
      "Equals": [
        "arn:aws:lambda:us-east-1:123456789012:function:myFunction"
      ]
    }
  ]
}
```

```

    },
    {
      "Name": "Log all Publish API calls on a specific SNS topic",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [
            "Data"
          ]
        },
        {
          "Field": "resources.type",
          "Equals": [
            "AWS::SNS::Topic"
          ]
        },
        {
          "Field": "eventName",
          "Equals": [
            "Publish"
          ]
        },
        {
          "Field": "resources.ARN",
          "Equals": [
            "arn:aws:sns:us-east-1:123456789012:myTopic.fifo"
          ]
        }
      ]
    }
  ]
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Ereignisse mithilfe erweiterter Event-Selektoren protokollieren](#). AWS CloudTrail

Beispiel 2: Konfigurieren Sie Event-Selektoren für einen Trail, um alle Verwaltungsereignisse und Datenereignisse zu protokollieren

Sie können bis zu 5 Ereignisauswahlen und bis zu 250 Datenressourcen für einen Trail konfigurieren. Event-Selektoren werden auch als einfache Event-Selektoren bezeichnet. Sie können Ereignisselektoren verwenden, um Verwaltungsereignisse und Datenereignisse für S3-Objekte, Lambda-Funktionen und DynamoDB-Tabellen zu protokollieren. Um Datenereignisse

für andere Ressourcentypen zu protokollieren, müssen Sie erweiterte Ereignisselectoren verwenden.

Das folgende Beispiel erstellt einen Event-Selektor für einen Trail, der so benannt ist, `TrailName` dass er alle Verwaltungsereignisse, Datenereignisse für zwei Amazon S3 S3-Bucket/Präfix-Kombinationen und Datenereignisse für eine einzelne AWS Lambda-Funktion mit dem Namen umfasst. `hello-world-python-function`

```
aws cloudtrail put-event-selectors \
  --trail-name TrailName \
  --event-selectors '[{"ReadWriteType": "All", "IncludeManagementEvents":
true, "DataResources": [{"Type": "AWS::S3::Object", "Values":
["arn:aws:s3:::mybucket/prefix", "arn:aws:s3:::mybucket2/prefix2"]},
{"Type": "AWS::Lambda::Function", "Values": ["arn:aws:lambda:us-
west-2:999999999999:function:hello-world-python-function"]}]]'
```

Ausgabe:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [
        {
          "Values": [
            "arn:aws:s3:::mybucket/prefix",
            "arn:aws:s3:::mybucket2/prefix2"
          ],
          "Type": "AWS::S3::Object"
        },
        {
          "Values": [
            "arn:aws:lambda:us-west-2:123456789012:function:hello-world-
python-function"
          ],
          "Type": "AWS::Lambda::Function"
        }
      ],
      "ReadWriteType": "All"
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
```

}

Weitere Informationen finden Sie im Benutzerhandbuch unter [Protokollieren von Ereignissen mithilfe grundlegender Event-Selektoren](#).AWS CloudTrail

Beispiel 3: Konfigurieren Sie Event-Selektoren für einen Trail, um Verwaltungsereignisse, alle S3-Datenereignisse für S3-Objekte und alle Lambda-Datenereignisse für Funktionen in Ihrem Konto zu protokollieren

Das folgende Beispiel erstellt einen Event-Selektor für einen Trail mit dem Namen `TrailName2`, der alle Verwaltungsereignisse und alle Datenereignisse für alle Amazon S3 S3-Buckets und AWS Lambda-Funktionen im Konto umfasst. AWS

```
aws cloudtrail put-event-selectors \
  --trail-name TrailName2 \
  --event-selectors '[{"ReadWriteType": "All", "IncludeManagementEvents":
    true, "DataResources": [{"Type": "AWS::S3::Object", "Values": ["arn:aws:s3"]},
    {"Type": "AWS::Lambda::Function", "Values": ["arn:aws:lambda"]}]]'
```

Ausgabe:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [
        {
          "Values": [
            "arn:aws:s3"
          ],
          "Type": "AWS::S3::Object"
        },
        {
          "Values": [
            "arn:aws:lambda"
          ],
          "Type": "AWS::Lambda::Function"
        }
      ],
      "ReadWriteType": "All"
    }
  ],
}
```

```
"TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName2"
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Protokollieren von Ereignissen mithilfe grundlegender Event-Selektoren](#).AWS CloudTrail

- APIEinzelheiten finden Sie [PutEventSelectors](#)in der AWS CLI Befehlsreferenz.

remove-tags

Das folgende Codebeispiel zeigt die Verwendungremove-tags.

AWS CLI

Um Tags für einen Trail zu entfernen

Der folgende remove-tags Befehl entfernt die angegebenen Tags fürTrail1:

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- APIEinzelheiten finden Sie [RemoveTags](#)in der AWS CLI Befehlsreferenz.

start-logging

Das folgende Codebeispiel zeigt die Verwendungstart-logging.

AWS CLI

Um mit der Protokollierung für einen Trail zu beginnen

Der folgende start-logging Befehl aktiviert die Protokollierung fürTrail1:

```
aws cloudtrail start-logging --name Trail1
```

- APIEinzelheiten finden Sie [StartLogging](#)in der AWS CLI Befehlsreferenz.

stop-logging

Das folgende Codebeispiel zeigt die Verwendungstop-logging.

AWS CLI

Um die Protokollierung einer Spur zu beenden

Der folgende stop-logging Befehl deaktiviert die Protokollierung für `Trail1`:

```
aws cloudtrail stop-logging --name Trail1
```

- API-Einheiten finden Sie [StopLogging](#) in der AWS CLI Befehlsreferenz.

update-subscription

Das folgende Codebeispiel zeigt die Verwendung `update-subscription`.

AWS CLI

Um die Konfigurationseinstellungen für einen Trail zu aktualisieren

Der folgende `update-subscription` Befehl aktualisiert den Trail, um einen neuen S3-Bucket und ein neues SNS Thema anzugeben:

```
aws cloudtrail update-subscription --name Trail1 --s3-new-bucket my-bucket-new --  
sns-new-topic my-topic-new
```

Ausgabe:

```
Setting up new S3 bucket my-bucket-new...  
Setting up new SNS topic my-topic-new...  
Creating/updating CloudTrail configuration...  
CloudTrail configuration:  
{  
  "trailList": [  
    {  
      "IncludeGlobalServiceEvents": true,  
      "Name": "Trail1",  
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",  
      "LogFileValidationEnabled": false,  
      "IsMultiRegionTrail": false,  
      "S3BucketName": "my-bucket-new",  
      "SnsTopicName": "my-topic-new",  
      "HomeRegion": "us-east-1"  
    }  
  ]  
}
```



```
],
"ResponseMetadata": {
"HTTPStatusCode": 200,
"RequestId": "31126f8a-c616-11e5-9cc6-2fd637936879"
}
}
```

- API-Einheiten finden Sie [UpdateSubscription](#) in der AWS CLI Befehlsreferenz.

update-trail

Das folgende Codebeispiel zeigt die Verwendung `update-trail`.

AWS CLI

Um einen Trail zu aktualisieren

Mit dem folgenden `update-trail` Befehl wird ein Trail aktualisiert, sodass er einen vorhandenen Bucket für die Protokollzustellung verwendet:

```
aws cloudtrail update-trail --name Trail1 --s3-bucket-name my-bucket
```

Ausgabe:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- API-Einheiten finden Sie [UpdateTrail](#) in der AWS CLI Befehlsreferenz.

validate-logs

Das folgende Codebeispiel zeigt die Verwendung `validate-logs`.

AWS CLI

Um eine Protokolldatei zu validieren

Der folgende `validate-logs` Befehl validiert die Protokolle für `Trail1`:

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

Ausgabe:

```
Validating log files for trail arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 between 2016-01-29T19:00:00Z and 2016-01-29T22:15:43Z
Results requested for 2016-01-29T19:00:00Z to 2016-01-29T22:15:43Z
Results found for 2016-01-29T19:24:57Z to 2016-01-29T21:24:57Z:
3/3 digest files valid
15/15 log files valid
```

- API-Einheiten finden Sie [ValidateLogs](#) in der AWS CLI Befehlsreferenz.

CloudWatch Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CloudWatch.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-alarms

Das folgende Codebeispiel zeigt die Verwendung `delete-alarms`.

AWS CLI

So löschen Sie einen Alarm

Im folgenden Beispiel wird der `delete-alarms` Befehl verwendet, um den CloudWatch Amazon-Alarm mit dem Namen „myalarm“ zu löschen:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Ausgabe:

```
This command returns to the prompt if successful.
```

- API-Einheiten finden Sie [DeleteAlarms](#) in der AWS CLI Befehlsreferenz.

describe-alarm-history

Das folgende Codebeispiel zeigt die Verwendung `describe-alarm-history`.

AWS CLI

So rufen Sie den Verlauf eines Alarms ab

Im folgenden Beispiel wird der `describe-alarm-history` Befehl verwendet, um den Verlauf für den CloudWatch Amazon-Alarm mit dem Namen „myalarm“ abzurufen:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type StateUpdate
```

Ausgabe:

```
{
  "AlarmHistoryItems": [
    {
      "Timestamp": "2014-04-09T18:59:06.442Z",
      "HistoryItemType": "StateUpdate",
      "AlarmName": "myalarm",
      "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not greater than the threshold (70.0). The most recent datapoints: [38.958, 40.292].\",\"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],\"threshold\":70.0}}}\",
      "HistorySummary": "Alarm updated from ALARM to OK"
    },
  ],
}
```

```

    {
      "Timestamp": "2014-04-09T18:59:05.805Z",
      "HistoryItemType": "StateUpdate",
      "AlarmName": "myalarm",
      "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue\": \"OK\"}, \"stateReason\": \"Threshold Crossed: 2 datapoints were not greater than the threshold (70.0). The most recent datapoints: [38.839999999999996, 39.714].\", \"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\", \"startDate\":\"2014-03-11T22:30:00.000+0000\", \"statistic\":\"Average\", \"period\":300, \"recentDatapoints\":[38.839999999999996,39.714], \"threshold\":70.0}}, \"newState\":{\"stateValue\":\"ALARM\", \"stateReason\":\"testing purposes\"}}",
      "HistorySummary": "Alarm updated from OK to ALARM"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeAlarmHistory](#) in der AWS CLI Befehlsreferenz.

describe-alarms-for-metric

Das folgende Codebeispiel zeigt die Verwendung `describe-alarms-for-metric`.

AWS CLI

So zeigen Sie Informationen über Alarme an, die einer Metrik zugeordnet sind

Im folgenden Beispiel wird der `describe-alarms-for-metric` Befehl verwendet, um Informationen zu allen Alarmen anzuzeigen, die mit der EC2 CPUUtilization Amazon-Metrik und der Instance mit der ID `i-0c986c72` verknüpft sind. :

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Ausgabe:

```

{
  "MetricAlarms": [
    {
      "EvaluationPeriods": 10,
      "AlarmArn": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:myHighCpuAlarm2",

```

```

    "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
    "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
    "ComparisonOperator": "GreaterThanOrEqualToThreshold",
    "AlarmActions": [
      "arn:aws:sns:us-east-1:111122223333:NotifyMe"
    ],
    "Namespace": "AWS/EC2",
    "AlarmDescription": "CPU usage exceeds 70 percent",
    "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\\\"2013-10-30T03:03:51.479+0000\\\",\\\"startDate\\\":\\\"2013-10-30T02:08:00.000+0000\\\",
\\\"statistic\\\":\\\"Average\\\",\\\"period\\\":300,\\\"recentDatapoints\\\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\\\"threshold\\\":70.0}\",
    "Period": 300,
    "StateValue": "OK",
    "Threshold": 70.0,
    "AlarmName": "myHighCpuAlarm2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-0c986c72"
      }
    ],
    "Statistic": "Average",
    "StateReason": "Threshold Crossed: 10 datapoints were not greater than
or equal to the threshold (70.0). The most recent datapoints: [40.760000000000005,
41.316].",
    "InsufficientDataActions": [],
    "OKActions": [],
    "ActionsEnabled": true,
    "MetricName": "CPUUtilization"
  },
  {
    "EvaluationPeriods": 2,
    "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
    "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
    "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
    "ComparisonOperator": "GreaterThanThreshold",
    "AlarmActions": [
      "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
    ],
    "Namespace": "AWS/EC2",
    "AlarmDescription": "CPU usage exceeds 70 percent",

```

```

        "StateReasonData": "{ \"version\": \"1.0\", \"queryDate\":
        \\\"2014-04-09T18:59:06.419+0000\\\", \"startDate\": \\\"2014-04-09T18:44:00.000+0000\\\",
        \\\"statistic\": \\\"Average\\\", \\\"period\": 300, \\\"recentDatapoints\": [38.958, 40.292],
        \\\"threshold\": 70.0} \",
        "Period": 300,
        "StateValue": "OK",
        "Threshold": 70.0,
        "AlarmName": "myHighCpuAlarm",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-0c986c72"
            }
        ],
        "Statistic": "Average",
        "StateReason": "Threshold Crossed: 2 datapoints were not greater than
        the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
        "InsufficientDataActions": [],
        "OKActions": [],
        "ActionsEnabled": false,
        "MetricName": "CPUUtilization"
    }
]
}

```

- API-Einheiten finden Sie [DescribeAlarmsForMetric](#) in der AWS CLI Befehlsreferenz.

describe-alarms

Das folgende Codebeispiel zeigt die Verwendung `describe-alarms`.

AWS CLI

So listen Sie Informationen über einen Alarm auf

Im folgenden Beispiel wird der `describe-alarms`-Befehl verwendet, um Informationen über den Alarm mit dem Namen „myalarm“ bereitzustellen:

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Ausgabe:

```

{
  "MetricAlarms": [
    {
      "EvaluationPeriods": 2,
      "AlarmArn": "arn:aws:cloudwatch:us-east-1:123456789012:alarm:myalarm",
      "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
      "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
      "ComparisonOperator": "GreaterThanThreshold",
      "AlarmActions": [
        "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
      ],
      "Namespace": "AWS/EC2",
      "AlarmDescription": "CPU usage exceeds 70 percent",
      "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],\"threshold\":70.0}\",
      "Period": 300,
      "StateValue": "OK",
      "Threshold": 70.0,
      "AlarmName": "myalarm",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-0c986c72"
        }
      ],
      "Statistic": "Average",
      "StateReason": "Threshold Crossed: 2 datapoints were not greater than the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
      "InsufficientDataActions": [],
      "OKActions": [],
      "ActionsEnabled": true,
      "MetricName": "CPUUtilization"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeAlarms](#) in der AWS CLI Befehlsreferenz.

disable-alarm-actions

Das folgende Codebeispiel zeigt die Verwendung `disable-alarm-actions`.

AWS CLI

So deaktivieren Sie Aktionen für einen Alarm

Das folgende Beispiel verwendet den `disable-alarm-actions`-Befehl, um alle Aktionen für den Alarm mit dem Namen „myalarm“ zu deaktivieren:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

- API-Einheiten finden Sie [DisableAlarmActions](#) in der AWS CLI Befehlsreferenz.

enable-alarm-actions

Das folgende Codebeispiel zeigt die Verwendung `enable-alarm-actions`.

AWS CLI

So aktivieren Sie alle Aktionen für einen Alarm

Das folgende Beispiel verwendet den `enable-alarm-actions`-Befehl, um alle Aktionen für den Alarm mit dem Namen „myalarm“ zu aktivieren:

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

- API-Einheiten finden Sie [EnableAlarmActions](#) in der AWS CLI Befehlsreferenz.

get-metric-statistics

Das folgende Codebeispiel zeigt die Verwendung `get-metric-statistics`.

AWS CLI

Um die CPU Auslastung pro EC2 Instanz zu ermitteln

Im folgenden Beispiel wird der `get-metric-statistics` Befehl verwendet, um die CPU Auslastung für eine EC2 Instanz mit der ID `i-abcdef` abzurufen.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Ausgabe:

```
{
  "Datapoints": [
    {
      "Timestamp": "2014-04-09T11:18:00Z",
      "Maximum": 44.79,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T20:18:00Z",
      "Maximum": 47.92,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T19:18:00Z",
      "Maximum": 50.85,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T09:18:00Z",
      "Maximum": 47.92,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T03:18:00Z",
      "Maximum": 76.84,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T21:18:00Z",
      "Maximum": 48.96,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T14:18:00Z",
```

```
    "Maximum": 47.92,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T08:18:00Z",  
    "Maximum": 47.92,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T16:18:00Z",  
    "Maximum": 45.55,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T06:18:00Z",  
    "Maximum": 47.92,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T13:18:00Z",  
    "Maximum": 45.08,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T05:18:00Z",  
    "Maximum": 47.92,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T18:18:00Z",  
    "Maximum": 46.88,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T17:18:00Z",  
    "Maximum": 52.08,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-04-09T07:18:00Z",  
    "Maximum": 47.92,  
    "Unit": "Percent"  
  },  
  {
```

```
    "Timestamp": "2014-04-09T02:18:00Z",
    "Maximum": 51.23,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-09T12:18:00Z",
    "Maximum": 47.67,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-08T23:18:00Z",
    "Maximum": 46.88,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-09T10:18:00Z",
    "Maximum": 51.91,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-09T04:18:00Z",
    "Maximum": 47.13,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-09T15:18:00Z",
    "Maximum": 48.96,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-09T00:18:00Z",
    "Maximum": 48.16,
    "Unit": "Percent"
  },
  {
    "Timestamp": "2014-04-09T01:18:00Z",
    "Maximum": 49.18,
    "Unit": "Percent"
  }
],
"Label": "CPUUtilization"
}
```

Angeben mehrerer Dimensionen

Das folgende Beispiel zeigt, wie mehrere Dimensionen angegeben werden können. Jede Dimension wird als Name/Wert-Paar mit einem Komma zwischen dem Namen und dem Wert angegeben. Mehrere Dimensionen sind durch ein Leerzeichen getrennt. Wenn eine einzelne Metrik mehrere Dimensionen enthält, müssen Sie für jede definierte Dimension einen Wert angeben.

Weitere Beispiele für die Verwendung des `get-metric-statistics` Befehls finden Sie unter [Get Statistics for a Metric](#) im Amazon CloudWatch Developer Guide.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace
--dimensions Name=InstanceID,Value=i-abcdef Name=InstanceType,Value=m1.small --
start-time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average
--period 60
```

- API-Einheiten finden Sie [GetMetricStatistics](#) in der AWS CLI Befehlsreferenz.

list-metrics

Das folgende Codebeispiel zeigt die Verwendung `list-metrics`.

AWS CLI

Um die Metriken für Amazon aufzulisten SNS

Das folgende `list-metrics` Beispiel zeigt die Metriken für Amazon SNS.

```
aws cloudwatch list-metrics \
--namespace "AWS/SNS"
```

Ausgabe:

```
{
  "Metrics": [
    {
      "Namespace": "AWS/SNS",
      "Dimensions": [
        {
          "Name": "TopicName",
          "Value": "NotifyMe"
        }
      ]
    }
  ]
}
```

```
    ],
    "MetricName": "PublishSize"
  },
  {
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "CF0"
      }
    ],
    "MetricName": "PublishSize"
  },
  {
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "NotifyMe"
      }
    ],
    "MetricName": "NumberOfNotificationsFailed"
  },
  {
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "NotifyMe"
      }
    ],
    "MetricName": "NumberOfNotificationsDelivered"
  },
  {
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "NotifyMe"
      }
    ],
    "MetricName": "NumberOfMessagesPublished"
  },
  {
```

```
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "CF0"
      }
    ],
    "MetricName": "NumberOfMessagesPublished"
  },
  {
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "CF0"
      }
    ],
    "MetricName": "NumberOfNotificationsDelivered"
  },
  {
    "Namespace": "AWS/SNS",
    "Dimensions": [
      {
        "Name": "TopicName",
        "Value": "CF0"
      }
    ],
    "MetricName": "NumberOfNotificationsFailed"
  }
]
```

- API-Einheiten finden Sie [ListMetrics](#) in der AWS CLI Befehlsreferenz.

put-metric-alarm

Das folgende Codebeispiel zeigt die Verwendung `put-metric-alarm`.

AWS CLI

Um eine E-Mail-Nachricht von Amazon Simple Notification Service zu senden, wenn die CPU Auslastung 70 Prozent übersteigt

Im folgenden Beispiel wird der `put-metric-alarm` Befehl verwendet, um eine E-Mail-Nachricht von Amazon Simple Notification Service zu senden, wenn die CPU Auslastung 70 Prozent übersteigt:

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück. Wenn ein Alarm mit demselben Namen bereits vorhanden ist, wird er durch den neuen Alarm überschrieben.

So geben Sie mehrere Dimensionen an

Das folgende Beispiel zeigt, wie mehrere Dimensionen angegeben werden können. Jede Dimension wird als Name/Wert-Paar mit einem Komma zwischen dem Namen und dem Wert angegeben. Mehrere Dimensionen werden durch ein Leerzeichen getrennt:

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-description "The default example alarm" --namespace "CW EXAMPLE METRICS" --metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions Name=key1,Value=value1 Name=key2,Value=value2
```

- API-Einheiten finden Sie [PutMetricAlarm](#) unter AWS CLI Befehlsreferenz.

put-metric-data

Das folgende Codebeispiel zeigt die Verwendung `put-metric-data`.

AWS CLI

Um eine benutzerdefinierte Metrik auf Amazon zu veröffentlichen CloudWatch

Im folgenden Beispiel wird der `put-metric-data` Befehl verwendet, um eine benutzerdefinierte Metrik auf Amazon zu veröffentlichen CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://metric.json
```

Die Werte für die Metrik selbst werden in der JSON Datei gespeichert, `metric.json`.

Hier ist der Inhalt dieser Datei:

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Weitere Informationen finden Sie unter Veröffentlichen benutzerdefinierter Metriken im Amazon CloudWatch Developer Guide.

So geben Sie mehrere Dimensionen an

Das folgende Beispiel zeigt, wie mehrere Dimensionen angegeben werden können. Jede Dimension wird als Name/Wert-Paar angegeben. Mehrere Dimensionen sind durch ein Komma getrennt:

```
aws cloudwatch put-metric-data --metric-name Buffers --
namespace MyNameSpace --unit Bytes --value 231434333 --
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- API-Einheiten finden Sie [PutMetricData](#) in der AWS CLI Befehlsreferenz.

set-alarm-state

Das folgende Codebeispiel zeigt die Verwendung `set-alarm-state`.

AWS CLI

Um den Status eines Alarms vorübergehend zu ändern

Im folgenden Beispiel wird der `set-alarm-state` Befehl verwendet, um den Status eines CloudWatch Amazon-Alarm mit dem Namen „myalarm“ vorübergehend zu ändern und ihn zu Testzwecken auf den ALARM Status zu setzen:


```
aws cloudwatch set-alarm-state --alarm-name "myAlarm" --state-value ALARM --state-reason "testing purposes"
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

- API-Einheiten finden Sie [SetAlarmState](#) in der AWS CLI Befehlsreferenz.

CloudWatch Log-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with CloudWatch Logs Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-log-group

Das folgende Codebeispiel zeigt die Verwendung `create-log-group`.

AWS CLI

Der folgende Befehl erstellt eine Protokollgruppe mit dem Namen `my-logs`:

```
aws logs create-log-group --log-group-name my-logs
```

- API-Einheiten finden Sie [CreateLogGroup](#) in der AWS CLI Befehlsreferenz.

create-log-stream

Das folgende Codebeispiel zeigt die Verwendung `create-log-stream`.

AWS CLI

Der folgende Befehl erstellt einen Protokollstream mit dem Namen `20150601` in der Protokollgruppe `my-logs`:

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- API-Einheiten finden Sie [CreateLogStream](#) in der AWS CLI Befehlsreferenz.

delete-log-group

Das folgende Codebeispiel zeigt die Verwendung `delete-log-group`.

AWS CLI

Der folgende Befehl löscht eine Protokollgruppe mit dem Namen `my-logs`:

```
aws logs delete-log-group --log-group-name my-logs
```

- API-Einheiten finden Sie [DeleteLogGroup](#) in der AWS CLI Befehlsreferenz.

delete-log-stream

Das folgende Codebeispiel zeigt die Verwendung `delete-log-stream`.

AWS CLI

Der folgende Befehl löscht einen Protokollstream mit dem Namen `20150531` aus einer Protokollgruppe mit dem Namen `my-logs`:

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- API-Einheiten finden Sie [DeleteLogStream](#) in der AWS CLI Befehlsreferenz.

delete-retention-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-retention-policy`.

AWS CLI

Mit dem folgenden Befehl wird die Aufbewahrungsrichtlinie entfernt, die zuvor auf eine Protokollgruppe mit dem Namen angewendet wurde `my-logs`:

```
aws logs delete-retention-policy --log-group-name my-logs
```

- API-Einheiten finden Sie [DeleteRetentionPolicy](#) in der AWS CLI Befehlsreferenz.

describe-log-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-log-groups`.

AWS CLI

Der folgende Befehl beschreibt eine Protokollgruppe mit dem Namen `my-logs`:

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Ausgabe:

```
{
  "logGroups": [
    {
      "storedBytes": 0,
      "metricFilterCount": 0,
      "creationTime": 1433189500783,
      "logGroupName": "my-logs",
      "retentionInDays": 5,
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeLogGroups](#) in der AWS CLI Befehlsreferenz.

describe-log-streams

Das folgende Codebeispiel zeigt die Verwendung `describe-log-streams`.

AWS CLI

Der folgende Befehl zeigt alle Protokollstreams, die mit dem Präfix 2015 in der Protokollgruppe `my-logs` beginnen:

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

Ausgabe:

```
{
  "logStreams": [
    {
      "creationTime": 1433189871774,
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-stream:20150531",
      "logStreamName": "20150531",
      "storedBytes": 0
    },
    {
      "creationTime": 1433189873898,
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-stream:20150601",
      "logStreamName": "20150601",
      "storedBytes": 0
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeLogStreams](#) in der AWS CLI Befehlsreferenz.

get-log-events

Das folgende Codebeispiel zeigt die Verwendung `get-log-events`.

AWS CLI

Der folgende Befehl ruft Protokollereignisse aus einem Protokollstream ab, der `20150601` in der Protokollgruppe `my-logs` benannt ist:

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Ausgabe:

```
{
  "nextForwardToken":
  "f/31961209122447488583055879464742346735121166569214640130",
  "events": [
    {
      "ingestionTime": 1433190494190,
      "timestamp": 1433190184356,
      "message": "Example Event 1"
    },
    {
      "ingestionTime": 1433190516679,
      "timestamp": 1433190184356,
      "message": "Example Event 1"
    },
    {
      "ingestionTime": 1433190494190,
      "timestamp": 1433190184358,
      "message": "Example Event 2"
    }
  ],
  "nextBackwardToken":
  "b/31961209122358285602261756944988674324553373268216709120"
}
```

- API-Einheiten finden Sie [GetLogEvents](#) in der AWS CLI Befehlsreferenz.

put-log-events

Das folgende Codebeispiel zeigt die Verwendung `put-log-events`.

AWS CLI

Mit dem folgenden Befehl werden Protokollereignisse in einem Protokollstream gespeichert, der `20150601` in der Protokollgruppe benannt ist `my-logs`:

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-
events file://events
```

Ausgabe:

```
{
  "nextSequenceToken": "49542672486831074009579604567656788214806863282469607346"
}
```

Das obige Beispiel liest eine JSON Reihe von Ereignissen aus einer Datei mit dem Namen `events` im aktuellen Verzeichnis:

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

Für jeden nachfolgenden Aufruf muss das nächste Sequenz-Token, das vom vorherigen Aufruf bereitgestellt wurde, mit der Sequenz-Token-Option angegeben werden:

```
aws logs put-log-events --log-group-name my-logs --log-
stream-name 20150601 --log-events file://events2 --sequence-
token "49542672486831074009579604567656788214806863282469607346"
```

Ausgabe:

```
{
  "nextSequenceToken": "49542672486831074009579604567900991230369019956308219826"
}
```

- API-Einheiten finden Sie [PutLogEvents](#) in der AWS CLI Befehlsreferenz.

put-retention-policy

Das folgende Codebeispiel zeigt die Verwendung `put-retention-policy`.

AWS CLI

Mit dem folgenden Befehl wird einer Protokollgruppe mit dem Namen eine Aufbewahrungsrichtlinie für 5 Tage hinzugefügt `my-logs`:

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- API-Einheiten finden Sie [PutRetentionPolicy](#) in der AWS CLI Befehlsreferenz.

CloudWatch Beispiele für Netzwerküberwachung mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit CloudWatch Network Monitoring Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-monitor

Das folgende Codebeispiel zeigt die Verwendung `create-monitor`.

AWS CLI

Beispiel 1: Um einen Netzwerkmonitor mit einem Aggregationszeitraum zu erstellen

Im folgenden `create-monitor` Beispiel wird ein Monitor `Example_NetworkMonitor` mit einem auf 30 Sekunden `aggregationPeriod` festgelegten Namen erstellt. Der `state` Anfangsbuchstabe des Monitors ist darauf `INACTIVE` zurückzuführen, dass ihm keine Sonden zugeordnet sind. Der Status ändert sich `ACTIVE` nur, wenn Sonden hinzugefügt werden. Sie können die Befehle [update-monitor](#) oder [create-probe verwenden, um diesem Monitor Sonden hinzuzufügen](#).

```
aws networkmonitor create-monitor \
  --monitor-name Example_NetworkMonitor \
  --aggregation-period 30
```

Ausgabe:

```
{
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
  "monitorName": "Example_NetworkMonitor",
  "state": "INACTIVE",
  "aggregationPeriod": 30,
  "tags": {}
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

Beispiel 2: So erstellen Sie einen Netzwerkmonitor mit einer Sonde, die Tags verwendet TCP und diese auch enthält

Im folgenden `create-monitor` Beispiel wird ein Monitor mit dem Namen `Example_NetworkMonitor` erstellt. Der Befehl erstellt außerdem einen Prüfpunkt, der das ICMP Protokoll verwendet und Tags enthält. Da in der Anfrage „no“ übergeben `aggregationPeriod` wird, ist die Standardeinstellung 60 Sekunden. Der `state` des Monitors mit der Sonde wird `PENDING` solange sein, bis der Monitor es ist `ACTIVE`. Dies kann mehrere Minuten dauern. Ab diesem Zeitpunkt wechseln `state` Sie zu `ACTIVE`, und Sie können mit der Anzeige der CloudWatch Messwerte beginnen.

```
aws networkmonitor create-monitor \
  --monitor-name Example_NetworkMonitor \
  --probes sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,probeTags={Name=Prob
  \
  --tags Monitor=Monitor1
```

Ausgabe:

```
{
  "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/
Example_NetworkMonitor",
```



```

    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 60,
    "tags": {
      "Monitor": "Monitor1"
    }
  }
}

```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

Beispiel 3: So erstellen Sie einen Netzwerkmonitor mit einer Sonde, die Tags verwendet ICMP und diese auch enthält

Im folgenden `create-monitor` Beispiel wird ein Monitor erstellt, der `Example_NetworkMonitor` mit einer Zahl `aggregationPeriod` von 30 Sekunden benannt ist. Der Befehl erstellt außerdem einen Prüfpunkt, der das ICMP Protokoll verwendet und Tags enthält. Da in der Anfrage „no“ übergeben `aggregationPeriod` wird, ist die Standardeinstellung 60 Sekunden. Der `state` des Monitors mit der Sonde wird `PENDING` solange sein, bis der Monitor es ist `ACTIVE`. Dies kann mehrere Minuten dauern. Ab diesem Zeitpunkt wechseln `state` Sie zu `ACTIVE`, und Sie können mit der Anzeige der CloudWatch Messwerte beginnen.

```

aws networkmonitor create-monitor \
  --monitor-name Example_NetworkMonitor \
  --aggregation-period 30 \
  --probes sourceArn=arn:aws:ec2:region111122223333:subnet/subnet-  

id,destination=10.0.0.100,protocol=ICMP,packetSize=56,probeTags={Name=Probe1} \
  --tags Monitor=Monitor1

```

Ausgabe:

```

{
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/  

Example_NetworkMonitor",
  "monitorName": "Example_NetworkMonitor",
  "state": "PENDING",
  "aggregationPeriod": 30,
  "tags": {
    "Monitor": "Monitor1"
  }
}

```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateMonitor](#) in der AWS CLI Befehlsreferenz.

create-probe

Das folgende Codebeispiel zeigt die Verwendung `create-probe`.

AWS CLI

Beispiel 1: So erstellen Sie einen Test, der `TCP` verwendet, und fügen ihn einem Netzwerkmonitor hinzu

Im folgenden `create-probe` Beispiel wird ein Prüfpunkt erstellt, der den `TCP` verwendet, und der Test wird einem Monitor mit dem Namen `Example_NetworkMonitor` hinzugefügt. Nach der Erstellung bleibt der `state` des Monitors mit der Sonde bestehen, `PENDING` bis der Monitor es ist `ACTIVE`. Dies kann mehrere Minuten dauern. Zu diesem Zeitpunkt wird der Status geändert `ACTIVE`, und Sie können mit der Anzeige der CloudWatch Messwerte beginnen.

```
aws networkmonitor create-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-  
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,tags={Name=Probe1}
```

Ausgabe:

```
{  
  "probeId": "probe-12345",  
  "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",  
  "destination": "10.0.0.100",  
  "destinationPort": 80,  
  "packetSize": 56,  
  "addressFamily": "IPV4",  
  "vpcId": "vpc-12345",  
  "state": "PENDING",  
  "createdAt": "2024-03-29T12:41:57.314000-04:00",  
  "modifiedAt": "2024-03-29T12:41:57.314000-04:00",  
  "tags": {  
    "Name": "Probe1"  
  }  
}
```

```
}
```

Beispiel 2: So erstellen Sie einen Test, der Probe using verwendet, ICMP und fügen ihn einem Netzwerkmonitor hinzu

Im folgenden `create-probe` Beispiel wird ein Test erstellt, der den verwendet, ICMP protocol und der Test wird einem Monitor mit dem Namen hinzugefügt `Example_NetworkMonitor`. Nach der Erstellung bleibt der state des Monitors mit der Sonde bestehen, PENDING bis der Monitor es ist ACTIVE. Dies kann mehrere Minuten dauern. Zu diesem Zeitpunkt wird der Status geändert ACTIVE, und Sie können mit der Anzeige der CloudWatch Messwerte beginnen.

```
aws networkmonitor create-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe sourceArn=arn:aws:ec2:region:012345678910:subnet/subnet-  
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,tags={Name=Probe1}
```

Ausgabe:

```
{  
  "probeId": "probe-12345",  
  "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",  
  "destination": "10.0.0.100",  
  "packetSize": 56,  
  "addressFamily": "IPV4",  
  "vpcId": "vpc-12345",  
  "state": "PENDING",  
  "createdAt": "2024-03-29T12:44:02.452000-04:00",  
  "modifiedAt": "2024-03-29T12:44:02.452000-04:00",  
  "tags": {  
    "Name": "Probe1"  
  }  
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API Einheiten finden Sie [CreateProbe](#) in der AWS CLI Befehlsreferenz.

delete-monitor

Das folgende Codebeispiel zeigt die Verwendung `delete-monitor`.

AWS CLI

Um einen Monitor zu löschen

Im folgenden `delete-monitor` Beispiel wird ein Monitor mit dem Namen `Example_NetworkMonitor` gelöscht.

```
aws networkmonitor delete-monitor \  
  --monitor-name Example_NetworkMonitor
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteMonitor](#) in der AWS CLI Befehlsreferenz.

`delete-probe`

Das folgende Codebeispiel zeigt die Verwendung `delete-probe`.

AWS CLI

Um eine Sonde zu löschen

Im folgenden `delete-probe` Beispiel wird ein Prüfpunkt mit der ID `probe-12345` aus einem Netzwerkmonitor mit dem Namen `Example_NetworkMonitor` gelöscht.

```
aws networkmonitor delete-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteProbe](#) in der AWS CLI Befehlsreferenz.

`get-monitor`

Das folgende Codebeispiel zeigt die Verwendung `get-monitor`.

AWS CLI

Um Monitorinformationen abzurufen

Im folgenden `get-monitor` Beispiel werden Informationen über einen Monitor mit dem Namen `Example_NetworkMonitor` abgerufen.

```
aws networkmonitor get-monitor \  
  --monitor-name Example_NetworkMonitor
```

Ausgabe:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "ACTIVE",  
  "aggregationPeriod": 60,  
  "tags": {},  
  "probes": [],  
  "createdAt": "2024-04-01T17:58:07.211000-04:00",  
  "modifiedAt": "2024-04-01T17:58:07.211000-04:00"  
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMonitor](#) in der AWS CLI Befehlsreferenz.

get-probe

Das folgende Codebeispiel zeigt die Verwendung `get-probe`.

AWS CLI

Um die Details der Sonde anzuzeigen

Im folgenden `get-probe` Beispiel werden Details zu einem Prüfpunkt zurückgegeben `probeIDprobe-12345`, der einem Monitor mit dem Namen zugeordnet ist `Example_NetworkMonitor`.

```
aws networkmonitor get-probe \  
  --monitor-name Example_NetworkMonitor
```

```
--monitor-name Example_NetworkMonitor \  
--probe-id probe-12345
```

Ausgabe:

```
{  
  "probeId": "probe-12345",  
  "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",  
  "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",  
  "destination": "10.0.0.100",  
  "destinationPort": 80,  
  "protocol": "TCP",  
  "packetSize": 56,  
  "addressFamily": "IPV4",  
  "vpcId": "vpc-12345",  
  "state": "ACTIVE",  
  "createdAt": "2024-03-29T12:41:57.314000-04:00",  
  "modifiedAt": "2024-03-29T12:42:28.610000-04:00",  
  "tags": {  
    "Name": "Probe1"  
  }  
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetProbe](#) in der AWS CLI Befehlsreferenz.

list-monitors

Das folgende Codebeispiel zeigt die Verwendung `list-monitors`.

AWS CLI

Beispiel 1: Um alle Monitore aufzulisten (einzelner Monitor)

Das folgende `list-monitors` Beispiel gibt eine Liste mit nur einem einzigen Monitor zurück. Der Monitor `state` ist `ACTIVE` und er hat eine `aggregationPeriod` von 60 Sekunden.

```
aws networkmonitor list-monitors
```

Ausgabe:

```
{
  "monitors": [{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "ACTIVE",
    "aggregationPeriod": 60,
    "tags": {
      "Monitor": "Monitor1"
    }
  ]
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

Beispiel 2: Um alle Monitore aufzulisten (mehrere Monitore)

Das folgende `list-monitors` Beispiel gibt eine Liste mit drei Monitoren zurück. Der `state` eines Monitors ist `ACTIVE` und generiert CloudWatch Metriken. Die Status der anderen beiden Monitore sind `INACTIVE` und generieren keine CloudWatch Metriken. Alle drei Monitore verwenden einen `aggregationPeriod` Wert von 60 Sekunden.

```
aws networkmonitor list-monitors
```

Ausgabe:

```
{
  "monitors": [
    {
      "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor",
      "monitorName": "Example_NetworkMonitor",
      "state": "INACTIVE",
      "aggregationPeriod": 60,
      "tags": {}
    },
    {
      "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor2",
```

```

        "monitorName": "Example_NetworkMonitor2",
        "state": "ACTIVE",
        "aggregationPeriod": 60,
        "tags": {
            "Monitor": "Monitor1"
        }
    },
    {
        "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
TestNetworkMonitor_CLI",
        "monitorName": "TestNetworkMonitor_CLI",
        "state": "INACTIVE",
        "aggregationPeriod": 60,
        "tags": {}
    }
]
}

```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ListMonitors](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel gibt eine Liste der Tags für einen Monitor mit dem Namen zurück `Example_NetworkMonitor`.

```

aws networkmonitor list-tags-for-resource \
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor

```

Ausgabe:

```

{
  "tags": {

```



```
    "Environment": "Dev",  
    "Application": "PetStore"  
  }  
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird ein Monitor mit Tags versehen, der `Example_NetworkMonitor` mit `Environment=Dev` und `Application=PetStore` tags benannt ist.

```
aws networkmonitor tag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
  --tags Environment=Dev,Application=PetStore
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung einer Ressource aufzuheben

Im folgenden `untag-resource` Beispiel wird ein `tag-keys` Parameter mit dem Schlüssel-Wert-Paar von `Environment Application` aus seiner Zuordnung zu einem Monitor mit dem Namen `Example_NetworkMonitor` entfernt.

```
aws networkmonitor untag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
  --tag-keys Environment Application
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-monitor

Das folgende Codebeispiel zeigt die Verwendung `update-monitor`.

AWS CLI

Um einen Monitor zu aktualisieren

Das folgende `update-monitor` Beispiel ändert die Werte eines Monitors `aggregationPeriod` von 60 Sekunden auf 30 Sekunden.

```
aws networkmonitor update-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

Ausgabe:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateMonitor](#) in der AWS CLI Befehlsreferenz.

update-probe

Das folgende Codebeispiel zeigt die Verwendung `update-probe`.

AWS CLI

Um eine Sonde zu aktualisieren

Im folgenden `update-probe` Beispiel wird die ursprüngliche `destination` IP-Adresse einer Probe aktualisiert und auch die Adresse `packetSize` zu `aktualisiert60`.

```
aws networkmonitor update-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345 \  
  --destination 10.0.0.150 \  
  --packet-size 60
```

Ausgabe:

```
{  
  "probeId": "probe-12345",  
  "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",  
  "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",  
  "destination": "10.0.0.150",  
  "destinationPort": 80,  
  "protocol": "TCP",  
  "packetSize": 60,  
  "addressFamily": "IPV4",  
  "vpcId": "vpc-12345",  
  "state": "PENDING",  
  "createdAt": "2024-03-29T12:41:57.314000-04:00",  
  "modifiedAt": "2024-03-29T13:52:23.115000-04:00",  
  "tags": {  
    "Name": "Probe1"  
  }  
}
```

```
  }  
}
```

Weitere Informationen finden Sie unter [So funktioniert Amazon CloudWatch Network Monitor](#) im CloudWatch Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateProbe](#) in der AWS CLI Befehlsreferenz.

CodeArtifact Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CodeArtifact.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-external-connection

Das folgende Codebeispiel zeigt die Verwendung `associate-external-connection`.

AWS CLI

Um eine externe Verbindung zu einem Repository hinzuzufügen

Das folgende `associate-external-connection` Beispiel fügt eine externe Verbindung zu npmjs.com zu einem Repository namens `test-repo` hinzu.

```
aws codeartifact associate-external-connection \
  --repository test-repo \
  --domain test-domain \
```

```
--external-connection public:npmjs
```

Ausgabe:

```
{
  "repository": {
    "name": "test-repo",
    "administratorAccount": "111122223333",
    "domainName": "test-domain",
    "domainOwner": "111122223333",
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/test-repo",
    "upstreams": [],
    "externalConnections": [
      {
        "externalConnectionName": "public:npmjs",
        "packageFormat": "npm",
        "status": "AVAILABLE"
      }
    ]
  }
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Hinzufügen einer externen Verbindung](#).AWS CodeArtifact

- API-Einheiten finden Sie [AssociateExternalConnection](#) in der AWS CLI Befehlsreferenz.

copy-package-versions

Das folgende Codebeispiel zeigt die Verwendung `copy-package-versions`.

AWS CLI

Um Paketversionen von einem Repository in ein anderes zu kopieren

Im Folgenden werden die Versionen 4.0.0 und 5.0.0 eines Pakets namens `test-package` von `my-repo` nach `test-repo copy-package-versions` verschoben.

```
aws codeartifact copy-package-versions \
  --domain test-domain \
  --source-repository my-repo \
```

```
--destination-repository test-repo \  
--format npm \  
--package test-package \  
--versions '["4.0.0", "5.0.0"]'
```

Ausgabe:

```
{  
  "format": "npm",  
  "package": "test-package",  
  "versions": [  
    {  
      "version": "5.0.0",  
      "revision": "REVISION-1-SAMPLE-6C81EFF7DA55CC",  
      "status": "Published"  
    },  
    {  
      "version": "4.0.0",  
      "revision": "REVISION-2-SAMPLE-55C752BEE772FC",  
      "status": "Published"  
    }  
  ]  
}
```

[Weitere Informationen finden Sie im Benutzerhandbuch unter Pakete zwischen Repositorys kopieren.AWS CodeArtifact](#)

- API-Einheiten finden Sie [CopyPackageVersions](#) in der AWS CLI Befehlsreferenz.

create-domain

Das folgende Codebeispiel zeigt die Verwendung `create-domain`.

AWS CLI

Um eine Domain zu erstellen

Im folgenden `create-domain` Beispiel wird eine Domäne mit dem Namen `test-domain` erstellt.

```
aws codeartifact create-domain \  
  --domain test-domain
```

Ausgabe:

```
{
  "domain": {
    "name": "test-domain",
    "owner": "111122223333",
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",
    "status": "Active",
    "createdTime": "2020-10-20T13:16:48.559000-04:00",
    "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
    "repositoryCount": 0,
    "assetSizeBytes": 0
  }
}
```

Weitere Informationen finden Sie unter [Create a domain](#) im AWS CodeArtifact Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDomain](#) in der AWS CLI Befehlsreferenz.

create-repository

Das folgende Codebeispiel zeigt die Verwendung `create-repository`.

AWS CLI

So erstellen Sie ein Repository

Das folgende `create-repository` Beispiel erstellt ein Repository mit dem Namen `test-repo` innerhalb einer Domäne namens `test-domain`.

```
aws codeartifact create-repository \
  --domain test-domain \
  --domain-owner 111122223333 \
  --repository test-repo \
  --description "This is a test repository."
```

Ausgabe:

```
{
  "repository": {
    "name": "test-repo",
```

```

    "administratorAccount": "111122223333",
    "domainName": "test-domain",
    "domainOwner": "111122223333",
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
    "description": "This is a test repository.",
    "upstreams": [],
    "externalConnections": []
  }
}

```

Weitere Informationen finden Sie unter [Create a domain](#) im Benutzerhandbuch.AWS CodeArtifact

- APIEinzelheiten finden Sie [CreateRepository](#) in der AWS CLI Befehlsreferenz.

delete-domain-permissions-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-domain-permissions-policy`.

AWS CLI

Um das Dokument mit den Berechtigungsrichtlinien aus einer Domäne zu löschen

Im folgenden `delete-domain-permissions-policy` Beispiel wird die Berechtigungsrichtlinie aus einer Domäne mit dem Namen `test-domain` gelöscht.

```

aws codeartifact delete-domain-permissions-policy \
  --domain test-domain

```

Ausgabe:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BasicDomainPolicy",
      "Action": [
        "codeartifact:GetDomainPermissionsPolicy",
        "codeartifact:ListRepositoriesInDomain",
        "codeartifact:GetAuthorizationToken",
        "codeartifact:CreateRepository"
      ],
    },
  ],
}

```



```
        "Effect": "Allow",
        "Resource": "*",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
        }
    ]
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Löschen einer Domänenrichtlinie](#).AWS CodeArtifact

- APIEinzelheiten finden Sie [DeleteDomainPermissionsPolicy](#) in der AWS CLI Befehlsreferenz.

delete-domain

Das folgende Codebeispiel zeigt die Verwendung `delete-domain`.

AWS CLI

Um eine Domain zu löschen

Im folgenden `delete-domain` Beispiel wird eine Domäne mit dem Namen `test-domain` gelöscht.

```
aws codeartifact delete-domain \
  --domain test-domain
```

Ausgabe:

```
{
  "domain": {
    "name": "test-domain",
    "owner": "417498243647",
    "arn": "arn:aws:codeartifact:us-west-2:417498243647:domain/test-domain",
    "status": "Deleted",
    "createdTime": "2020-10-20T13:16:48.559000-04:00",
    "encryptionKey": "arn:aws:kms:us-west-2:417498243647:key/c9fe2447-0795-4fda-afbe-8464574ae162",
    "repositoryCount": 0,
    "assetSizeBytes": 0
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Löschen einer Domäne](#) im AWS CodeArtifact Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDomain](#) in der AWS CLI Befehlsreferenz.

delete-package-versions

Das folgende Codebeispiel zeigt die Verwendung `delete-package-versions`.

AWS CLI

Um Paketversionen zu löschen

Im folgenden `delete-package-versions` Beispiel wird Version 4.0.0 eines Pakets namens `test-package` gelöscht.

```
aws codeartifact delete-package-versions \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --versions 4.0.0
```

Ausgabe:

```
{
  "successfulVersions": {
    "4.0.0": {
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
      "status": "Deleted"
    }
  },
  "failedVersions": {}
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Löschen einer Paketversion](#).AWS CodeArtifact

- API-Einheiten finden Sie [DeletePackageVersions](#) in der AWS CLI Befehlsreferenz.

delete-repository-permissions-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-repository-permissions-policy`.

AWS CLI

Um eine Berechtigungsrichtlinie aus einem Repository zu löschen

Im folgenden `delete-repository-permissions-policy` Beispiel wird die Berechtigungsrichtlinie aus einem Repository mit dem Namen `test-repo` gelöscht.

```
aws codeartifact delete-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo
```

Ausgabe:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::111122223333:root"  
      },  
      "Action": [  
        "codeartifact:DescribePackageVersion",  
        "codeartifact:DescribeRepository",  
        "codeartifact:GetPackageVersionReadme",  
        "codeartifact:GetRepositoryEndpoint",  
        "codeartifact:ListPackages",  
        "codeartifact:ListPackageVersions",  
        "codeartifact:ListPackageVersionAssets",  
        "codeartifact:ListPackageVersionDependencies",  
        "codeartifact:ReadFromRepository"  
      ],  
      "Resource": "*"   
    }  
  ]  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Löschen einer Richtlinie](#).AWS CodeArtifact

- API-Einheiten finden Sie [DeleteRepositoryPermissionsPolicy](#) in der AWS CLI Befehlsreferenz.

delete-repository

Das folgende Codebeispiel zeigt die Verwendung `delete-repository`.

AWS CLI

So löschen Sie ein Repository

Das folgende `delete-repository` Beispiel löscht ein Repository, das `test-repo` in einer Domäne namens `test-domain` ist.

```
aws codeartifact delete-repository \  
  --domain test-domain \  
  --repository test-repo
```

Ausgabe:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "description": "This is a test repository",  
    "upstreams": [],  
    "externalConnections": []  
  }  
}
```

Weitere Informationen finden Sie im AWS CodeArtifact Benutzerhandbuch unter [Löschen eines Repositorys](#).

- API-Einheiten finden Sie [DeleteRepository](#) in der AWS CLI Befehlsreferenz.

describe-domain

Das folgende Codebeispiel zeigt die Verwendung `describe-domain`.

AWS CLI

Um Informationen über eine Domain zu erhalten

Im folgenden `describe-domain` Beispiel wird ein `DomainDescription` Objekt für eine Domäne mit dem Namen `test-domain` zurückgegeben.

```
aws codeartifact describe-domain \  
  --domain test-domain
```

Ausgabe:

```
{  
  "domain": {  
    "name": "test-domain",  
    "owner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",  
    "status": "Active",  
    "createdTime": "2020-10-20T13:16:48.559000-04:00",  
    "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",  
    "repositoryCount": 2,  
    "assetSizeBytes": 0,  
    "s3BucketArn": "arn:aws:s3:::assets-111122223333-us-west-2"  
  }  
}
```

Weitere Informationen finden Sie unter [Domänenübersicht](#) im AWS CodeArtifact Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDomain](#) in der AWS CLI Befehlsreferenz.

describe-repository

Das folgende Codebeispiel zeigt die Verwendung `describe-repository`.

AWS CLI

Um Informationen über ein Repository zu erhalten

Das folgende `describe-repository` Beispiel gibt ein `RepositoryDescription` Objekt für ein Repository mit dem Namen `test-repo` zurück.

```
aws codeartifact describe-repository \  
  --domain test-domain \  
  --repository test-repo
```

Ausgabe:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "description": "This is a test repository.",  
    "upstreams": [],  
    "externalConnections": []  
  }  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Create a domain](#).AWS CodeArtifact

- API-Einheiten finden Sie [DescribeRepository](#) in der AWS CLI Befehlsreferenz.

disassociate-external-connection

Das folgende Codebeispiel zeigt die Verwendung `disassociate-external-connection`.

AWS CLI

Um eine externe Verbindung aus einem Repository zu entfernen

Im folgenden `disassociate-external-connection` Beispiel wird eine externe Verbindung zu `npmjs.com` aus einem Repository namens `test-repo` entfernt.

```
aws codeartifact disassociate-external-connection \  
  --repository test-repo \  
  --domain test-domain \  
  --external-connection public:npmjs
```

Ausgabe:

```
{
  "repository": {
    "name": "test-repo",
    "administratorAccount": "111122223333",
    "domainName": "test-domain",
    "domainOwner": "111122223333",
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
    "upstreams": [],
    "externalConnections": []
  }
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Entfernen einer externen Verbindung](#).AWS CodeArtifact

- API-Einheiten finden Sie [DisassociateExternalConnection](#) in der AWS CLI Befehlsreferenz.

dispose-package-versions

Das folgende Codebeispiel zeigt die Verwendung `dispose-package-versions`.

AWS CLI

Um die Assets einer Paketversion zu löschen und ihren Status auf Disposed zu setzen

Im folgenden `dispose-package-versions` Beispiel werden die Elemente der Testpaket-Version 4.0.0 gelöscht und ihr Status auf Disposed gesetzt.

```
aws codeartifact dispose-package-versions \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --versions 4.0.0
```

Ausgabe:

```
{
  "successfulVersions": {
    "4.0.0": {
```

```
        "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
        "status": "Disposed"
    }
},
"failedVersions": {}
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Arbeiten mit CodeArtifact Paketen](#).AWS CodeArtifact

- APIEinzelheiten finden Sie [DisposePackageVersions](#) in der AWS CLI Befehlsreferenz.

get-authorization-token

Das folgende Codebeispiel zeigt die Verwendung `get-authorization-token`.

AWS CLI

Um ein Autorisierungstoken zu erhalten

Im folgenden `get-authorization-token` Beispiel wird ein CodeArtifact Autorisierungstoken abgerufen.

```
aws codeartifact get-authorization-token \
  --domain test-domain \
  --query authorizationToken \
  --output text
```

Ausgabe:

```
This command will return the authorization token. You can store the output in an
environment variable when calling the command.
```

Weitere Informationen finden [Sie unter Konfiguration von pip ohne den Login-Befehl](#) im AWS CodeArtifact Benutzerhandbuch.

- APIEinzelheiten finden Sie [GetAuthorizationToken](#) in der AWS CLI Befehlsreferenz.

get-domain-permissions-policy

Das folgende Codebeispiel zeigt die Verwendung `get-domain-permissions-policy`.

AWS CLI

Um das Dokument mit den Berechtigungsrichtlinien für eine Domain abzurufen

Im folgenden `get-domain-permissions-policy` Beispiel wird die Berechtigungsrichtlinie an eine Domäne mit dem Namen `test-domain` angehängt.

```
aws codeartifact get-domain-permissions-policy \  
  --domain test-domain
```

Ausgabe:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "BasicDomainPolicy",  
      "Action": [  
        "codeartifact:GetDomainPermissionsPolicy",  
        "codeartifact:ListRepositoriesInDomain",  
        "codeartifact:GetAuthorizationToken",  
        "codeartifact:CreateRepository"  
      ],  
      "Effect": "Allow",  
      "Resource": "*",  
      "Principal": {  
        "AWS": "arn:aws:iam::111122223333:root"  
      }  
    }  
  ]  
}
```

Weitere Informationen finden [Sie unter Lesen einer Domänenrichtlinie](#) im AWS CodeArtifact Benutzerhandbuch.

- API-Einheiten finden Sie [GetDomainPermissionsPolicy](#) in der AWS CLI Befehlsreferenz.

get-package-version-asset

Das folgende Codebeispiel zeigt die Verwendung `get-package-version-asset`.

AWS CLI

Um ein Asset aus einer Paketversion abzurufen

Im folgenden `get-package-version-asset` Beispiel wird das `package.tgz` Asset für Version 4.0.0 eines npm-Pakets namens `test-package` abgerufen.

```
aws codeartifact get-package-version-asset \  
  --domain test-domain \  
  --repository test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0 \  
  --asset 'package.tgz' \  
  outfileName
```

Ausgabe:

The output for this command will also store the raw asset in the file provided in place of `outfileName`.

```
{  
  "assetName": "package.tgz",  
  "packageVersion": "4.0.0",  
  "packageVersionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Paketversionselemente auflisten](#).AWS CodeArtifact

- API-Einheiten finden Sie [GetPackageVersionAsset](#) in der AWS CLI Befehlsreferenz.

`get-package-version-readme`

Das folgende Codebeispiel zeigt die Verwendung `get-package-version-readme`.

AWS CLI

Um die Readme-Datei einer Paketversion abzurufen

Im folgenden `get-package-version-readme` Beispiel wird die Readme-Datei für Version 4.0.0 eines npm-Pakets namens `test-package` abgerufen.

```
aws codeartifact get-package-version-readme \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

Ausgabe:

```
{  
  "format": "npm",  
  "package": "test-package",  
  "version": "4.0.0",  
  "readme": "<div align=\"center\">\n  <a href=\"https://github.com/test-package/  
testpack\"> ... more content ... \n",  
  "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="  
}
```

Weitere Informationen finden Sie in der [Readme-Datei zur Paketversion anzeigen](#) im Benutzerhandbuch.AWS CodeArtifact

- APIEinzelheiten finden Sie [GetPackageVersionReadme](#) in der AWS CLI Befehlsreferenz.

get-repository-endpoint

Das folgende Codebeispiel zeigt die Verwendung get-repository-endpoint.

AWS CLI

Um den URL Endpunkt eines Repositorys abzurufen

Das folgende get-repository-endpoint Beispiel gibt den NPM-Endpunkt für das Test-Repository zurück.

```
aws codeartifact get-repository-endpoint \  
  --domain test-domain \  
  --repository test-repo \  
  --format npm
```

Ausgabe:

```
{
  "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-
west-2.amazonaws.com/npm/test-repo/"
}
```

Weitere Informationen finden Sie unter [Connect zu einem Repository](#) herstellen im AWS CodeArtifact Benutzerhandbuch.

- API-Einheiten finden Sie [GetRepositoryEndpoint](#) in der AWS CLI Befehlsreferenz.

get-repository-permissions-policy

Das folgende Codebeispiel zeigt die Verwendung `get-repository-permissions-policy`.

AWS CLI

Um das Dokument mit den Berechtigungsrichtlinien für ein Repository abzurufen

Im folgenden `get-repository-permissions-policy` Beispiel wird die Berechtigungsrichtlinie an ein Repository mit dem Namen `test-repo` angehängt.

```
aws codeartifact get-repository-permissions-policy \
  --domain test-domain \
  --repository test-repo
```

Ausgabe:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "codeartifact:DescribePackageVersion",
        "codeartifact:DescribeRepository",
        "codeartifact:GetPackageVersionReadme",
        "codeartifact:GetRepositoryEndpoint",
        "codeartifact:ListPackages",
        "codeartifact:ListPackageVersions",

```

```
        "codeartifact:ListPackageVersionAssets",
        "codeartifact:ListPackageVersionDependencies",
        "codeartifact:ReadFromRepository"
    ],
    "Resource": "*"
}
]
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Eine Richtlinie lesen](#).AWS CodeArtifact

- API-Einheiten finden Sie [GetRepositoryPermissionsPolicy](#) in der AWS CLI Befehlsreferenz.

list-domains

Das folgende Codebeispiel zeigt die Verwendung `list-domains`.

AWS CLI

Um Domains aufzulisten

Im folgenden `list-domains` Beispiel wird eine Zusammenfassung aller Domänen zurückgegeben, die dem AWS Konto gehören, das den Anruf tätigt.

```
aws codeartifact list-domains
```

Ausgabe:

```
{
  "domains": [
    {
      "name": "my-domain",
      "owner": "111122223333",
      "status": "Active",
      "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    {
      "name": "test-domain",
      "owner": "111122223333",
      "status": "Active",

```

```

        "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    }
]
}

```

Weitere Informationen finden Sie [CodeArtifactim AWS CodeArtifact Benutzerhandbuch unter Arbeiten mit Domänen](#).

- APIEinzelheiten finden Sie [ListDomains](#)in der AWS CLI Befehlsreferenz.

list-package-version-assets

Das folgende Codebeispiel zeigt die Verwendung `list-package-version-assets`.

AWS CLI

Um die Ressourcen einer Paketversion anzuzeigen

Im folgenden `list-package-version-assets` Beispiel werden die Assets für Version 4.0.0 eines npm-Pakets namens `test-package` abgerufen.

```

aws codeartifact list-package-version-assets \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --package-version 4.0.0

```

Ausgabe:

```

{
  "format": "npm",
  "package": "test-package",
  "version": "4.0.0",
  "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
  "assets": [
    {
      "name": "package.tgz",
      "size": 316680,
      "hashes": {
        "MD5": "60078ec6d9e76b89fb55c860832742b2",

```

```

        "SHA-1": "b44a9b6297bcb698f1c51a3545a2b3b368d59c52",
        "SHA-256":
"d2aa8c6afc3c8591765785a37d1c5acae482a8eb3ab9729ed28922692454f2e2",
        "SHA-512":
"3e585d15c8a594e20d7de57b362ea81754c011acb2641a19f1b72c8531ea39825896bab344ae616a0a5a824cb9
    }
  }
]
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Paketversionselemente auflisten](#).AWS CodeArtifact

- APIEinzelheiten finden Sie [ListPackageVersionAssets](#)in der AWS CLI Befehlsreferenz.

list-package-version-dependencies

Das folgende Codebeispiel zeigt die Verwendung `list-package-version-dependencies`.

AWS CLI

Um die Abhängigkeiten einer Paketversion anzuzeigen

Im folgenden `list-package-version-dependencies` Beispiel werden die Abhängigkeiten für Version 4.0.0 eines npm-Pakets namens `test-package` abgerufen.

```

aws codeartifact list-package-version-dependencies \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --package-version 4.0.0

```

Ausgabe:

```

{
  "format": "npm",
  "package": "test-package",
  "version": "4.0.0",
  "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
  "dependencies": [
    {
      "namespace": "testns",

```

```

        "package": "testdep1",
        "dependencyType": "regular",
        "versionRequirement": "1.8.5"
    },
    {
        "namespace": "testns",
        "package": "testdep2",
        "dependencyType": "regular",
        "versionRequirement": "1.8.5"
    }
]
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Paketversionsdetails und Abhängigkeiten anzeigen und aktualisieren](#).AWS CodeArtifact

- APIEinzelheiten finden Sie [ListPackageVersionDependencies](#)in der AWS CLI Befehlsreferenz.

list-package-versions

Das folgende Codebeispiel zeigt die Verwendung `list-package-versions`.

AWS CLI

Um Paketversionen für ein Paket aufzulisten

Das folgende `list-package-versions` Beispiel gibt eine Liste von Paketversionen für ein Paket mit dem Namen `zurückkind-of`.

```

aws codeartifact list-package-versions \
  --package kind-of \
  --domain test-domain \
  --repository test-repo \
  --format npm

```

Ausgabe:

```

{
  "defaultDisplayVersion": "1.0.1",
  "format": "npm",
  "package": "kind-of",
  "versions": [

```



```
{
  "version": "1.0.1",
  "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC",
  "status": "Published"
},
{
  "version": "1.0.0",
  "revision": "REVISION-SAMPLE-2-C752BEEF6D2CFC",
  "status": "Published"
},
{
  "version": "0.1.2",
  "revision": "REVISION-SAMPLE-3-654S65A5C5E1FC",
  "status": "Published"
},
{
  "version": "0.1.1",
  "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC",
  "status": "Published"
},
{
  "version": "0.1.0",
  "revision": "REVISION-SAMPLE-4-AF669139B772FC",
  "status": "Published"
}
]
```

Weitere Informationen finden Sie im AWS CodeArtifact Benutzerhandbuch unter [Paketversionen auflisten](#).

- API-Einheiten finden Sie [ListPackageVersions](#) in der AWS CLI Befehlsreferenz.

list-packages

Das folgende Codebeispiel zeigt die Verwendung `list-packages`.

AWS CLI

Um Pakete in einem Repository aufzulisten

Das folgende `list-packages` Beispiel listet Pakete in einem Repository auf, das `test-repo` in einer Domäne namens `test-domain` ist.

```
aws codeartifact list-packages \  
  --domain test-domain \  
  --repository test-repo
```

Ausgabe:

```
{  
  "packages": [  
    {  
      "format": "npm",  
      "package": "lodash"  
    },  
    {  
      "format": "python",  
      "package": "test-package"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im AWS CodeArtifact Benutzerhandbuch unter [Paketnamen auflisten](#).

- API-Einheiten finden Sie [ListPackages](#) in der AWS CLI Befehlsreferenz.

list-repositories-in-domain

Das folgende Codebeispiel zeigt die Verwendung `list-repositories-in-domain`.

AWS CLI

Um Repositories in einer Domain aufzulisten

Das folgende `list-repositories-in-domain` Beispiel gibt eine Zusammenfassung aller Repositories in der Testdomänenendomäne zurück.

```
aws codeartifact list-repositories-in-domain \  
  --domain test-domain
```

Ausgabe:

```
{
```

```
"repositories": [  
  {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-  
domain/test-repo",  
    "description": "This is a test repository."  
  },  
  {  
    "name": "test-repo2",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-  
domain/test-repo2",  
    "description": "This is a test repository."  
  }  
]  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Repositories auflisten](#).AWS CodeArtifact

- API-Einheiten finden Sie [ListRepositoriesInDomain](#) in der AWS CLI Befehlsreferenz.

list-repositories

Das folgende Codebeispiel zeigt die Verwendung `list-repositories`.

AWS CLI

Um Repositories aufzulisten

Das folgende `list-repositories` Beispiel gibt eine Zusammenfassung aller Repositories in der Domäne zurück, die dem AWS Konto gehört, das den Anruf tätigt.

```
aws codeartifact list-repositories
```

Ausgabe:

```
{
```

```
"repositories": [  
  {  
    "name": "npm-store",  
    "administratorAccount": "111122223333",  
    "domainName": "my-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-  
domain/npm-store",  
    "description": "Provides npm artifacts from npm, Inc."  
  },  
  {  
    "name": "target-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "my-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-  
domain/target-repo",  
    "description": "test target repo"  
  },  
  {  
    "name": "test-repo2",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-  
domain/test-repo2",  
    "description": "This is a test repository."  
  }  
]  
}
```

Weitere Informationen finden Sie im AWS CodeArtifact Benutzerhandbuch unter [Repositories auflisten](#).

- API-Einheiten finden Sie [ListRepositories](#) in der AWS CLI Befehlsreferenz.

login

Das folgende Codebeispiel zeigt die Verwendung `login`.

AWS CLI

Um die Authentifizierung für Ihr Repository mit dem `login`-Befehl zu konfigurieren

Im folgenden Login Beispiel wird der npm-Paketmanager mit einem Repository namens test-repo in einer Domain namens test-domain konfiguriert.

```
aws codeartifact login \  
  --domain test-domain \  
  --repository test-repo \  
  --tool npm
```

Ausgabe:

```
Successfully configured npm to use AWS CodeArtifact repository https://test-  
domain-111122223333.d.codeartifact.us-west-2.amazonaws.com/npm/test-repo/  
Login expires in 12 hours at 2020-11-12 01:53:16-05:00
```

Weitere Informationen finden Sie unter [Erste Schritte mit](#) dem im Benutzerhandbuch. AWS CLI
AWS CodeArtifact

- API-Einheiten finden Sie unter [Login](#) in der AWS CLI Befehlsreferenz.

put-domain-permissions-policy

Das folgende Codebeispiel zeigt die Verwendung put-domain-permissions-policy.

AWS CLI

Um eine Berechtigungsrichtlinie an eine Domain anzuhängen

Im folgenden put-domain-permissions-policy Beispiel wird eine Berechtigungsrichtlinie, die in der Datei policy.json definiert ist, an eine Domain mit dem Namen test-domain angehängt.

```
aws codeartifact put-domain-permissions-policy \  
  --domain test-domain \  
  --policy-document file://PATH/TO/policy.json
```

Ausgabe:

```
{  
  "policy": {  
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-  
domain",
```

```

    "document": "{ ...policy document content...}",
    "revision": "MQ1yyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxx="
  }
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Eine Domänenrichtlinie einrichten](#).AWS CodeArtifact

- API-Einheiten finden Sie [PutDomainPermissionsPolicy](#) in der AWS CLI Befehlsreferenz.

put-repository-permissions-policy

Das folgende Codebeispiel zeigt die Verwendung `put-repository-permissions-policy`.

AWS CLI

Um eine Berechtigungsrichtlinie an ein Repository anzuhängen

Im folgenden `put-repository-permissions-policy` Beispiel wird eine Berechtigungsrichtlinie, die in der Datei `policy.json` definiert ist, an ein Repository mit dem Namen `test-repo` angehängt.

```

aws codeartifact put-repository-permissions-policy \
  --domain test-domain \
  --repository test-repo \
  --policy-document file://PATH/TO/policy.json

```

Ausgabe:

```

{
  "policy": {
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-domain/test-repo",
    "document": "{ ...policy document content...}",
    "revision": "MQ1yyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxx="
  }
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Eine Richtlinie einrichten](#).AWS CodeArtifact

- API-Einheiten finden Sie [PutRepositoryPermissionsPolicy](#) in der AWS CLI Befehlsreferenz.

update-package-versions-status

Das folgende Codebeispiel zeigt die Verwendung `update-package-versions-status`.

AWS CLI

Um den Status der Paketversion zu aktualisieren

Im folgenden `update-package-versions-status` Beispiel wird der Status von Version 4.0.0 des Testpaket-Pakets auf `Archived` aktualisiert.

```
aws codeartifact update-package-versions-status \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0 \  
  --target-status Archived
```

Ausgabe:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Archived"  
    }  
  },  
  "failedVersions": {}  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Versionsstatus des Aktualisierungspakets](#).AWS CodeArtifact

- API-Einheiten finden Sie [UpdatePackageVersionsStatus](#) in der AWS CLI Befehlsreferenz.

update-repository

Das folgende Codebeispiel zeigt die Verwendung `update-repository`.

AWS CLI

Um ein Repository zu aktualisieren

Im folgenden `update-repository` Beispiel wird die Beschreibung eines Repos mit dem Namen `test-repo` in einer Domäne namens `test-domain` auf „Dies ist eine aktualisierte Beschreibung“ aktualisiert.

```
aws codeartifact update-repository \  
  --domain test-domain \  
  --repository test-repo \  
  --description "this is an updated description"
```

Ausgabe:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "description": "this is an updated description",  
    "upstreams": [],  
    "externalConnections": []  
  }  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Eine Repository-Konfiguration anzeigen oder ändern](#).AWS CodeArtifact

- API-Einheiten finden Sie [UpdateRepository](#) in der AWS CLI Befehlsreferenz.

CodeBuild Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CodeBuild.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-delete-builds

Das folgende Codebeispiel zeigt die Verwendung `batch-delete-builds`.

AWS CLI

Um Builds zu löschen AWS CodeBuild.

Im folgenden `batch-delete-builds` Beispiel werden Builds CodeBuild mit den angegebenen IDs Werten gelöscht.

```
aws codebuild batch-delete-builds --ids my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE
```

Ausgabe:

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE"
  ]
}
```

Weitere Informationen finden Sie unter [Delete Builds \(AWS CLI\)](#) im AWS CodeBuild Benutzerhandbuch.

- API-Einheiten finden Sie [BatchDeleteBuilds](#) in der AWS CLI Befehlsreferenz.

batch-get-build-batches

Das folgende Codebeispiel zeigt die Verwendung `batch-get-build-batches`.

AWS CLI

Um Details von Builds anzuzeigen AWS CodeBuild.

Im folgenden `batch-get-build-batches` Beispiel werden Informationen zu Build-Batches CodeBuild mit den angegebenen IDs Werten abgerufen.

```
aws codebuild batch-get-build-batches \
  --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```

Ausgabe:

```
{
  "buildBatches": [
    {
      "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
      "arn": "arn:aws:codebuild:us-west-2:123456789012:build-batch/codebuild-
demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
      "startTime": "2020-11-03T21:52:20.775000+00:00",
      "endTime": "2020-11-03T21:56:59.784000+00:00",
      "currentPhase": "SUCCEEDED",
      "buildBatchStatus": "SUCCEEDED",
      "resolvedSourceVersion": "0a6546f68309560d08a310daac92314c4d378f6b",
      "projectName": "codebuild-demo-project",
      "phases": [
        {
          "phaseType": "SUBMITTED",
          "phaseStatus": "SUCCEEDED",
          "startTime": "2020-11-03T21:52:20.775000+00:00",
          "endTime": "2020-11-03T21:52:20.976000+00:00",
          "durationInSeconds": 0
        },
        {
          "phaseType": "DOWNLOAD_BATCHSPEC",
          "phaseStatus": "SUCCEEDED",
          "startTime": "2020-11-03T21:52:20.976000+00:00",
          "endTime": "2020-11-03T21:52:57.401000+00:00",
          "durationInSeconds": 36
        }
      ]
    }
  ]
}
```

```
    {
      "phaseType": "IN_PROGRESS",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2020-11-03T21:52:57.401000+00:00",
      "endTime": "2020-11-03T21:56:59.751000+00:00",
      "durationInSeconds": 242
    },
    {
      "phaseType": "COMBINE_ARTIFACTS",
      "phaseStatus": "SUCCEEDED",
      "startTime": "2020-11-03T21:56:59.751000+00:00",
      "endTime": "2020-11-03T21:56:59.784000+00:00",
      "durationInSeconds": 0
    },
    {
      "phaseType": "SUCCEEDED",
      "startTime": "2020-11-03T21:56:59.784000+00:00"
    }
  ],
  "source": {
    "type": "GITHUB",
    "location": "https://github.com/my-repo/codebuild-demo-project.git",
    "gitCloneDepth": 1,
    "gitSubmodulesConfig": {
      "fetchSubmodules": false
    },
    "reportBuildStatus": false,
    "insecureSsl": false
  },
  "secondarySources": [],
  "secondarySourceVersions": [],
  "artifacts": {
    "location": ""
  },
  "secondaryArtifacts": [],
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
```

```
    "imagePullCredentialsType": "CODEBUILD"
  },
  "logConfig": {
    "cloudWatchLogs": {
      "status": "ENABLED"
    },
    "s3Logs": {
      "status": "DISABLED",
      "encryptionDisabled": false
    }
  },
  "buildTimeoutInMinutes": 60,
  "queuedTimeoutInMinutes": 480,
  "complete": true,
  "initiator": "Strohm",
  "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
  "buildBatchNumber": 6,
  "buildBatchConfig": {
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/
codebuild-demo-project",
    "restrictions": {
      "maximumBuildsAllowed": 100
    },
    "timeoutInMins": 480
  },
  "buildGroups": [
    {
      "identifier": "DOWNLOAD_SOURCE",
      "ignoreFailure": false,
      "currentBuildSummary": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:379737d8-bc35-48ec-97fd-776d27545315",
        "requestedOn": "2020-11-03T21:52:21.394000+00:00",
        "buildStatus": "SUCCEEDED",
        "primaryArtifact": {
          "type": "no_artifacts",
          "identifier": "DOWNLOAD_SOURCE"
        },
        "secondaryArtifacts": []
      },
      "secondaryArtifacts": []
    }
  ],
  {
    "identifier": "linux_small",
    "dependsOn": [],
```

```
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:dd785171-ed84-4bb6-8ede-ceeb86e54bdb",
            "requestedOn": "2020-11-03T21:52:57.604000+00:00",
            "buildStatus": "SUCCEEDED",
            "primaryArtifact": {
                "type": "no_artifacts",
                "identifier": "linux_small"
            },
            "secondaryArtifacts": []
        },
    },
    {
        "identifier": "linux_medium",
        "dependsOn": [
            "linux_small"
        ],
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:97cf7bd4-5313-4786-8243-4aef350a1267",
            "requestedOn": "2020-11-03T21:54:18.474000+00:00",
            "buildStatus": "SUCCEEDED",
            "primaryArtifact": {
                "type": "no_artifacts",
                "identifier": "linux_medium"
            },
            "secondaryArtifacts": []
        },
    },
    {
        "identifier": "linux_large",
        "dependsOn": [
            "linux_medium"
        ],
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:60a194cd-0d03-4337-9db1-d41476a17d27",
            "requestedOn": "2020-11-03T21:55:39.203000+00:00",
            "buildStatus": "SUCCEEDED",
            "primaryArtifact": {
                "type": "no_artifacts",
```

```

        "identifier": "linux_large"
      },
      "secondaryArtifacts": []
    }
  ]
}
],
"buildBatchesNotFound": []
}

```

Weitere Informationen finden Sie unter Batch-Builds in AWS CodeBuild < <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html> >) __ im Benutzerhandbuch.AWS CodeBuild

- APIEinzelheiten finden Sie in der Befehlsreferenz [BatchGetBuildBatches](#).AWS CLI

batch-get-builds

Das folgende Codebeispiel zeigt die Verwendung batch-get-builds.

AWS CLI

Um Details von Builds anzuzeigen AWS CodeBuild.

Im folgenden batch-get-builds Beispiel werden Informationen zu Builds CodeBuild mit den angegebenen Werten abgerufenIDs.

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

Ausgabe:

```

{
  "buildsNotFound": [],
  "builds": [
    {
      "artifacts": {
        "md5sum": "0e95edf915048a0c22efe6d139fff837",
        "location": "arn:aws:s3:::codepipeline-us-west-2-820783811474/CodeBuild-Python-Pip/BuildArtif/6DJsqQa",
        "encryptionDisabled": false,

```

```
    "sha256sum":
      "cfa0df33a090966a737f64ae4fe498969fdc842a0c9aec540bf93c37ac0d05a2"
    },
    "logs": {
      "cloudWatchLogs": {
        "status": "ENABLED"
      },
      "s3Logs": {
        "status": "DISABLED"
      },
      "streamName": "46472baf-8f6b-43c2-9255-b3b963af2732",
      "groupName": "/aws/codebuild/codebuild-demo-project",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=46472baf-8f6b-43c2-9255-b3b963af2732"
    },
    "timeoutInMinutes": 60,
    "environment": {
      "privilegedMode": false,
      "computeType": "BUILD_GENERAL1_MEDIUM",
      "image": "aws/codebuild/windows-base:1.0",
      "environmentVariables": [],
      "type": "WINDOWS_CONTAINER"
    },
    "projectName": "codebuild-demo-project",
    "buildComplete": true,
    "source": {
      "gitCloneDepth": 1,
      "insecureSsl": false,
      "type": "CODEPIPELINE"
    },
    "buildStatus": "SUCCEEDED",
    "secondaryArtifacts": [],
    "phases": [
      {
        "durationInSeconds": 0,
        "startTime": 1548717462.122,
        "phaseType": "SUBMITTED",
        "endTime": 1548717462.484,
        "phaseStatus": "SUCCEEDED"
      },
      {
        "durationInSeconds": 0,
        "startTime": 1548717462.484,
```

```
    "phaseType": "QUEUED",
    "endTime": 1548717462.775,
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 34,
    "endTime": 1548717496.909,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717462.775,
    "phaseType": "PROVISIONING",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 15,
    "endTime": 1548717512.555,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717496.909,
    "phaseType": "DOWNLOAD_SOURCE",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 0,
    "endTime": 1548717512.734,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717512.555,
    "phaseType": "INSTALL",
    "phaseStatus": "SUCCEEDED"
  },
  {
```



```
    "durationInSeconds": 0,
    "endTime": 1548717512.924,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717512.734,
    "phaseType": "PRE_BUILD",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 9,
    "endTime": 1548717522.254,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717512.924,
    "phaseType": "BUILD",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 3,
    "endTime": 1548717525.498,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717522.254,
    "phaseType": "POST_BUILD",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 9,
    "endTime": 1548717534.646,
    "contexts": [
      {
        "statusCode": "",
```

```

        "message": ""
      }
    ],
    "startTime": 1548717525.498,
    "phaseType": "UPLOAD_ARTIFACTS",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 2,
    "endTime": 1548717536.846,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717534.646,
    "phaseType": "FINALIZING",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "startTime": 1548717536.846,
    "phaseType": "COMPLETED"
  }
],
"startTime": 1548717462.122,
"encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
"initiator": "codepipeline/CodeBuild-Pipeline",
"secondarySources": [],
"serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",
"currentPhase": "COMPLETED",
"id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
"cache": {
  "type": "NO_CACHE"
},
"sourceVersion": "arn:aws:s3:::codepipeline-us-west-2-820783811474/CodeBuild-Python-Pip/SourceArti/1TspnN3.zip",
"endTime": 1548717536.846,
"arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
"queuedTimeoutInMinutes": 480,
"resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
},

```

```
{
  "artifacts": {
    "md5sum": "",
    "overrideArtifactName": false,
    "location": "arn:aws:s3:::my-artifacts/codebuild-demo-project",
    "encryptionDisabled": false,
    "sha256sum": ""
  },
  "logs": {
    "cloudWatchLogs": {
      "status": "ENABLED"
    },
    "s3Logs": {
      "status": "DISABLED"
    },
    "streamName": "4dea3ca4-20ec-4898-b22a-a9eb9292775d",
    "groupName": "/aws/codebuild/codebuild-demo-project",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=4dea3ca4-20ec-4898-b22a-a9eb9292775d"
  },
  "timeoutInMinutes": 60,
  "environment": {
    "privilegedMode": false,
    "computeType": "BUILD_GENERAL1_MEDIUM",
    "image": "aws/codebuild/windows-base:1.0",
    "environmentVariables": [],
    "type": "WINDOWS_CONTAINER"
  },
  "projectName": "codebuild-demo-project",
  "buildComplete": true,
  "source": {
    "gitCloneDepth": 1,
    "location": "https://github.com/my-repo/codebuild-demo-project.git",
    "insecureSsl": false,
    "reportBuildStatus": false,
    "type": "GITHUB"
  },
  "buildStatus": "SUCCEEDED",
  "secondaryArtifacts": [],
  "phases": [
    {
      "durationInSeconds": 0,
      "startTime": 1548716241.89,

```

```
    "phaseType": "SUBMITTED",
    "endTime": 1548716242.241,
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 0,
    "startTime": 1548716242.241,
    "phaseType": "QUEUED",
    "endTime": 1548716242.536,
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 33,
    "endTime": 1548716276.171,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548716242.536,
    "phaseType": "PROVISIONING",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 15,
    "endTime": 1548716291.809,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548716276.171,
    "phaseType": "DOWNLOAD_SOURCE",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 0,
    "endTime": 1548716291.993,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ]
  }
]
```

```
    }
  ],
  "startTime": 1548716291.809,
  "phaseType": "INSTALL",
  "phaseStatus": "SUCCEEDED"
},
{
  "durationInSeconds": 0,
  "endTime": 1548716292.191,
  "contexts": [
    {
      "statusCode": "",
      "message": ""
    }
  ],
  "startTime": 1548716291.993,
  "phaseType": "PRE_BUILD",
  "phaseStatus": "SUCCEEDED"
},
{
  "durationInSeconds": 9,
  "endTime": 1548716301.622,
  "contexts": [
    {
      "statusCode": "",
      "message": ""
    }
  ],
  "startTime": 1548716292.191,
  "phaseType": "BUILD",
  "phaseStatus": "SUCCEEDED"
},
{
  "durationInSeconds": 3,
  "endTime": 1548716304.783,
  "contexts": [
    {
      "statusCode": "",
      "message": ""
    }
  ],
  "startTime": 1548716301.622,
  "phaseType": "POST_BUILD",
  "phaseStatus": "SUCCEEDED"
}
```

```
    },
    {
      "durationInSeconds": 8,
      "endTime": 1548716313.775,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716304.783,
      "phaseType": "UPLOAD_ARTIFACTS",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 2,
      "endTime": 1548716315.935,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716313.775,
      "phaseType": "FINALIZING",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "startTime": 1548716315.935,
      "phaseType": "COMPLETED"
    }
  ],
  "startTime": 1548716241.89,
  "secondarySourceVersions": [],
  "initiator": "my-codebuild-project",
  "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
  "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
  "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
  "currentPhase": "COMPLETED",
  "id": "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
  "cache": {
    "type": "NO_CACHE"
  }
}
```

```
    },
    "endTime": 1548716315.935,
    "secondarySources": [],
    "queuedTimeoutInMinutes": 480,
    "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
  }
]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Build-Details anzeigen \(AWS CLI\)](#).

- API-Einheiten finden Sie [BatchGetBuilds](#) in der AWS CLI Befehlsreferenz.

batch-get-projects

Das folgende Codebeispiel zeigt die Verwendung `batch-get-projects`.

AWS CLI

Um eine Liste von AWS CodeBuild Build-Projektnamen zu erhalten.

Im folgenden `batch-get-projects` Beispiel wird eine namentlich angegebene Liste von CodeBuild Build-Projekten abgerufen.

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2 my-other-demo-project
```

In der folgenden Ausgabe listet das `projectsNotFound` Array alle Build-Projektnamen auf, die angegeben, aber nicht gefunden wurden. Das Array `projects` listet Details für jedes Build-Projekt auf, für das Informationen gefunden wurden.

```
{
  "projectsNotFound": [],
  "projects": [
    {
      "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
      "name": "codebuild-demo-project2",
      "queuedTimeoutInMinutes": 480,
      "timeoutInMinutes": 60,
      "source": {
```

```

        "buildspec": "version: 0.2\n\n#env:\n #variables:\n # key:\n\n\"value\"\n # key: \"value\"\n #parameter-store:\n # key: \"value\"\n\n# key: \"value\"\n\n#phases:\n #install:\n #commands:\n # - command\n\n# - command\n #pre_build:\n #commands:\n # - command\n # - command\n\n build:\n #commands:\n # - command\n # - command\n\n#post_build:\n\n#commands:\n # - command\n # - command\n\n#artifacts:\n #files:\n #\n - location\n # - location\n #name: $(date +%Y-%m-%d)\n #discard-paths: yes\n\n#base-directory: location\n#cache:\n #paths:\n # - paths",
        "type": "NO_SOURCE",
        "insecureSsl": false,
        "gitCloneDepth": 1
    },
    "artifacts": {
        "type": "NO_ARTIFACTS"
    },
    "badge": {
        "badgeEnabled": false
    },
    "lastModified": 1540588091.108,
    "created": 1540588091.108,
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/test-for-
sample",
    "secondarySources": [],
    "secondaryArtifacts": [],
    "cache": {
        "type": "NO_CACHE"
    },
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-test-
role",
    "environment": {
        "image": "aws/codebuild/java:openjdk-8",
        "privilegedMode": true,
        "type": "LINUX_CONTAINER",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": []
    },
    "tags": []
},
{
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "name": "my-other-demo-project",
    "queuedTimeoutInMinutes": 480,
    "timeoutInMinutes": 60,
    "source": {

```



```
        "location": "https://github.com/iversonic/codedeploy-sample.git",
        "reportBuildStatus": false,
        "buildspec": "buildspec.yml",
        "insecureSsl": false,
        "gitCloneDepth": 1,
        "type": "GITHUB",
        "auth": {
            "type": "OAUTH"
        }
    },
    "artifacts": {
        "type": "NO_ARTIFACTS"
    },
    "badge": {
        "badgeEnabled": false
    },
    "lastModified": 1523401711.73,
    "created": 1523401711.73,
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/Project2",
    "cache": {
        "type": "NO_CACHE"
    },
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/codebuild-Project2-service-role",
    "environment": {
        "image": "aws/codebuild/nodejs:4.4.7",
        "privilegedMode": false,
        "type": "LINUX_CONTAINER",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": []
    },
    "tags": []
}
]
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Details eines Build-Projekts anzeigen \(AWS CLI\)](#).

- API-Einheiten finden Sie [BatchGetProjects](#) in der AWS CLI Befehlsreferenz.

batch-get-report-groups

Das folgende Codebeispiel zeigt die Verwendung `batch-get-report-groups`.

AWS CLI

Um Informationen zu einer oder mehreren Berichtsgruppen in zu erhalten AWS CodeBuild.

Im folgenden `batch-get-report-groups` Beispiel werden Informationen über die Berichtsgruppe mit den angegebenen ARN Werten abgerufen.

```
aws codebuild batch-get-report-groups \
  --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>
```

Ausgabe:

```
{
  "reportGroups": [
    {
      "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-
group-name>",
      "name": "report-group-name",
      "type": "TEST",
      "exportConfig": {
        "exportConfigType": "NO_EXPORT"
      },
      "created": "2020-10-01T18:04:08.466000+00:00",
      "lastModified": "2020-10-01T18:04:08.466000+00:00",
      "tags": []
    }
  ],
  "reportGroupsNotFound": []
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [BatchGetReportGroups](#) in der AWS CLI Befehlsreferenz.

batch-get-reports

Das folgende Codebeispiel zeigt die Verwendung `batch-get-reports`.

AWS CLI

Um Informationen zu einem oder mehreren Berichten in zu erhalten AWS CodeBuild.

Im folgenden `batch-get-reports` Beispiel werden Informationen zu den Berichten mit den angegebenen ARNs Werten abgerufen.

```
aws codebuild batch-get-reports \  
  --report-arns arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report 1 ID> arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report 2 ID>
```

Ausgabe:

```
{  
  "reports": [  
    {  
      "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report 1 ID>",  
      "type": "TEST",  
      "name": "<report-group-name>",  
      "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>",  
      "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-reports:<ID>",  
      "status": "FAILED",  
      "created": "2020-10-01T11:25:22.531000-07:00",  
      "expired": "2020-10-31T11:25:22-07:00",  
      "exportConfig": {  
        "exportConfigType": "NO_EXPORT"  
      },  
      "truncated": false,  
      "testSummary": {  
        "total": 28,  
        "statusCounts": {  
          "ERROR": 5,  
          "FAILED": 1,  
          "SKIPPED": 4,  
          "SUCCEEDED": 18,  
          "UNKNOWN": 0  
        },  
        "durationInNanoSeconds": 94000000  
      },  
    }  
  ]  
}
```

```

    },
    {
      "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 2 ID>",
      "type": "TEST",
      "name": "<report-group-name>",
      "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>",
      "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-
reports:<ID>",
      "status": "FAILED",
      "created": "2020-10-01T11:13:05.816000-07:00",
      "expired": "2020-10-31T11:13:05-07:00",
      "exportConfig": {
        "exportConfigType": "NO_EXPORT"
      },
      "truncated": false,
      "testSummary": {
        "total": 28,
        "statusCounts": {
          "ERROR": 5,
          "FAILED": 1,
          "SKIPPED": 4,
          "SUCCEEDED": 18,
          "UNKNOWN": 0
        },
        "durationInNanoSeconds": 94000000
      },
    }
  ],
  "reportsNotFound": []
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichten](#).

- API-Einheiten finden Sie [BatchGetReports](#) in der AWS CLI Befehlsreferenz.

create-project

Das folgende Codebeispiel zeigt die Verwendung `create-project`.

AWS CLI

Beispiel 1: Um ein AWS CodeBuild Build-Projekt zu erstellen

Im folgenden create-project Beispiel wird ein CodeBuild Build-Projekt mit Quelldateien aus einem S3-Bucket erstellt

```
aws codebuild create-project \
  --name "my-demo-project" \
  --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-  
input-bucket/my-source.zip\"}" \
  --artifacts {"\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-  
output-bucket\""} \
  --environment {"\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/  
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
  --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role"
```

Ausgabe:

```
{
  "project": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
    "name": "my-cli-demo-project",
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role",
    "lastModified": 1556839783.274,
    "badge": {
      "badgeEnabled": false
    },
    "queuedTimeoutInMinutes": 480,
    "environment": {
      "image": "aws/codebuild/standard:1.0",
      "computeType": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "imagePullCredentialsType": "CODEBUILD",
      "privilegedMode": false,
      "environmentVariables": []
    },
    "artifacts": {
      "location": "codebuild-us-west-2-123456789012-output-bucket",
      "name": "my-cli-demo-project",
```

```

        "namespaceType": "NONE",
        "type": "S3",
        "packaging": "NONE",
        "encryptionDisabled": false
    },
    "source": {
        "type": "S3",
        "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
        "insecureSsl": false
    },
    "timeoutInMinutes": 60,
    "cache": {
        "type": "NO_CACHE"
    },
    "created": 1556839783.274
}
}

```

Beispiel 2: Um ein AWS CodeBuild Build-Projekt mit einer JSON Eingabedatei für die Parameter zu erstellen

Im folgenden `create-project` Beispiel wird ein CodeBuild Build-Projekt erstellt, indem alle erforderlichen Parameter in einer JSON Eingabedatei übergeben werden. Erstellen Sie die Eingabedateivorlage, indem Sie den Befehl nur mit dem `ausführen--generate-cli-skeleton` parameter.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Die JSON Eingabedatei `create-project.json` enthält den folgenden Inhalt:

```

{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
  },
  "environment": {

```

```

    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:1.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "serviceIAMRole"
}

```

Ausgabe:

```

{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:1.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
    },
    "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
    "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
  }
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Erstellen eines Build-Projekts \(AWS CLI\)](#).

- API-Einheiten finden Sie [CreateProject](#) in der AWS CLI Befehlsreferenz.

create-report-group

Das folgende Codebeispiel zeigt die Verwendung `create-report-group`.

AWS CLI

Um eine Berichtsgruppe in zu erstellen AWS CodeBuild.

Im folgenden `create-report-group` Beispiel wird eine neue Berichtsgruppe erstellt.

```
aws codebuild create-report-group \  
  --cli-input-json file://create-report-group-source.json
```

Inhalt `create-report-group-source` von `json`:

```
{  
  "name": "cli-created-report-group",  
  "type": "TEST",  
  "exportConfig": {  
    "exportConfigType": "S3",  
    "s3Destination": {  
      "bucket": "my-s3-bucket",  
      "path": "",  
      "packaging": "ZIP",  
      "encryptionDisabled": true  
    }  
  }  
}
```

Ausgabe:

```
{  
  "reportGroup": {  
    "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-group",  
    "name": "cli-created-report-group",  
    "type": "TEST",  
    "exportConfig": {  
      "exportConfigType": "S3",
```



```

        "s3Destination": {
            "bucket": "my-s3-bucket",
            "path": "",
            "packaging": "ZIP",
            "encryptionDisabled": true
        }
    },
    "created": 1602020026.775,
    "lastModified": 1602020026.775
}
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [CreateReportGroup](#) in der AWS CLI Befehlsreferenz.

create-webhook

Das folgende Codebeispiel zeigt die Verwendung `create-webhook`.

AWS CLI

Um Webhook-Filter für ein AWS CodeBuild Projekt zu erstellen

Im folgenden `create-webhook` Beispiel wird ein Webhook für ein CodeBuild Projekt mit dem Namen `erstelltmy-project`, das zwei Filtergruppen hat. Die erste Filtergruppe gibt Pull-Anfragen an, die in Verzweigungen mit Git-Referenznamen, die dem regulären Ausdruck `^refs/heads/master$` entsprechen, und mit Kopfreferenzen, die `^refs/heads/myBranch$` entsprechen, erstellt, aktualisiert oder erneut geöffnet werden. Die zweite Filtergruppe spezifiziert Push-Anfragen für Branches mit Git-Referenznamen, die nicht dem regulären Ausdruck `^refs/heads/myBranch$` entsprechen.

```

aws codebuild create-webhook \
  --project-name my-project \
  --filter-groups "[[{"type": "EVENT", "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED"}, {"type": "HEAD_REF", "pattern": "^refs/heads/myBranch$", "excludeMatchedPattern": true}, {"type": "BASE_REF", "pattern": "^refs/heads/master$", "excludeMatchedPattern": true}], [{"type": "EVENT", "pattern": "PUSH"}, {"type": "HEAD_REF", "pattern": "^refs/heads/myBranch$", "excludeMatchedPattern": true}]]"

```

Ausgabe:

```

{
  "webhook": {
    "payloadUrl": "https://codebuild.us-west-2.amazonaws.com/webhooks?
t=eyJlbnNyeXB0ZWREYXRhIjoiVVl5MGtoeGRwSzZFRXl2Wnh4bld1Z0tKZ291TVpQNEtFamQ3RDlDYWpRaGIreVFrdm
    "url": "https://api.github.com/repos/iversonic/codedeploy-sample/
hooks/105190656",
    "lastModifiedSecret": 1556311319.069,
    "filterGroups": [
      [
        {
          "type": "EVENT",
          "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED",
          "excludeMatchedPattern": false
        },
        {
          "type": "HEAD_REF",
          "pattern": "refs/heads/myBranch$",
          "excludeMatchedPattern": true
        },
        {
          "type": "BASE_REF",
          "pattern": "refs/heads/master$",
          "excludeMatchedPattern": true
        }
      ],
      [
        {
          "type": "EVENT",
          "pattern": "PUSH",
          "excludeMatchedPattern": false
        },
        {
          "type": "HEAD_REF",
          "pattern": "refs/heads/myBranch$",
          "excludeMatchedPattern": true
        }
      ]
    ]
  }
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [GitHub Webhook-Ereignisse filtern \(SDK\)](#).

- API-Einheiten finden Sie [CreateWebhook](#) in der AWS CLI Befehlsreferenz.

delete-build-batch

Das folgende Codebeispiel zeigt die Verwendung `delete-build-batch`.

AWS CLI

Um ein Batch-Build in zu löschen AWS CodeBuild.

Im folgenden `delete-build-batch` Beispiel wird der angegebene Batch-Build gelöscht.

```
aws codebuild delete-build-batch \
  --id <project-name>:<batch-ID>
```

Ausgabe:

```
{
  "statusCode": "BATCH_DELETED",
  "buildsDeleted": [
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>"
  ],
  "buildsNotDeleted": []
}
```

Weitere Informationen finden Sie [im AWS CodeBuild Benutzerhandbuch unter Batch-Builds AWS CodeBuild](#) in.

- API-Einheiten finden Sie [DeleteBuildBatch](#) in der AWS CLI Befehlsreferenz.

delete-project

Das folgende Codebeispiel zeigt die Verwendung `delete-project`.

AWS CLI

Um ein AWS CodeBuild Build-Projekt zu löschen

Im folgenden `delete-project` Beispiel wird das angegebene CodeBuild Build-Projekt gelöscht.

```
aws codebuild delete-project --name my-project
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Löschen eines Build-Projekts \(AWS CLI\)](#).

- API-Einheiten finden Sie [DeleteProject](#) in der AWS CLI Befehlsreferenz.

`delete-report-group`

Das folgende Codebeispiel zeigt die Verwendung `delete-report-group`.

AWS CLI

Um einen Bericht zu löschen, gruppiert sich in AWS CodeBuild.

Im folgenden `delete-report-group` Beispiel wird die Berichtsgruppe mit den angegebenen ARN Werten gelöscht.

```
aws codebuild delete-report-group \  
  --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [DeleteReportGroup](#) in der AWS CLI Befehlsreferenz.

`delete-report`

Das folgende Codebeispiel zeigt die Verwendung `delete-report`.

AWS CLI

Um einen Bericht in zu löschen AWS CodeBuild.

Im folgenden `delete-report` Beispiel wird der angegebene Bericht gelöscht.

```
aws codebuild delete-report \  
  --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-  
name>:<report-ID>
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichten](#).

- API-Einheiten finden Sie [DeleteReport](#) in der AWS CLI Befehlsreferenz.

delete-source-credentials

Das folgende Codebeispiel zeigt die Verwendung `delete-source-credentials`.

AWS CLI

Um die Verbindung zu einem Quellanbieter zu trennen und dessen Zugriffstoken zu entfernen.

Im folgenden `delete-source-credentials` Beispiel wird die Verbindung zu einem Quellanbieter getrennt und dessen Token entfernt. Die Anmeldeinformationen ARN der Quelle, die für die Verbindung mit dem Quellanbieter verwendet werden, bestimmen, welche Quellanmeldedaten verwendet werden.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Ausgabe:

```
{  
  "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"  
}
```

Weitere Informationen finden Sie unter [Connect von Quellanbietern mit Zugriffstoken \(CLI\)](#) im AWS CodeBuild Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSourceCredentials](#) in der AWS CLI Befehlsreferenz.

delete-webhook

Das folgende Codebeispiel zeigt die Verwendung `delete-webhook`.

AWS CLI

Um einen Webhook-Filter aus einem AWS CodeBuild Projekt zu löschen

Im folgenden `delete-webhook` Beispiel wird ein Webhook aus dem angegebenen Projekt gelöscht. CodeBuild

```
aws codebuild delete-webhook --project-name my-project
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Automatisches Ausführen von Builds beenden \(AWS CLI\)](#).

- API-Einheiten finden Sie [DeleteWebhook](#) in der AWS CLI Befehlsreferenz.

describe-code-coverages

Das folgende Codebeispiel zeigt die Verwendung `describe-code-coverages`.

AWS CLI

Detaillierte Informationen zu den Testergebnissen der Codeabdeckung finden Sie unter AWS CodeBuild.

Im folgenden `describe-code-coverages` Beispiel werden Informationen zu den Testergebnissen für die Codeabdeckung im angegebenen Bericht abgerufen.

```
aws codebuild describe-code-coverages \  
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Ausgabe:

```
{  
  "codeCoverages": [  
    {  
      "id": "20a0adcc-db13-4b66-804b-ecaf9f852855",  
      "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-group-name>:<report-ID>",  
    }  
  ]  
}
```

```

    "filePath": "<source-file-1-path>",
    "lineCoveragePercentage": 83.33,
    "linesCovered": 5,
    "linesMissed": 1,
    "branchCoveragePercentage": 50.0,
    "branchesCovered": 1,
    "branchesMissed": 1,
    "expired": "2020-11-20T21:22:45+00:00"
  },
  {
    "id": "0887162d-bf57-4cf1-a164-e432373d1a83",
    "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-
group-name>:<report-ID>",
    "filePath": "<source-file-2-path>",
    "lineCoveragePercentage": 90.9,
    "linesCovered": 10,
    "linesMissed": 1,
    "branchCoveragePercentage": 50.0,
    "branchesCovered": 1,
    "branchesMissed": 1,
    "expired": "2020-11-20T21:22:45+00:00"
  }
]
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Berichte zur Codeabdeckung](#).

- API-Einheiten finden Sie [DescribeCodeCoverages](#) in der AWS CLI Befehlsreferenz.

describe-test-cases

Das folgende Codebeispiel zeigt die Verwendung `describe-test-cases`.

AWS CLI

Um detaillierte Informationen zu Testfällen zu erhalten, finden Sie in AWS CodeBuild.

Im folgenden `describe-test-cases` Beispiel werden Informationen zu den Testfällen im angegebenen Bericht abgerufen.

```
aws codebuild describe-test-cases \
```

```
--report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Ausgabe:

```
{
  "testCases": [
    {
      "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>",
      "testRawDataPath": "<test-report-path>",
      "prefix": "NUnit.Tests.Assemblies.MockTestFixture",
      "name": "NUnit.Tests.Assemblies.MockTestFixture.NotRunnableTest",
      "status": "ERROR",
      "durationInNanoSeconds": 0,
      "message": "No arguments were provided\n",
      "expired": "2020-11-20T17:52:10+00:00"
    },
    {
      "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>",
      "testRawDataPath": "<test-report-path>",
      "prefix": "NUnit.Tests.Assemblies.MockTestFixture",
      "name": "NUnit.Tests.Assemblies.MockTestFixture.TestWithException",
      "status": "ERROR",
      "durationInNanoSeconds": 0,
      "message": "System.ApplicationException : Intentional Exception
\nat NUnit.Tests.Assemblies.MockTestFixture.MethodThrowsException()\nat
NUnit.Tests.Assemblies.MockTestFixture.TestWithException()\n\n",
      "expired": "2020-11-20T17:52:10+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie [AWS CodeBuild im AWS CodeBuild Benutzerhandbuch unter Arbeiten mit Testberichten](#).

- API-Einheiten finden Sie [DescribeTestCases](#) in der AWS CLI Befehlsreferenz.

import-source-credentials

Das folgende Codebeispiel zeigt die Verwendung `import-source-credentials`.

AWS CLI

Connect einen AWS CodeBuild Benutzer mit einem Quellanbieter, indem Sie Anmeldeinformationen für den Quellanbieter importieren.

Im folgenden `import-source-credentials` Beispiel wird ein Token für ein Bitbucket-Repository importiert, das `BASIC_AUTH` als Authentifizierungstyp verwendet.

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

Ausgabe:

```
{
  "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"
}
```

Weitere Informationen finden Sie unter [Connect von Quellanbietern mit Zugriffstoken \(CLI\)](#) im AWS CodeBuild Benutzerhandbuch.

- API-Einheiten finden Sie [ImportSourceCredentials](#) in der AWS CLI Befehlsreferenz.

invalidate-project-cache

Das folgende Codebeispiel zeigt die Verwendung `invalidate-project-cache`.

AWS CLI

Um den Cache für ein AWS CodeBuild Build-Projekt zurückzusetzen.

Im folgenden `invalidate-project-cache` Beispiel wird der Cache für das angegebene CodeBuild Projekt zurückgesetzt.

```
aws codebuild invalidate-project-cache --project-name my-project
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Build Caching CodeBuild im AWS](#) CodeBuild Benutzerhandbuch.

- API-Einheiten finden Sie [InvalidateProjectCache](#) in der AWS CLI Befehlsreferenz.

list-build-batches-for-project

Das folgende Codebeispiel zeigt die Verwendung `list-build-batches-for-project`.

AWS CLI

Um Batch-Builds für ein bestimmtes Build-Projekt in aufzulisten AWS CodeBuild.

Das folgende `list-build-batches-for-project` Beispiel listet die CodeBuild Batch-Builds für das angegebene Projekt auf.

```
aws codebuild list-build-batches-for-project \  
  --project-name "<project-name>"
```

Ausgabe:

```
{  
  "ids": [  
    "<project-name>:<batch-ID>",  
    "<project-name>:<batch-ID>"  
  ]  
}
```

Weitere Informationen finden Sie [im AWS CodeBuild Benutzerhandbuch unter Batch-Builds AWS CodeBuild](#) in.

- API-Einheiten finden Sie [ListBuildBatchesForProject](#) in der AWS CLI Befehlsreferenz.

list-build-batches

Das folgende Codebeispiel zeigt die Verwendung `list-build-batches`.

AWS CLI

Um Batch-Builds aufzulisten AWS CodeBuild.

Das folgende `list-build-batches` Beispiel listet die CodeBuild Batch-Builds für das Girokonto auf.

```
aws codebuild list-build-batches
```

Ausgabe:

```
{
  "ids": [
    "<project-name>:<batch-ID>",
    "<project-name>:<batch-ID>"
  ]
}
```

Weitere Informationen finden Sie unter Batch-Builds in AWS CodeBuild < <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html> > __ im Benutzerhandbuch.AWS CodeBuild

- APIEinzelheiten finden Sie in der Befehlsreferenz [ListBuildBatches](#).AWS CLI

list-builds-for-project

Das folgende Codebeispiel zeigt die Verwendung `list-builds-for-project`.

AWS CLI

Um eine Liste von Builds für ein AWS CodeBuild Build-Projekt anzuzeigen.

Im folgenden `list-builds-for-project` Beispiel wird der Build für das angegebene CodeBuild Build-Projekt IDs in absteigender Reihenfolge aufgeführt.

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order DESCENDING
```

Ausgabe:

```
{
  "ids": [
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-11111example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-22222example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-33333example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-44444example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-55555example"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Anzeigen einer Build-Liste IDs für ein Build-Projekt \(AWS CLI\)](#)

- API-Einheiten finden Sie [ListBuildsForProject](#) in der AWS CLI Befehlsreferenz.

list-builds

Das folgende Codebeispiel zeigt die Verwendung `list-builds`.

AWS CLI

Um eine Liste von AWS CodeBuild Builds zu erhalten IDs.

Im folgenden `list-builds` Beispiel wird eine in aufsteigender Reihenfolge CodeBuild IDs sortierte Liste abgerufen.

```
aws codebuild list-builds --sort-order ASCENDING
```

Die Ausgabe enthält einen `nextToken` Wert, der angibt, dass mehr Ausgaben verfügbar sind.

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for
  brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Führen Sie diesen Befehl erneut aus und geben Sie den `nextToken` Wert in der vorherigen Antwort als Parameter an, um den nächsten Teil der Ausgabe abzurufen. Wiederholen Sie den Vorgang, bis Sie in der Antwort keinen `nextToken` Wert mehr erhalten.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Nächster Teil der Ausgabe:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
```

```

    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Eine Liste von Build IDs \(AWS CLI\) anzeigen](#)

- API-Einheiten finden Sie [ListBuilds](#) in der AWS CLI Befehlsreferenz.

list-curated-environment-images

Das folgende Codebeispiel zeigt die Verwendung `list-curated-environment-images`.

AWS CLI

Um eine Liste der von Ihnen verwalteten Docker-Images zu erhalten AWS CodeBuild, die Sie für Ihre Builds verwenden können.

Das folgende `list-curated-environment-images` Beispiel listet die Docker-Images auf, die von verwaltet werden und CodeBuild die für Builds verwendet werden können. :

```
aws codebuild list-curated-environment-images
```

Ausgabe:

```

{
  "platforms": [
    {
      "platform": "AMAZON_LINUX",
      "languages": [
        {
          "language": "JAVA",
          "images": [
            {
              "description": "AWS ElasticBeanstalk - Java 7 Running on
Amazon Linux 64bit v2.1.3",
              "name": "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3",
              "versions": [
                "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3-1.0.0"
              ]
            }
          ],
        }
      ],
    }
  ],
}

```

```

        {
            "description": "AWS ElasticBeanstalk - Java 8 Running on
Amazon Linux 64bit v2.1.3",
            "name": "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3",
            "versions": [
                "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3-1.0.0"
            ]
        },
        ... LIST TRUNCATED FOR BREVITY ...
    ]
}
]
}
]
}
}

```

Weitere Informationen finden Sie CodeBuild im AWS CodeBuild Benutzerhandbuch unter [Docker-Images bereitgestellt von](#)

- APIEinzelheiten finden Sie [ListCuratedEnvironmentImages](#) in der AWS CLI Befehlsreferenz.

list-projects

Das folgende Codebeispiel zeigt die Verwendung `list-projects`.

AWS CLI

Um eine Liste von AWS CodeBuild Build-Projektnamen zu erhalten.

Im folgenden `list-projects` Beispiel wird eine Liste von CodeBuild Build-Projekten abgerufen, die in aufsteigender Reihenfolge nach Namen sortiert sind.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Die Ausgabe enthält einen `nextToken` Wert, der angibt, dass mehr Ausgaben verfügbar sind.

```

{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U
+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
  ]
}

```

```

        ... The full list of build project names has been omitted for
brevity ...
        "codebuild-demo-project99"
    ]
}

```

Führen Sie diesen Befehl erneut aus und geben Sie den `nextToken` Wert aus der vorherigen Antwort als Parameter an, um den nächsten Teil der Ausgabe abzurufen. Wiederholen Sie den Vorgang, bis Sie in der Antwort keinen `nextToken` Wert mehr erhalten.

```

aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
  "projects": [
    "codebuild-demo-project100",
    "codebuild-demo-project101",

    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"
  ]
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Eine Liste mit Namen von Build-Projekten anzeigen \(AWS CLI\)](#).

- API-Einheiten finden Sie [ListProjects](#) in der AWS CLI Befehlsreferenz.

list-report-groups

Das folgende Codebeispiel zeigt die Verwendung `list-report-groups`.

AWS CLI

Um eine Liste der Berichtsgruppe abzurufen, ARNs in AWS CodeBuild.

Im folgenden `list-report-groups` Beispiel wird die Berichtsgruppe ARNs für das Konto in der Region abgerufen.

```
aws codebuild list-report-groups
```

Ausgabe:

```
{
  "reportGroups": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [ListReportGroups](#) in der AWS CLI Befehlsreferenz.

list-reports-for-report-group

Das folgende Codebeispiel zeigt die Verwendung `list-reports-for-report-group`.

AWS CLI

Um eine Liste der Berichte in einer Berichtsgruppe in abzurufen AWS CodeBuild.

Im folgenden `list-report-for-report-groups` Beispiel werden die Berichte in der angegebenen Berichtsgruppe für das Konto in der Region abgerufen.

```
aws codebuild list-reports-for-report-group \
  --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-
  group-name>
```

Ausgabe:

```
{
  "reports": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-1",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-2",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-3"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [ListReportsForReportGroup](#) in der AWS CLI Befehlsreferenz.

list-reports

Das folgende Codebeispiel zeigt die Verwendung `list-reports`.

AWS CLI

Um eine Liste der Berichte für das Girokonto in abzurufen AWS CodeBuild.

Im folgenden `list-reports` Beispiel werden die ARNs Berichte für das Girokonto abgerufen.

```
aws codebuild list-reports
```

Ausgabe:

```
{
  "reports": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report ID>",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report ID>",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report ID>"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichten](#).

- API-Einheiten finden Sie [ListReports](#) in der AWS CLI Befehlsreferenz.

list-shared-projects

Das folgende Codebeispiel zeigt die Verwendung `list-shared-projects`.

AWS CLI

Um das gemeinsam genutzte Projekt in aufzulisten AWS CodeBuild.

Im folgenden `list-shared-projects` Beispiel werden die CodeBuild gemeinsam genutzten Projekte aufgeführt, die für das aktuelle Konto verfügbar sind.

```
aws codebuild list-shared-projects
```

Ausgabe:

```
{
  "projects": [
    "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-
name-1>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-name-2>"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit gemeinsam genutzten Projekten](#).

- API-Einheiten finden Sie [ListSharedProjects](#) in der AWS CLI Befehlsreferenz.

list-shared-report-groups

Das folgende Codebeispiel zeigt die Verwendung `list-shared-report-groups`.

AWS CLI

Um eine Liste der gemeinsam genutzten Berichtsgruppe ARNs abzurufen AWS CodeBuild.

Im folgenden `list-shared-report-groups` Beispiel wird die Berichtsgruppe ARNs für das Konto in der Region abgerufen.

```
aws codebuild list-shared-report-groups
```

Ausgabe:

```
{
  "reportGroups": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [ListSharedReportGroups](#) in der AWS CLI Befehlsreferenz.

list-source-credentials

Das folgende Codebeispiel zeigt die Verwendung `list-source-credentials`.

AWS CLI

Um eine Liste von anzuzeigen `sourceCredentialsObjects`

Das folgende `list-source-credentials` Beispiel listet Token für ein AWS Konto auf, das mit einem Bitbucket-Konto und einem GitHub Konto verbunden ist. Jedes `sourceCredentialsInfos` Objekt in der Antwort enthält Informationen zu den verbundenen Quellenmeldedaten.

```
aws codebuild list-source-credentials
```

Ausgabe:

```
{
  "sourceCredentialsInfos": [
    {
      "serverType": "BITBUCKET",
      "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket",
      "authType": "BASIC_AUTH"
    },
    {
      "serverType": "GITHUB",
      "arn": "arn:aws:codebuild:us-west-2:123456789012:token/github",
      "authType": "OAUTH"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Connect von Quellenanbietern mit Zugriffstoken \(CLI\)](#) im AWS CodeBuild Benutzerhandbuch.

- API-Einheiten finden Sie [ListSourceCredentials](#) in der AWS CLI Befehlsreferenz.

retry-build-batch

Das folgende Codebeispiel zeigt die Verwendung `retry-build-batch`.

AWS CLI

Um einen fehlgeschlagenen Batch-Build in AWS CodeBuild zu wiederholen.

Im folgenden `retry-build-batch` Beispiel wird der angegebene Batch-Build neu gestartet.

```
aws codebuild retry-build-batch \  
  --id <project-name>:<batch-ID>
```

Ausgabe:

```
{  
  "buildBatch": {  
    "id": "<project-name>:<batch-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-  
name>:<batch-ID>",  
    "startTime": "2020-10-21T17:26:23.099000+00:00",  
    "currentPhase": "SUBMITTED",  
    "buildBatchStatus": "IN_PROGRESS",  
    "resolvedSourceVersion": "3a9e11cb419e8fff14b03883dc4e64f6155aaa7e",  
    "projectName": "<project-name>",  
    "phases": [  
      {  
        "phaseType": "SUBMITTED",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T17:26:23.099000+00:00",  
        "endTime": "2020-10-21T17:26:23.457000+00:00",  
        "durationInSeconds": 0  
      },  
      {  
        "phaseType": "DOWNLOAD_BATCHSPEC",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T17:26:23.457000+00:00",  
        "endTime": "2020-10-21T17:26:54.902000+00:00",  
        "durationInSeconds": 31  
      },  
      {  
        "phaseType": "IN_PROGRESS",  
        "phaseStatus": "CLIENT_ERROR",  
        "startTime": "2020-10-21T17:26:54.902000+00:00",
```

```

        "endTime": "2020-10-21T17:28:16.060000+00:00",
        "durationInSeconds": 81
    },
    {
        "phaseType": "FAILED",
        "phaseStatus": "RETRY",
        "startTime": "2020-10-21T17:28:16.060000+00:00",
        "endTime": "2020-10-21T17:29:39.709000+00:00",
        "durationInSeconds": 83
    },
    {
        "phaseType": "SUBMITTED",
        "startTime": "2020-10-21T17:29:39.709000+00:00"
    }
],
"source": {
    "type": "GITHUB",
    "location": "https://github.com/strohm-a/<project-name>-graph.git",
    "gitCloneDepth": 1,
    "gitSubmodulesConfig": {
        "fetchSubmodules": false
    },
    "reportBuildStatus": false,
    "insecureSsl": false
},
"secondarySources": [],
"secondarySourceVersions": [],
"artifacts": {
    "location": ""
},
"secondaryArtifacts": [],
"cache": {
    "type": "NO_CACHE"
},
"environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"logConfig": {
    "cloudWatchLogs": {

```

```

        "status": "ENABLED"
    },
    "s3Logs": {
        "status": "DISABLED",
        "encryptionDisabled": false
    }
},
"buildTimeoutInMinutes": 60,
"queuedTimeoutInMinutes": 480,
"complete": false,
"initiator": "<username>",
"encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
"buildBatchNumber": 4,
"buildBatchConfig": {
    "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
    "restrictions": {
        "maximumBuildsAllowed": 100
    },
    "timeoutInMins": 480
},
"buildGroups": [
    {
        "identifier": "DOWNLOAD_SOURCE",
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
            "requestedOn": "2020-10-21T17:26:23.889000+00:00",
            "buildStatus": "SUCCEEDED",
            "primaryArtifact": {
                "type": "no_artifacts",
                "identifier": "DOWNLOAD_SOURCE"
            },
            "secondaryArtifacts": []
        }
    },
    {
        "identifier": "linux_small",
        "dependsOn": [],
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",

```

```

        "requestedOn": "2020-10-21T17:26:55.115000+00:00",
        "buildStatus": "FAILED",
        "primaryArtifact": {
            "type": "no_artifacts",
            "identifier": "linux_small"
        },
        "secondaryArtifacts": []
    },
    {
        "identifier": "linux_medium",
        "dependsOn": [
            "linux_small"
        ],
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
            "requestedOn": "2020-10-21T17:26:54.594000+00:00",
            "buildStatus": "STOPPED"
        }
    },
    {
        "identifier": "linux_large",
        "dependsOn": [
            "linux_medium"
        ],
        "ignoreFailure": false,
        "currentBuildSummary": {
            "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
            "requestedOn": "2020-10-21T17:26:54.701000+00:00",
            "buildStatus": "STOPPED"
        }
    }
]
}
}

```

Weitere Informationen finden Sie [im AWS CodeBuild Benutzerhandbuch unter Batch-Builds AWS CodeBuild](#) in.

- API-Einheiten finden Sie [RetryBuildBatch](#) in der AWS CLI Befehlsreferenz.

retry-build

Das folgende Codebeispiel zeigt die Verwendung `retry-build`.

AWS CLI

Um ein fehlgeschlagenes Build-In AWS CodeBuild erneut zu versuchen.

Im folgenden `retry-build` Beispiel wird der angegebene Build neu gestartet.

```
aws codebuild retry-build \  
  --id <project-name>:<build-ID>
```

Ausgabe:

```
{  
  "build": {  
    "id": "<project-name>:<build-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-  
name>:<build-ID>",  
    "buildNumber": 9,  
    "startTime": "2020-10-21T17:51:38.161000+00:00",  
    "currentPhase": "QUEUED",  
    "buildStatus": "IN_PROGRESS",  
    "projectName": "<project-name>",  
    "phases": [  
      {  
        "phaseType": "SUBMITTED",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T17:51:38.161000+00:00",  
        "endTime": "2020-10-21T17:51:38.210000+00:00",  
        "durationInSeconds": 0  
      },  
      {  
        "phaseType": "QUEUED",  
        "startTime": "2020-10-21T17:51:38.210000+00:00"  
      }  
    ],  
    "source": {  
      "type": "GITHUB",  
      "location": "<GitHub-repo-URL>",  
      "gitCloneDepth": 1,  
      "gitSubmodulesConfig": {  
        "fetchSubmodules": false  
      }  
    }  
  }  
}
```



```
    },
    "reportBuildStatus": false,
    "insecureSsl": false
  },
  "secondarySources": [],
  "secondarySourceVersions": [],
  "artifacts": {
    "location": ""
  },
  "secondaryArtifacts": [],
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
  },
  "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-role-
name>",
  "logs": {
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=<region-ID>#logEvent:group=null;stream=null",
    "cloudWatchLogsArn": "arn:aws:logs:<region-ID>:<account-ID>:log-
group:null:log-stream:null",
    "cloudWatchLogs": {
      "status": "ENABLED"
    },
    "s3Logs": {
      "status": "DISABLED",
      "encryptionDisabled": false
    }
  },
  "timeoutInMinutes": 60,
  "queuedTimeoutInMinutes": 480,
  "buildComplete": false,
  "initiator": "<username>",
  "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3"
}
}
```

Weitere Informationen finden Sie [im AWS CodeBuild Benutzerhandbuch unter Batch-Builds AWS CodeBuild](#) in.

- API-Einheiten finden Sie [RetryBuild](#) in der AWS CLI Befehlsreferenz.

start-build-batch

Das folgende Codebeispiel zeigt die Verwendung `start-build-batch`.

AWS CLI

Um einen Batch-Build zu starten, verwenden Sie `aws codebuild start-build-batch`.

Im folgenden `start-build-batch` Beispiel wird ein Batch-Build des angegebenen Projekts gestartet.

```
aws codebuild start-build-batch \  
  --project-name <project-name>
```

Ausgabe:

```
{  
  "buildBatch": {  
    "id": "<project-name>:<batch-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-name>:<batch-ID>",  
    "startTime": "2020-10-21T16:54:24.740000+00:00",  
    "currentPhase": "SUBMITTED",  
    "buildBatchStatus": "IN_PROGRESS",  
    "projectName": "<project-name>",  
    "source": {  
      "type": "GITHUB",  
      "location": "<GitHub-repo-URL>",  
      "gitCloneDepth": 1,  
      "gitSubmodulesConfig": {  
        "fetchSubmodules": false  
      },  
      "reportBuildStatus": false,  
      "insecureSsl": false  
    },  
    "secondarySources": [],  
    "secondarySourceVersions": [],  
    "artifacts": {
```

```

    "location": ""
  },
  "secondaryArtifacts": [],
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
  },
  "logConfig": {
    "cloudWatchLogs": {
      "status": "ENABLED"
    },
    "s3Logs": {
      "status": "DISABLED",
      "encryptionDisabled": false
    }
  },
  "buildTimeoutInMinutes": 60,
  "queuedTimeoutInMinutes": 480,
  "complete": false,
  "initiator": "<username>",
  "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
  "buildBatchNumber": 3,
  "buildBatchConfig": {
    "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-
role-name>",
    "restrictions": {
      "maximumBuildsAllowed": 100
    },
    "timeoutInMins": 480
  }
}
}

```

Weitere Informationen finden Sie [im AWS CodeBuild Benutzerhandbuch unter Batch-Builds AWS CodeBuild](#) in.

- API-Einheiten finden Sie [StartBuildBatch](#) in der AWS CLI Befehlsreferenz.

start-build

Das folgende Codebeispiel zeigt die Verwendung `start-build`.

AWS CLI

Um mit der Ausführung eines AWS CodeBuild Build-Projekts zu beginnen.

Im folgenden `start-build` Beispiel wird ein Build für das angegebene CodeBuild Projekt gestartet. Der Build überschreibt sowohl die Projekteinstellung für die Anzahl der Minuten, für die der Build in die Warteschlange gestellt werden darf, bevor das Timeout eintritt, als auch die Artefakteinstellungen des Projekts.

```
aws codebuild start-build \  
  --project-name "my-demo-project" \  
  --queued-timeout-in-minutes-override 5 \  
  --artifacts-override {"type": "S3","location": "arn:aws:s3::artifacts-override","overrideArtifactName":true}
```

Ausgabe:

```
{  
  "build": {  
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",  
    "buildStatus": "IN_PROGRESS",  
    "buildComplete": false,  
    "projectName": "my-demo-project",  
    "timeoutInMinutes": 60,  
    "source": {  
      "insecureSsl": false,  
      "type": "S3",  
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-source.zip"  
    },  
    "queuedTimeoutInMinutes": 5,  
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",  
    "currentPhase": "QUEUED",  
    "startTime": 1556905683.568,  
    "environment": {  
      "computeType": "BUILD_GENERAL1_MEDIUM",  
      "environmentVariables": [],  
    }  
  }  
}
```

```
    "type": "LINUX_CONTAINER",
    "privilegedMode": false,
    "image": "aws/codebuild/standard:1.0",
    "imagePullCredentialsType": "CODEBUILD"
  },
  "phases": [
    {
      "phaseStatus": "SUCCEEDED",
      "startTime": 1556905683.568,
      "phaseType": "SUBMITTED",
      "durationInSeconds": 0,
      "endTime": 1556905684.524
    },
    {
      "startTime": 1556905684.524,
      "phaseType": "QUEUED"
    }
  ],
  "logs": {
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=null;stream=null"
  },
  "artifacts": {
    "encryptionDisabled": false,
    "location": "arn:aws:s3:::artifacts-override/my-demo-project",
    "overrideArtifactName": true
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
  "initiator": "my-aws-account-name",
  "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
}
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Run a Build \(AWS CLI\)](#).AWS CodeBuild

- API-Einheiten finden Sie [StartBuild](#) in der AWS CLI Befehlsreferenz.

stop-build-batch

Das folgende Codebeispiel zeigt die Verwendung stop-build-batch.

AWS CLI

Um einen laufenden Batch-Build in AWS CodeBuild Bearbeitung zu beenden.

Im folgenden stop-build-batch Beispiel wird der angegebene Batch-Build gestoppt.

```
aws codebuild stop-build-batch \  
  --id <project-name>:<batch-ID>
```

Ausgabe:

```
{  
  "buildBatch": {  
    "id": "<project-name>:<batch-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-name>:<batch-ID>",  
    "startTime": "2020-10-21T16:54:24.740000+00:00",  
    "endTime": "2020-10-21T16:56:05.152000+00:00",  
    "currentPhase": "STOPPED",  
    "buildBatchStatus": "STOPPED",  
    "resolvedSourceVersion": "aef7744ed069c51098e15c360f4102cd2cd1ad64",  
    "projectName": "<project-name>",  
    "phases": [  
      {  
        "phaseType": "SUBMITTED",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T16:54:24.740000+00:00",  
        "endTime": "2020-10-21T16:54:25.039000+00:00",  
        "durationInSeconds": 0  
      },  
      {  
        "phaseType": "DOWNLOAD_BATCHSPEC",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T16:54:25.039000+00:00",  
        "endTime": "2020-10-21T16:54:56.583000+00:00",  
        "durationInSeconds": 31  
      },  
      {  
        "phaseType": "IN_PROGRESS",  
        "phaseStatus": "STOPPED",
```

```
        "startTime": "2020-10-21T16:54:56.583000+00:00",
        "endTime": "2020-10-21T16:56:05.152000+00:00",
        "durationInSeconds": 68
    },
    {
        "phaseType": "STOPPED",
        "startTime": "2020-10-21T16:56:05.152000+00:00"
    }
],
"source": {
    "type": "GITHUB",
    "location": "<GitHub-repo-URL>",
    "gitCloneDepth": 1,
    "gitSubmodulesConfig": {
        "fetchSubmodules": false
    },
    "reportBuildStatus": false,
    "insecureSsl": false
},
"secondarySources": [],
"secondarySourceVersions": [],
"artifacts": {
    "location": ""
},
"secondaryArtifacts": [],
"cache": {
    "type": "NO_CACHE"
},
"environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"logConfig": {
    "cloudWatchLogs": {
        "status": "ENABLED"
    },
    "s3Logs": {
        "status": "DISABLED",
        "encryptionDisabled": false
    }
}
```

```

    },
    "buildTimeoutInMinutes": 60,
    "queuedTimeoutInMinutes": 480,
    "complete": true,
    "initiator": "Strohm",
    "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
    "buildBatchNumber": 3,
    "buildBatchConfig": {
      "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
      "restrictions": {
        "maximumBuildsAllowed": 100
      },
      "timeoutInMins": 480
    },
    "buildGroups": [
      {
        "identifier": "DOWNLOAD_SOURCE",
        "ignoreFailure": false,
        "currentBuildSummary": {
          "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
          "requestedOn": "2020-10-21T16:54:25.468000+00:00",
          "buildStatus": "SUCCEEDED",
          "primaryArtifact": {
            "type": "no_artifacts",
            "identifier": "DOWNLOAD_SOURCE"
          },
          "secondaryArtifacts": []
        }
      },
      {
        "identifier": "linux_small",
        "dependsOn": [],
        "ignoreFailure": false,
        "currentBuildSummary": {
          "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
          "requestedOn": "2020-10-21T16:54:56.833000+00:00",
          "buildStatus": "IN_PROGRESS"
        }
      },
      {
        "identifier": "linux_medium",

```



```

    "dependsOn": [
      "linux_small"
    ],
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
      "requestedOn": "2020-10-21T16:54:56.211000+00:00",
      "buildStatus": "PENDING"
    }
  },
  {
    "identifier": "linux_large",
    "dependsOn": [
      "linux_medium"
    ],
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
      "requestedOn": "2020-10-21T16:54:56.330000+00:00",
      "buildStatus": "PENDING"
    }
  }
]
}
}

```

Weitere Informationen finden Sie [im AWS CodeBuild Benutzerhandbuch unter Batch-Builds AWS CodeBuild](#) in.

- API-Einheiten finden Sie [StopBuildBatch](#) in der AWS CLI Befehlsreferenz.

stop-build

Das folgende Codebeispiel zeigt die Verwendung `stop-build`.

AWS CLI

Um einen Build eines AWS CodeBuild Build-Projekts zu beenden.

Das folgende `stop-build` Beispiel stoppt den angegebenen CodeBuild Build.

```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

Ausgabe:

```
{
  "build": {
    "startTime": 1556906956.318,
    "initiator": "my-aws-account-name",
    "projectName": "my-demo-project",
    "currentPhase": "COMPLETED",
    "cache": {
      "type": "NO_CACHE"
    },
    "source": {
      "insecureSsl": false,
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
      "type": "S3"
    },
    "id": "my-demo-project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
    "endTime": 1556906974.781,
    "phases": [
      {
        "durationInSeconds": 0,
        "phaseType": "SUBMITTED",
        "endTime": 1556906956.935,
        "phaseStatus": "SUCCEEDED",
        "startTime": 1556906956.318
      },
      {
        "durationInSeconds": 1,
        "phaseType": "QUEUED",
        "endTime": 1556906958.272,
        "phaseStatus": "SUCCEEDED",
        "startTime": 1556906956.935
      },
      {
        "phaseType": "PROVISIONING",
        "phaseStatus": "SUCCEEDED",
        "durationInSeconds": 14,
        "contexts": [
          {
            "message": "",

```

```
        "statusCode": ""
      }
    ],
    "endTime": 1556906972.847,
    "startTime": 1556906958.272
  },
  {
    "phaseType": "DOWNLOAD_SOURCE",
    "phaseStatus": "SUCCEEDED",
    "durationInSeconds": 0,
    "contexts": [
      {
        "message": "",
        "statusCode": ""
      }
    ],
    "endTime": 1556906973.552,
    "startTime": 1556906972.847
  },
  {
    "phaseType": "INSTALL",
    "phaseStatus": "SUCCEEDED",
    "durationInSeconds": 0,
    "contexts": [
      {
        "message": "",
        "statusCode": ""
      }
    ],
    "endTime": 1556906973.75,
    "startTime": 1556906973.552
  },
  {
    "phaseType": "PRE_BUILD",
    "phaseStatus": "SUCCEEDED",
    "durationInSeconds": 0,
    "contexts": [
      {
        "message": "",
        "statusCode": ""
      }
    ],
    "endTime": 1556906973.937,
    "startTime": 1556906973.75
  }
}
```

```
    },
    {
      "durationInSeconds": 0,
      "phaseType": "BUILD",
      "endTime": 1556906974.781,
      "phaseStatus": "STOPPED",
      "startTime": 1556906973.937
    },
    {
      "phaseType": "COMPLETED",
      "startTime": 1556906974.781
    }
  ],
  "artifacts": {
    "location": "arn:aws:s3::artifacts-override/my-demo-project",
    "encryptionDisabled": false,
    "overrideArtifactName": true
  },
  "buildComplete": true,
  "buildStatus": "STOPPED",
  "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
  "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
  "queuedTimeoutInMinutes": 5,
  "timeoutInMinutes": 60,
  "environment": {
    "type": "LINUX_CONTAINER",
    "environmentVariables": [],
    "computeType": "BUILD_GENERAL1_MEDIUM",
    "privilegedMode": false,
    "image": "aws/codebuild/standard:1.0",
    "imagePullCredentialsType": "CODEBUILD"
  },
  "logs": {
    "streamName": "1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-demo-project;stream=1a2b3c4d-5678-90ab-
cdef-11111EXAMPLE",
    "groupName": "/aws/codebuild/my-demo-project"
  },
  "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE"
}
```

```
}

```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Einen Build beenden \(AWS CLI\)](#).

- API-Einheiten finden Sie [StopBuild](#) in der AWS CLI Befehlsreferenz.

update-project

Das folgende Codebeispiel zeigt die Verwendung `update-project`.

AWS CLI

Um die Einstellungen eines AWS CodeBuild Build-Projekts zu ändern.

Im folgenden `update-project` Beispiel werden die Einstellungen des angegebenen CodeBuild Build-Projekts mit dem Namen geändert `my-demo-project`.

```
aws codebuild update-project --name "my-demo-project" \
  --description "This project is updated" \
  --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source-2.zip\"}" \
  --artifacts {"\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-
output-bucket-2\""} \
  --environment {"\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
  --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```

In der Ausgabe werden die aktualisierten Einstellungen angezeigt.

```
{
  "project": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
    "environment": {
      "privilegedMode": false,
      "environmentVariables": [],
      "type": "LINUX_CONTAINER",
      "image": "aws/codebuild/standard:1.0",
      "computeType": "BUILD_GENERAL1_MEDIUM",
      "imagePullCredentialsType": "CODEBUILD"
    }
  }
}
```

```
    },
    "queuedTimeoutInMinutes": 480,
    "description": "This project is updated",
    "artifacts": {
      "packaging": "NONE",
      "name": "my-demo-project",
      "type": "S3",
      "namespaceType": "NONE",
      "encryptionDisabled": false,
      "location": "codebuild-us-west-2-123456789012-output-bucket-2"
    },
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "badge": {
      "badgeEnabled": false
    },
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
    "lastModified": 1556840545.967,
    "tags": [],
    "timeoutInMinutes": 60,
    "created": 1556839783.274,
    "name": "my-demo-project",
    "cache": {
      "type": "NO_CACHE"
    },
    "source": {
      "type": "S3",
      "insecureSsl": false,
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source-2.zip"
    }
  }
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Ändern der Einstellungen eines Build-Projekts \(AWS CLI\)](#)

- API-Einheiten finden Sie [UpdateProject](#) in der AWS CLI Befehlsreferenz.

update-report-group

Das folgende Codebeispiel zeigt die Verwendung `update-report-group`.

AWS CLI

Um eine Berichtsgruppe in zu aktualisieren AWS CodeBuild.

Im folgenden `update-report-group` Beispiel wird der Exporttyp der Berichtsgruppe in „NO_EXPORT“ geändert.

```
aws codebuild update-report-group \  
  --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-  
group \  
  --export-config="exportConfigType=NO_EXPORT"
```

Ausgabe:

```
{  
  "reportGroup": {  
    "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-  
report-group",  
    "name": "cli-created-report-group",  
    "type": "TEST",  
    "exportConfig": {  
      "exportConfigType": "NO_EXPORT"  
    },  
    "created": 1602020686.009,  
    "lastModified": 1602021033.454,  
    "tags": []  
  }  
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Arbeiten mit Berichtsgruppen](#).

- API-Einheiten finden Sie [UpdateReportGroup](#) in der AWS CLI Befehlsreferenz.

update-webhook

Das folgende Codebeispiel zeigt die Verwendung `update-webhook`.

AWS CLI

Um den Webhook für ein AWS CodeBuild Projekt zu aktualisieren

Im folgenden `update-webhook` Beispiel wird ein Webhook für das angegebene CodeBuild Projekt mit zwei Filtergruppen aktualisiert. Der `--rotate-secret` Parameter gibt an, dass der geheime Schlüssel des Projekts jedes Mal GitHub rotiert wird, wenn eine Codeänderung einen Build auslöst. Die erste Filtergruppe gibt Pull-Anfragen an, die in Verzweigungen mit Git-Referenznamen, die dem regulären Ausdruck `^refs/heads/master$` entsprechen, und mit Kopfreferenzen, die `^refs/heads/myBranch$` entsprechen, erstellt, aktualisiert oder erneut geöffnet werden. Die zweite Filtergruppe spezifiziert Push-Anfragen für Branches mit Git-Referenznamen, die nicht dem regulären Ausdruck entsprechen `^refs/heads/myBranch$`.

```
aws codebuild update-webhook \
  --project-name Project2 \
  --rotate-secret \
  --filter-groups "[[{"type":"EVENT","pattern":"PULL_REQUEST_CREATED,
PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED"}, {"type":"HEAD_REF","pattern
":"^refs/heads/myBranch$"},"excludeMatchedPattern":true}, {"type":"BASE_REF
","pattern":"^refs/heads/master$"},"excludeMatchedPattern":true}], [{"type":
"EVENT","pattern":"PUSH"}, {"type":"HEAD_REF","pattern":"^refs/heads/
myBranch$"},"excludeMatchedPattern":true}]]"
```

Ausgabe:

```
{
  "webhook": {
    "filterGroups": [
      [
        {
          "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED",
          "type": "EVENT"
        },
        {
          "excludeMatchedPattern": true,
          "pattern": "refs/heads/myBranch$",
          "type": "HEAD_REF"
        },
        {
          "excludeMatchedPattern": true,
          "pattern": "refs/heads/master$",
          "type": "BASE_REF"
        }
      ],
      [
```



```
    {
      "pattern": "PUSH",
      "type": "EVENT"
    },
    {
      "excludeMatchedPattern": true,
      "pattern": "refs/heads/myBranch$",
      "type": "HEAD_REF"
    }
  ],
  "lastModifiedSecret": 1556312220.133
}
```

Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Ändern der Einstellungen eines Build-Projekts \(AWS CLI\)](#)

- API-Einheiten finden Sie [UpdateWebhook](#) in der AWS CLI Befehlsreferenz.

CodeCommit Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CodeCommit.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-approval-rule-template-with-repository

Das folgende Codebeispiel zeigt die Verwendung `associate-approval-rule-template-with-repository`.

AWS CLI

Um eine Vorlage für Genehmigungsregeln einem Repository zuzuordnen

Im folgenden `associate-approval-rule-template-with-repository` Beispiel wird die angegebene Vorlage für Genehmigungsregeln einem Repository mit dem Namen `zugeordnetMyDemoRepo`.

```
aws codecommit associate-approval-rule-template-with-repository \  
  --repository-name MyDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Zuordnen einer Genehmigungsregelvorgabe zu einem Repository](#).

- API-Einheiten finden Sie [AssociateApprovalRuleTemplateWithRepository](#) unter AWS CLI Befehlsreferenz.

`batch-associate-approval-rule-template-with-repositories`

Das folgende Codebeispiel zeigt die Verwendung `batch-associate-approval-rule-template-with-repositories`.

AWS CLI

Um eine Genehmigungsregelvorgabe mehreren Repositories in einem einzigen Vorgang zuzuordnen

Im folgenden `batch-associate-approval-rule-template-with-repositories` Beispiel wird die angegebene Vorlage für Genehmigungsregeln den Repositories mit dem Namen `MyDemoRepo` und `zugeordnet.MyOtherDemoRepo`.

Hinweis: Vorlagen für Genehmigungsregeln sind spezifisch für die AWS Region, in der sie erstellt wurden. Sie können nur Repositories in dieser AWS Region zugeordnet werden.

```
aws codecommit batch-associate-approval-rule-template-with-repositories \  
  --repository-names MyDemoRepo, MyOtherDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

Ausgabe:

```
{
  "associatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Zuordnen einer Genehmigungsregelvorlage zu einem Repository](#).

- API-Einheiten finden Sie [BatchAssociateApprovalRuleTemplateWithRepositories](#) unter AWS CLI Befehlsreferenz.

batch-describe-merge-conflicts

Das folgende Codebeispiel zeigt die Verwendung `batch-describe-merge-conflicts`.

AWS CLI

Um Informationen über Zusammenführungskonflikte in allen Dateien oder einer Teilmenge von Dateien in einer Zusammenführung zwischen zwei Commit-Spezifizierern abzurufen

Im folgenden `batch-describe-merge-conflicts` Beispiel werden die Zusammenführungskonflikte beim Zusammenführen eines Quellzweigs `feature-randomizationfeature` mit einem benannten Zielzweig `main` anhand der `THREE_WAY_MERGE` Strategie in einem Projektarchiv mit dem Namen `MyDemoRepo`

```
aws codecommit batch-describe-merge-conflicts \
  --source-commit-specifier feature-randomizationfeature \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE \
  --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "conflicts": [
    {
      "conflictMetadata": {
        "filePath": "readme.md",
```

```
    "fileSizes": {
      "source": 139,
      "destination": 230,
      "base": 85
    },
    "fileModes": {
      "source": "NORMAL",
      "destination": "NORMAL",
      "base": "NORMAL"
    },
    "objectTypes": {
      "source": "FILE",
      "destination": "FILE",
      "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
      "source": false,
      "destination": false,
      "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
      "source": "M",
      "destination": "M"
    }
  },
  "mergeHunks": [
    {
      "isConflict": true,
      "source": {
        "startLine": 0,
        "endLine": 3,
        "hunkContent": "VGhpcyBpEXAMPLE=="
      },
      "destination": {
        "startLine": 0,
        "endLine": 1,
        "hunkContent": "VXNlIHRoEXAMPLE="
      }
    }
  ]
}
```

```

    }
  ],
  "errors": [],
  "destinationCommitId": "86958e0aEXAMPLE",
  "sourceCommitId": "6ccd57fdEXAMPLE",
  "baseCommitId": "767b6958EXAMPLE"
}

```

Weitere Informationen finden Sie unter [Konflikte in einer Pull-Anfrage](#) lösen im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [BatchDescribeMergeConflicts](#) in der AWS CLI Befehlsreferenz.

batch-disassociate-approval-rule-template-from-repositories

Das folgende Codebeispiel zeigt die Verwendung `batch-disassociate-approval-rule-template-from-repositories`.

AWS CLI

Um eine Vorlage für Genehmigungsregeln in einem einzigen Vorgang von mehreren Repositorys zu trennen

Im folgenden `batch-disassociate-approval-rule-template-from-repositories` Beispiel wird die Zuordnung der angegebenen Genehmigungsregelvorlage zu den Repositorys mit dem Namen `MyDemoRepo` und `MyOtherDemoRepo` aufgehoben.

```

aws codecommit batch-disassociate-approval-rule-template-from-repositories \
  --repository-names MyDemoRepo, MyOtherDemoRepo \
  --approval-rule-template-name 1-approval-rule-for-all pull requests

```

Ausgabe:

```

{
  "disassociatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Trennen der Zuordnung zu einer Vorlage für Genehmigungsregeln](#).AWS CodeCommit

- API-Einheiten finden Sie [BatchDisassociateApprovalRuleTemplateFromRepositories](#) in der AWS CLI Befehlsreferenz.

batch-get-commits

Das folgende Codebeispiel zeigt die Verwendung `batch-get-commits`.

AWS CLI

Um Informationen über mehrere Commits anzuzeigen

Im folgenden `batch-get-commits` Beispiel werden Details zu den angegebenen Commits angezeigt.

```
aws codecommit batch-get-commits \
  --repository-name MyDemoRepo \
  --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

Ausgabe:

```
{
  "commits": [
    {
      "additionalData": "",
      "committer": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "author": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "commitId": "317f8570EXAMPLE",
      "treeId": "1f330709EXAMPLE",
      "parents": [
        "6e147360EXAMPLE"
      ],
      "message": "Change variable name and add new response element"
    }
  ]
}
```

```
  },
  {
    "additionalData": "",
    "committer": {
      "date": "1508280542 -0800",
      "name": "Li Juan",
      "email": "li_juan@example.com"
    },
    "author": {
      "date": "1508280542 -0800",
      "name": "Li Juan",
      "email": "li_juan@example.com"
    },
    "commitId": "4c925148EXAMPLE",
    "treeId": "1f330709EXAMPLE",
    "parents": [
      "317f8570EXAMPLE"
    ],
    "message": "Added new class"
  }
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Commit-Details anzeigen](#).

- API-Einheiten finden Sie [BatchGetCommits](#) in der AWS CLI Befehlsreferenz.

batch-get-repositories

Das folgende Codebeispiel zeigt die Verwendung `batch-get-repositories`.

AWS CLI

Um Details zu mehreren Repositories anzuzeigen

Dieses Beispiel zeigt Details zu mehreren AWS CodeCommit Repositories.

```
aws codecommit batch-get-repositories \
  --repository-names MyDemoRepo MyOtherDemoRepo
```

Ausgabe:

```
{
```

```

    "repositoriesNotFound": [],
    "repositories": [
      {
        "creationDate": 1429203623.625,
        "defaultBranch": "main",
        "repositoryName": "MyDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo",
        "lastModifiedDate": 1430783812.0869999,
        "repositoryDescription": "My demonstration repository",
        "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/
MyDemoRepo",
        "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"
        "accountId": "111111111111"
      },
      {
        "creationDate": 1429203623.627,
        "defaultBranch": "main",
        "repositoryName": "MyOtherDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/
MyOtherDemoRepo",
        "lastModifiedDate": 1430783812.0889999,
        "repositoryDescription": "My other demonstration repository",
        "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/
MyOtherDemoRepo",
        "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE",
        "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"
        "accountId": "111111111111"
      }
    ],
    "repositoriesNotFound": []
  }

```

- API-Einheiten finden Sie [BatchGetRepositories](#) in der AWS CLI Befehlsreferenz.

create-approval-rule-template

Das folgende Codebeispiel zeigt die Verwendung `create-approval-rule-template`.

AWS CLI

Um eine Vorlage für Genehmigungsregeln zu erstellen

Im folgenden `create-approval-rule-template` Beispiel wird eine Vorlage für Genehmigungsregeln erstellt, die so benannt ist `2-approver-rule-for-main`. The template requires two users who assume the role of `CodeCommitReview`, dass alle Pull-Requests genehmigt werden, bevor sie mit dem main Branch zusammengeführt werden können.

```
aws codecommit create-approval-rule-template \
  --approval-rule-template-name 2-approver-rule-for-main \
  --approval-rule-template-description "Requires two developers from the team to
  approve the pull request if the destination branch is main" \
  --approval-rule-template-content "{\"Version\": \"2018-11-08\",
  \"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
  \": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
  [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

Ausgabe:

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "2-approver-rule-for-main",
    "creationDate": 1571356106.936,
    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\",
    \"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
    \": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
    [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "approvalRuleTemplateDescription": "Requires two developers from the team to
    approve the pull request if the destination branch is main",
    "lastModifiedDate": 1571356106.936,
    "ruleContentSha256": "4711b576EXAMPLE"
  }
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Erstellen einer Vorlage für Genehmigungsregeln](#).

- API-Einheiten finden Sie [CreateApprovalRuleTemplate](#) in der AWS CLI Befehlsreferenz.

create-branch

Das folgende Codebeispiel zeigt die Verwendung `create-branch`.

AWS CLI

Um einen Zweig zu erstellen

In diesem Beispiel wird ein Branch in einem AWS CodeCommit Repository erstellt. Dieser Befehl liefert nur eine Ausgabe, wenn Fehler aufgetreten sind.

Befehl:

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
--commit-id 317f8570EXAMPLE
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [CreateBranch](#) in der AWS CLI Befehlsreferenz.

create-commit

Das folgende Codebeispiel zeigt die Verwendung `create-commit`.

AWS CLI

Um einen Commit zu erstellen

Das folgende `create-commit` Beispiel zeigt, wie Sie einen ersten Commit für ein Repository erstellen, das eine `readme.md` Datei zu einem `MyDemoRepo` im `main` Branch benannten Repository hinzufügt.

```
aws codecommit create-commit \  
  --repository-name MyDemoRepo \  
  --branch-name main \  
  --put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

Ausgabe:

```
{  
  "filesAdded": [  
    {  
      "blobId": "5e1c309d-EXAMPLE",
```

```

        "absolutePath": "readme.md",
        "fileMode": "NORMAL"
    }
  ],
  "commitId": "4df8b524-EXAMPLE",
  "treeId": "55b57003-EXAMPLE",
  "filesDeleted": [],
  "filesUpdated": []
}

```

Weitere Informationen finden Sie unter [Create a Commit AWS CodeCommit im AWS CodeCommit Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateCommit](#) in der AWS CLI Befehlsreferenz.

create-pull-request-approval-rule

Das folgende Codebeispiel zeigt die Verwendung `create-pull-request-approval-rule`.

AWS CLI

Um eine Genehmigungsregel für eine Pull-Anfrage zu erstellen

Im folgenden `create-pull-request-approval-rule` Beispiel wird eine Genehmigungsregel erstellt, die `Require two approved approvers` nach der angegebenen Pull-Anfrage benannt ist. Die Regel gibt an, dass zwei Genehmigungen aus einem Genehmigungs-pool erforderlich sind. Der Pool umfasst alle Benutzer, die darauf zugreifen, CodeCommit indem sie die Rolle von `CodeCommitReview` im `123456789012` AWS Konto übernehmen. Er umfasst auch entweder einen IAM Benutzer oder einen Verbundbenutzer, der `Nikhil_Jayashankar` aus demselben AWS Konto stammt.

```

aws codecommit create-pull-request-approval-rule \
  --approval-rule-name "Require two approved approvers" \
  --approval-rule-content "{\"Version\": \"2018-11-08\", \"Statements\":
  [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
  \": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
  \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"

```

Ausgabe:

```
{
```

```

"approvalRule": {
  "approvalRuleName": "Require two approved approvers",
  "lastModifiedDate": 1570752871.932,
  "ruleContentSha256": "7c44e6ebEXAMPLE",
  "creationDate": 1570752871.932,
  "approvalRuleId": "aac33506-EXAMPLE",
  "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":
[{\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}}\",
  "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
}
}

```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Erstellen einer Genehmigungsregel](#).

- API-Einheiten finden Sie [CreatePullRequestApprovalRule](#) in der AWS CLI Befehlsreferenz.

create-pull-request

Das folgende Codebeispiel zeigt die Verwendung `create-pull-request`.

AWS CLI

Um einen Pull-Request zu erstellen

Im folgenden `create-pull-request` Beispiel wird eine Pull-Anfrage mit dem Namen „Pronunciation Difficulty Analyzer“ mit der Beschreibung „Bitte überprüfen Sie diese Änderungen bis Dienstag“ erstellt, die auf den Quell-Branch „jane-branch“ abzielt und mit dem Standardbranch „main“ in einem Repository namens 'zusammengeführt werden soll. AWS CodeCommit MyDemoRepo

```

aws codecommit create-pull-request \
  --title "My Pull Request" \
  --description "Please review these changes by Tuesday" \
  --client-request-token 123Example \
  --targets repositoryName=MyDemoRepo,sourceReference=MyNewBranch

```

Ausgabe:

```
{
```

```

"pullRequest": {
  "approvalRules": [
    {
      "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\\\"DestinationReferences\\\": [\\\"refs/heads/main\\\"],\\\"Statements\\\": [{\\\"Type
\\\": \\\"Approvers\\\",\\\"NumberOfApprovalsNeeded\\\": 2,\\\"ApprovalPoolMembers\\\":
[\\\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\\\"]}]}",
      "approvalRuleId": "dd8b17fe-EXAMPLE",
      "approvalRuleName": "2-approver-rule-for-main",
      "creationDate": 1571356106.936,
      "lastModifiedDate": 571356106.936,
      "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
      "originApprovalRuleTemplate": {
        "approvalRuleTemplateId": "dd3d22fe-EXAMPLE",
        "approvalRuleTemplateName": "2-approver-rule-for-main"
      },
      "ruleContentSha256": "4711b576EXAMPLE"
    }
  ],
  "authorArn": "arn:aws:iam::111111111111:user/Jane_Doe",
  "description": "Please review these changes by Tuesday",
  "title": "Pronunciation difficulty analyzer",
  "pullRequestTargets": [
    {
      "destinationCommit": "5d036259EXAMPLE",
      "destinationReference": "refs/heads/main",
      "repositoryName": "MyDemoRepo",
      "sourceCommit": "317f8570EXAMPLE",
      "sourceReference": "refs/heads/jane-branch",
      "mergeMetadata": {
        "isMerged": false
      }
    }
  ],
  "lastActivityDate": 1508962823.285,
  "pullRequestId": "42",
  "clientRequestToken": "123Example",
  "pullRequestStatus": "OPEN",
  "creationDate": 1508962823.285
}
}

```

- Einzelheiten finden Sie in der Befehlsreferenz. API [CreatePullRequest](#)AWS CLI

create-repository

Das folgende Codebeispiel zeigt die Verwendung `create-repository`.

AWS CLI

So erstellen Sie ein Repository

In diesem Beispiel wird ein Repository erstellt und es dem AWS Konto des Benutzers zugeordnet.

Befehl:

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-  
description "My demonstration repository"
```

Ausgabe:

```
{  
  "repositoryMetadata": {  
    "repositoryName": "MyDemoRepo",  
    "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/  
repos/MyDemoRepo",  
    "lastModifiedDate": 1444766838.027,  
    "repositoryDescription": "My demonstration repository",  
    "cloneUrlHttp": "https://git-codecommit.us-east-1.amazonaws.com/v1/  
repos/MyDemoRepo",  
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
    "Arn": "arn:aws:codecommit:us-  
east-1:111111111111EXAMPLE:MyDemoRepo",  
    "accountId": "111111111111"  
  }  
}
```

- API-Einheiten finden Sie [CreateRepository](#) in der AWS CLI Befehlsreferenz.

create-unreferenced-merge-commit

Das folgende Codebeispiel zeigt die Verwendung `create-unreferenced-merge-commit`.

AWS CLI

Um einen nicht referenzierten Commit zu erstellen, der das Ergebnis der Zusammenführung von zwei Commit-Spezifizierern darstellt

Im folgenden `create-unreferenced-merge-commit` Beispiel wird ein Commit erstellt, das die Ergebnisse einer Zusammenführung zwischen einem Quell-Branche `bugfix-1234` mit einem Ziel-Branche darstellt, der `main` mithilfe der MERGE Strategie `THREE_WAY_MERGE` in einem Repository mit dem Namen benannt wurde. `MyDemoRepo`

```
aws codecommit create-unreferenced-merge-commit \
  --source-commit-specifier bugfix-1234 \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE \
  --repository-name MyDemoRepo \
  --name "Maria Garcia" \
  --email "maria_garcia@example.com" \
  --commit-message "Testing the results of this merge."
```

Ausgabe:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Weitere Informationen finden Sie unter [Konflikte in einer Pull-Anfrage](#) lösen im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [CreateUnreferencedMergeCommit](#) in der AWS CLI Befehlsreferenz.

credential-helper

Das folgende Codebeispiel zeigt die Verwendung `credential-helper`.

AWS CLI

Um den Credential Helper einzurichten, der in der AWS CLI Datei `with` enthalten ist AWS CodeCommit

Das `credential-helper` Hilfsprogramm ist nicht dafür konzipiert, direkt von der AWS CLI aufgerufen zu werden. Stattdessen soll es als Parameter für den `git config` Befehl zum Einrichten Ihres lokalen Computers verwendet werden. Es ermöglicht Git, eine kryptografisch signierte Version Ihrer IAM Benutzeranmeldedaten oder Ihrer EC2 Amazon-Instance-Rolle zu verwenden HTTPS, wann immer Git sich authentifizieren muss, um mit Repositorys AWS zu interagieren. CodeCommit

```
git config --global credential.helper '!aws codecommit credential-helper $@'  
git config --global credential.UseHttpPath true
```

Ausgabe:

```
[credential]  
  helper = !aws codecommit credential-helper $@  
  UseHttpPath = true
```

Weitere Informationen finden Sie im Benutzerhandbuch unter Einrichtung für die AWS CodeCommit Verwendung anderer Methoden. AWS CodeCommit Lesen Sie den Inhalt sorgfältig durch und befolgen Sie dann die Verfahren in einem der folgenden Themen: Für HTTPS Verbindungen unter Linux, macOS oder Unix oder Für HTTPS Verbindungen unter Windows im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [CredentialHelper](#) in der AWS CLI Befehlsreferenz.

delete-approval-rule-template

Das folgende Codebeispiel zeigt die Verwendung `delete-approval-rule-template`.

AWS CLI

Um eine Vorlage für Genehmigungsregeln zu löschen

Im folgenden `delete-approval-rule-template` Beispiel wird die angegebene Vorlage für Genehmigungsregeln gelöscht.

```
aws codecommit delete-approval-rule-template \  
  --approval-rule-template-name 1-approver-for-all-pull-requests
```

Ausgabe:

```
{  
  "approvalRuleTemplateId": "41de97b7-EXAMPLE"  
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Löschen einer Vorlage für Genehmigungsregeln](#).

- API-Einheiten finden Sie [DeleteApprovalRuleTemplate](#) in der AWS CLI Befehlsreferenz.

delete-branch

Das folgende Codebeispiel zeigt die Verwendung `delete-branch`.

AWS CLI

Um einen Zweig zu löschen

Dieses Beispiel zeigt, wie ein Branch in einem AWS CodeCommit Repository gelöscht wird.

Befehl:

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Ausgabe:

```
{
  "branch": {
    "commitId": "317f8570EXAMPLE",
    "branchName": "MyNewBranch"
  }
}
```

- API-Einheiten finden Sie [DeleteBranch](#) in der AWS CLI Befehlsreferenz.

delete-comment-content

Das folgende Codebeispiel zeigt die Verwendung `delete-comment-content`.

AWS CLI

Um den Inhalt eines Kommentars zu löschen

Sie können den Inhalt eines Kommentars nur löschen, wenn Sie den Kommentar selbst erstellt haben. Dieses Beispiel zeigt, wie der Inhalt eines Kommentars mit der vom System generierten ID gelöscht wird. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit delete-comment-content \
  --comment-id ff30b348EXAMPLEb9aa670f
```

Ausgabe:

```
{
  "comment": {
    "creationDate": 1508369768.142,
    "deleted": true,
    "lastModifiedDate": 1508369842.278,
    "clientRequestToken": "123Example",
    "commentId": "ff30b348EXAMPLEb9aa670f",
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "callerReactions": [],
    "reactionCounts":
    {
      "CLAP" : 1
    }
  }
}
```

- API-Einheiten finden Sie unter [DeleteCommentContent AWS CLI Befehlsreferenz](#).

delete-file

Das folgende Codebeispiel zeigt die Verwendung `delete-file`.

AWS CLI

Um eine Datei zu löschen

Das folgende `delete-file` Beispiel zeigt, wie eine Datei mit dem Namen `README.md` aus einem Branch `main` mit der neuesten Commit-ID von `c5709475EXAMPLE` in einem Repository mit dem Namen `gelöscht wirdMyDemoRepo`.

```
aws codecommit delete-file \
  --repository-name MyDemoRepo \
  --branch-name main \
  --file-path README.md \
  --parent-commit-id c5709475EXAMPLE
```

Ausgabe:

```
{
  "blobId": "559b44fEXAMPLE",
  "commitId": "353cf655EXAMPLE",
```

```
"filePath":"README.md",
"treeId":"6bc824cEXAMPLE"
}
```

Weitere Informationen finden Sie [AWS CodeCommit API Referenzhandbuch unter Bearbeiten oder Löschen einer Datei](#).

- API-Einheiten finden Sie [DeleteFile](#) in der AWS CLI Befehlsreferenz.

delete-pull-request-approval-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-pull-request-approval-rule`.

AWS CLI

Um eine Genehmigungsregel für eine Pull-Anfrage zu löschen

Im folgenden `delete-pull-request-approval-rule` Beispiel wird die Genehmigungsregel gelöscht, die `My Approval Rule` für die angegebene Pull-Anfrage benannt ist.

```
aws codecommit delete-pull-request-approval-rule \
  --approval-rule-name "My Approval Rule" \
  --pull-request-id 15
```

Ausgabe:

```
{
  "approvalRuleId": "077d8e8a8-EXAMPLE"
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Bearbeiten oder Löschen einer Genehmigungsregel](#).

- API-Einheiten finden Sie [DeletePullRequestApprovalRule](#) in der AWS CLI Befehlsreferenz.

delete-repository

Das folgende Codebeispiel zeigt die Verwendung `delete-repository`.

AWS CLI

So löschen Sie ein Repository

Dieses Beispiel zeigt, wie ein AWS CodeCommit Repository gelöscht wird.

Befehl:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

- API-Einheiten finden Sie [DeleteRepository](#) in der AWS CLI Befehlsreferenz.

describe-merge-conflicts

Das folgende Codebeispiel zeigt die Verwendung `describe-merge-conflicts`.

AWS CLI

Um detaillierte Informationen zu Zusammenführungskonflikten zu erhalten

Im folgenden `describe-merge-conflicts` Beispiel werden die Zusammenführungskonflikte für eine Datei mit dem Namen `readme.md` im angegebenen Quell- und Zielzweig mithilfe der MERGE Strategie `THREE_WAY_MERGE` ermittelt.

```
aws codecommit describe-merge-conflicts \
  --source-commit-specifier feature-randomizationfeature \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE \
  --file-path readme.md \
  --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "conflictMetadata": {
    "filePath": "readme.md",
    "fileSizes": {
      "source": 139,
      "destination": 230,
      "base": 85
    }
  }
}
```

```
    },
    "fileModes": {
      "source": "NORMAL",
      "destination": "NORMAL",
      "base": "NORMAL"
    },
    "objectTypes": {
      "source": "FILE",
      "destination": "FILE",
      "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
      "source": false,
      "destination": false,
      "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
      "source": "M",
      "destination": "M"
    }
  },
  "mergeHunks": [
    {
      "isConflict": true,
      "source": {
        "startLine": 0,
        "endLine": 3,
        "hunkContent": "VGhpcyBpEXAMPLE="
      },
      "destination": {
        "startLine": 0,
        "endLine": 1,
        "hunkContent": "VXNlIHRoEXAMPLE="
      }
    }
  ],
  "destinationCommitId": "86958e0aEXAMPLE",
  "sourceCommitId": "6ccd57fdEXAMPLE",
  "baseCommitId": "767b69580EXAMPLE"
```

```
}
```

Weitere Informationen finden Sie unter [Konflikte in einer Pull-Anfrage](#) lösen im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMergeConflicts](#) in der AWS CLI Befehlsreferenz.

describe-pull-request-events

Das folgende Codebeispiel zeigt die Verwendung `describe-pull-request-events`.

AWS CLI

Um Ereignisse in einer Pull-Anfrage anzuzeigen

Im folgenden `describe-pull-request-events` Beispiel werden die Ereignisse für eine Pull-Anfrage mit der ID '8' abgerufen.

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Ausgabe:

```
{
  "pullRequestEvents": [
    {
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_CREATED",
      "eventDate": 1510341779.53,
      "actor": "arn:aws:iam::111111111111:user/Zhang_Wei"
    },
    {
      "pullRequestStatusChangedEventMetadata": {
        "pullRequestStatus": "CLOSED"
      },
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_STATUS_CHANGED",
      "eventDate": 1510341930.72,
      "actor": "arn:aws:iam::111111111111:user/Jane_Doe"
    }
  ]
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [DescribePullRequestEvents](#).AWS CLI

disassociate-approval-rule-template-from-repository

Das folgende Codebeispiel zeigt die Verwendung `disassociate-approval-rule-template-from-repository`.

AWS CLI

Um die Zuordnung einer Genehmigungsregelvorlage zu einem Repository aufzuheben

Im folgenden `disassociate-approval-rule-template-from-repository` Beispiel wird die Zuordnung der angegebenen Genehmigungsregelvorlage zu einem Repository mit dem Namen aufgehoben. `MyDemoRepo`

```
aws codecommit disassociate-approval-rule-template-from-repository \  
  --repository-name MyDemoRepo \  
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Trennen der Zuordnung zu einer Vorlage für Genehmigungsregeln](#).

- API-Einheiten finden Sie [DisassociateApprovalRuleTemplateFromRepository](#) in der AWS CLI Befehlsreferenz.

evaluate-pull-request-approval-rules

Das folgende Codebeispiel zeigt die Verwendung `evaluate-pull-request-approval-rules`.

AWS CLI

Um zu bewerten, ob für eine Pull-Anfrage alle Genehmigungsregeln erfüllt sind

Im folgenden `evaluate-pull-request-approval-rules` Beispiel wird der Status der Genehmigungsregeln für den angegebenen Pull-Request ausgewertet. In diesem Beispiel wurde eine Genehmigungsregel für die Pull-Anfrage nicht erfüllt, sodass die Ausgabe des Befehls den `approved` Wert anzeigt. `false`

```
aws codecommit evaluate-pull-request-approval-rules \  
  --pull-request-id 1-approver-rule-for-all-pull-requests
```

```
--pull-request-id 27 \  
--revision-id 9f29d167EXAMPLE
```

Ausgabe:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ],  
    "overridden": false,  
    "approvalRulesSatisfied": []  
  }  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Zusammenführen einer Pull-Anfrage](#).AWS CodeCommit

- API-Einheiten finden Sie [EvaluatePullRequestApprovalRules](#) in der AWS CLI Befehlsreferenz.

get-approval-rule-template

Das folgende Codebeispiel zeigt die Verwendung `get-approval-rule-template`.

AWS CLI

Um den Inhalt einer Vorlage für Genehmigungsregeln abzurufen

Im folgenden `get-approval-rule-template` Beispiel wird der Inhalt einer Genehmigungsregelvorlage mit dem Namen `1-approver-rule-for-all-pull-requests` abgerufen.

```
aws codecommit get-approval-rule-template \  
--approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Ausgabe:

```
{  
  "approvalRuleTemplate": {
```



```

    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\":
    [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":
    [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "ruleContentSha256": "621181bbEXAMPLE",
    "lastModifiedDate": 1571356106.936,
    "creationDate": 1571356106.936,
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan",
    "approvalRuleTemplateId": "a29abb15-EXAMPLE",
    "approvalRuleTemplateDescription": "All pull requests must be approved by
    one developer on the team."
  }
}

```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#).

- API-Einheiten finden Sie [GetApprovalRuleTemplate](#) in der AWS CLI Befehlsreferenz.

get-blob

Das folgende Codebeispiel zeigt die Verwendung von `get-blob`.

AWS CLI

So zeigen Sie Informationen zu einem Git-Blob-Objekt an

Das folgende `get-blob` Beispiel ruft Informationen über einen Git-Blob mit der ID '2eb4af3b' in einem Repository namens EXAMPLE " ab. AWS CodeCommit MyDemoRepo

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```

Ausgabe:

```
{
  "content": "QSBcCaW5hcnkgTGfYToEXAMPLE="
}
```

- Einzelheiten API finden Sie unter Befehlsreferenz. [GetBlob](#) AWS CLI

get-branch

Das folgende Codebeispiel zeigt die Verwendung `get-branch`.

AWS CLI

Um Informationen über eine Filiale zu erhalten

In diesem Beispiel werden Informationen über einen Branch in einem AWS CodeCommit Repository abgerufen.

Befehl:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Ausgabe:

```
{
  "BranchInfo": {
    "commitID": "317f8570EXAMPLE",
    "branchName": "MyNewBranch"
  }
}
```

- API-Einheiten finden Sie [GetBranch](#) in der AWS CLI Befehlsreferenz.

get-comment-reactions

Das folgende Codebeispiel zeigt die Verwendung `get-comment-reactions`.

AWS CLI

Um Emoji-Reaktionen auf einen Kommentar anzuzeigen

Das folgende `get-comment-reactions` Beispiel listet alle Emoji-Reaktionen auf einen Kommentar mit der ID von `abcd1234EXAMPLEb5678efgh` auf. Wenn die Schriftart für Ihre Shell die Anzeige von Emoji Version 1.0 unterstützt, wird dies in der Ausgabe für `emoji` das Emoji angezeigt.

```
aws codecommit get-comment-reactions \
  --comment-id abcd1234EXAMPLEb5678efgh
```

Ausgabe:

```
{
  "reactionsForComment": {
    [
      {
        "reaction": {
          "emoji": "??",
          "shortCode": "thumbsup",
          "unicode": "U+1F44D"
        },
        "users": [
          "arn:aws:iam::123456789012:user/Li_Juan",
          "arn:aws:iam::123456789012:user/Mary_Major",
          "arn:aws:iam::123456789012:user/Jorge_Souza"
        ]
      },
      {
        "reaction": {
          "emoji": "??",
          "shortCode": "thumbsdown",
          "unicode": "U+1F44E"
        },
        "users": [
          "arn:aws:iam::123456789012:user/Nikhil_Jayashankar"
        ]
      },
      {
        "reaction": {
          "emoji": "??",
          "shortCode": "confused",
          "unicode": "U+1F615"
        },
        "users": [
          "arn:aws:iam::123456789012:user/Saanvi_Sarkar"
        ]
      }
    ]
  }
}
```

Weitere Informationen finden Sie [im AWS CodeCommit AWS CodeCommit Benutzerhandbuch unter Kommentieren eines Commits](#).

- API-Einheiten finden Sie [GetCommentReactions](#) in der AWS CLI Befehlsreferenz.

get-comment

Das folgende Codebeispiel zeigt die Verwendung `get-comment`.

AWS CLI

Um Details eines Kommentars anzuzeigen

Dieses Beispiel zeigt, wie Details eines Kommentars mit der vom System generierten Kommentar-ID von angezeigt werden. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit get-comment \  
--comment-id ff30b348EXAMPLEb9aa670f
```

Ausgabe:

```
{  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "123Example",  
    "commentId": "ff30b348EXAMPLEb9aa670f",  
    "content": "Whoops - I meant to add this comment to the line, but I don't  
see how to delete it.",  
    "creationDate": 1508369768.142,  
    "deleted": false,  
    "commentId": "",  
    "lastModifiedDate": 1508369842.278,  
    "callerReactions": [],  
    "reactionCounts":  
    {  
      "SMILE" : 6,  
      "THUMBSUP" : 1  
    }  
  }  
}
```

- API-Einheiten finden Sie unter [GetComment AWS CLI](#) Befehlsreferenz.

get-comments-for-compared-commit

Das folgende Codebeispiel zeigt die Verwendung `get-comments-for-compared-commit`.

AWS CLI

Um Kommentare zu einem Commit anzusehen

Dieses Beispiel zeigt, wie Kommentare angezeigt werden, die zum Vergleich zwischen zwei Commits in einem Repository mit dem Namen `MyDemoRepo` gemacht wurden.

```
aws codecommit get-comments-for-compared-commit \  
  --repository-name MyDemoRepo \  
  --before-commit-ID 6e147360EXAMPLE \  
  --after-commit-id 317f8570EXAMPLE
```

Ausgabe:

```
{  
  "commentsForComparedCommitData": [  
    {  
      "afterBlobId": "1f330709EXAMPLE",  
      "afterCommitId": "317f8570EXAMPLE",  
      "beforeBlobId": "80906a4cEXAMPLE",  
      "beforeCommitId": "6e147360EXAMPLE",  
      "comments": [  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
          "clientRequestToken": "123Example",  
          "commentId": "ff30b348EXAMPLEb9aa670f",  
          "content": "Whoops - I meant to add this comment to the line,  
not the file, but I don't see how to delete it.",  
          "creationDate": 1508369768.142,  
          "deleted": false,  
          "CommentId": "123abc-EXAMPLE",  
          "lastModifiedDate": 1508369842.278,  
          "callerReactions": [],  
          "reactionCounts":  
            {  
              "SMILE" : 6,  
              "THUMBSUP" : 1  
            }  
        }  
      ],  
    }  
  ],  
}
```

```

    {
      "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
      "clientRequestToken": "123Example",
      "commentId": "553b509bEXAMPLE56198325",
      "content": "Can you add a test case for this?",
      "creationDate": 1508369612.240,
      "deleted": false,
      "commentId": "456def-EXAMPLE",
      "lastModifiedDate": 1508369612.240,
      "callerReactions": [],
      "reactionCounts":
        {
          "THUMBSUP" : 2
        }
    }
  ],
  "location": {
    "filePath": "cl_sample.js",
    "filePosition": 1232,
    "relativeFileVersion": "after"
  },
  "repositoryName": "MyDemoRepo"
}
],
"nextToken": "exampleToken"
}

```

- API-Einheiten finden Sie [GetCommentsForComparedCommit](#) in der AWS CLI Befehlsreferenz.

get-comments-for-pull-request

Das folgende Codebeispiel zeigt die Verwendung `get-comments-for-pull-request`.

AWS CLI

Um Kommentare zu einer Pull-Anfrage anzuzeigen

Dieses Beispiel zeigt, wie Kommentare zu einer Pull-Anfrage in einem Repository mit dem Namen `MyDemoRepo` angezeigt werden.

```
aws codecommit get-comments-for-pull-request \
```

```
--repository-name MyDemoRepo \  
--before-commit-ID 317f8570EXAMPLE \  
--after-commit-id 5d036259EXAMPLE
```

Ausgabe:

```
{  
  "commentsForPullRequestData": [  
    {  
      "afterBlobId": "1f330709EXAMPLE",  
      "afterCommitId": "5d036259EXAMPLE",  
      "beforeBlobId": "80906a4cEXAMPLE",  
      "beforeCommitId": "317f8570EXAMPLE",  
      "comments": [  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",  
          "clientRequestToken": "",  
          "commentId": "abcd1234EXAMPLEb5678efgh",  
          "content": "These don't appear to be used anywhere. Can we  
remove them?",  
          "creationDate": 1508369622.123,  
          "deleted": false,  
          "lastModifiedDate": 1508369622.123,  
          "callerReactions": [],  
          "reactionCounts":  
            {  
              "THUMBSUP" : 6,  
              "CONFUSED" : 1  
            }  
        },  
        {  
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
          "clientRequestToken": "",  
          "commentId": "442b498bEXAMPLE5756813",  
          "content": "Good catch. I'll remove them.",  
          "creationDate": 1508369829.104,  
          "deleted": false,  
          "lastModifiedDate": 150836912.273,  
          "callerReactions": ["THUMBSUP"]  
          "reactionCounts":  
            {  
              "THUMBSUP" : 14  
            }  
        }  
      ]  
    }  
  ]  
}
```

```

    }
  ],
  "location": {
    "filePath": "ahs_count.py",
    "filePosition": 367,
    "relativeFileVersion": "AFTER"
  },
  "repositoryName": "MyDemoRepo",
  "pullRequestId": "42"
}
],
"nextToken": "exampleToken"
}

```

- API-Einheiten finden Sie [GetCommentsForPullRequest](#) in der AWS CLI Befehlsreferenz.

get-commit

Das folgende Codebeispiel zeigt die Verwendung `get-commit`.

AWS CLI

Um Informationen über einen Commit in einem Repository anzuzeigen

Dieses Beispiel zeigt Details zu einem Commit mit der vom System generierten ID '7e9fd3091thisisanexamplethisisanexample1' in einem Repository mit dem Namen ". AWS CodeCommit MyDemoRepo

Befehl:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id 7e9fd3091thisisanexamplethisisanexample1
```

Ausgabe:

```

{
  "commit": {
    "additionalData": "",
    "committer": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    }
  }
}

```



```

    },
    "author": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "treeId": "347a3408thisisanexamplereidexample",
    "parents": [
      "7aa87a031thisisanexamplethisisanexample1"
    ],
    "message": "Fix incorrect variable name"
  }
}

```

- API [GetCommit](#) Einzelheiten AWS CLI finden Sie in der Befehlsreferenz.

get-differences

Das folgende Codebeispiel zeigt die Verwendung `get-differences`.

AWS CLI

Um Informationen über Unterschiede für einen Commit-Spezifizierer in einem Repository zu erhalten

In diesem Beispiel werden Metadateninformationen zu Änderungen zwischen zwei Commit-Spezifizierern (BranchHEAD, Tag oder andere vollqualifizierte Verweise wie CommitIDs) in einem umbenannten Ordner im AWS CodeCommit Repository mit dem Namen angezeigt. `MyDemoRepo` Das Beispiel enthält mehrere Optionen, die nicht erforderlich sind, darunter `--before-commit-specifier`, `--before-path` und `--after-path`, um besser zu veranschaulichen, wie Sie diese Optionen verwenden können, um die Ergebnisse einzuschränken. Die Antwort beinhaltet Berechtigungen für den Dateimodus.

Befehl:

```

aws codecommit get-differences --repository-name MyDemoRepo --before-
commit-specifier 955bba12thisisanexamplethisisanexample --after-commit-
specifier 14a95463thisisanexamplethisisanexample --before-path tmp/example-folder --
after-path tmp/renamed-folder

```

Ausgabe:

```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
        "mode": "100644"
      },
      "changeType": "M",
      "beforeBlob": {
        "path": "blob.txt",
        "blobId": "bf7fcf281thisisanexamplethisisanexample1",
        "mode": "100644"
      }
    }
  ]
}
```

- API-Einheiten finden Sie [GetDifferences](#) in der AWS CLI Befehlsreferenz.

get-file

Das folgende Codebeispiel zeigt die Verwendung `get-file`.

AWS CLI

Um den Base-64-codierten Inhalt einer Datei in einem Repository abzurufen, AWS CodeCommit

Das folgende `get-file` Beispiel zeigt, wie der Base-64-kodierte Inhalt einer Datei abgerufen werden kann, die `README.md` aus einem Zweig benannt ist, der in einem Repository mit dem Namen `main` benannt ist.

```
aws codecommit get-file \
  --repository-name MyDemoRepo \
  --commit-specifier main \
  --file-path README.md
```

Ausgabe:

```
{
  "blobId": "559b44fEXAMPLE",
```

```
"commitId":"c5709475EXAMPLE",
"fileContent":"IyBQaHVzEXAMPLE",
"filePath":"README.md",
"fileMode":"NORMAL",
"fileSize":1563
}
```

Weitere Informationen finden Sie [GetFile](#) im Referenzhandbuch.AWS CodeCommit API

- APIEinzelheiten finden Sie [GetFile](#) in der AWS CLI Befehlsreferenz.

get-folder

Das folgende Codebeispiel zeigt die Verwendung `get-folder`.

AWS CLI

Um den Inhalt eines Ordners in einem AWS CodeCommit Repository abzurufen

Das folgende `get-folder` Beispiel zeigt, wie Sie den Inhalt eines Ordners der obersten Ebene aus einem Repository mit dem Namen `MyDemoRepo` abrufen.

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

Ausgabe:

```
{
  "commitId":"c5709475EXAMPLE",
  "files":[
    {
      "absolutePath": ".gitignore",
      "blobId": "74094e8bEXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": ".gitignore"
    },
    {
      "absolutePath": "Gemfile",
      "blobId": "9ceb72f6EXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": "Gemfile"
    },
    {
```

```

        "absolutePath":"Gemfile.lock",
        "blobId":"795c4a2aEXAMPLE",
        "fileMode":"NORMAL",
        "relativePath":"Gemfile.lock"
    },
    {
        "absolutePath":"LICENSE.txt",
        "blobId":"0c7932c8EXAMPLE",
        "fileMode":"NORMAL",
        "relativePath":"LICENSE.txt"
    },
    {
        "absolutePath":"README.md",
        "blobId":"559b44feEXAMPLE",
        "fileMode":"NORMAL",
        "relativePath":"README.md"
    }
],
"folderPath":"",
"subFolders":[
    {
        "absolutePath":"public",
        "relativePath":"public",
        "treeId":"d5e92ae3aEXAMPLE"
    },
    {
        "absolutePath":"tmp",
        "relativePath":"tmp",
        "treeId":"d564d0bcEXAMPLE"
    }
],
"subModules":[],
"symbolicLinks":[],
"treeId":"7b3c4dadEXAMPLE"
}

```

Weitere Informationen finden Sie `GetFolder` im AWS CodeCommit APIReferenzhandbuch.

- APIEinzelheiten finden Sie [GetFolder](#) in der AWS CLI Befehlsreferenz.

get-merge-commit

Das folgende Codebeispiel zeigt die Verwendung `get-merge-commit`.

AWS CLI

Um detaillierte Informationen zu einem Merge-Commit zu erhalten

Das folgende `get-merge-commit` Beispiel zeigt Details zu einem Merge-Commit für den Quell-Branch `bugfix-bug1234` mit einem Ziel-Branch, der nach der MERGE Strategie `THREE_WAY_MERGE` benannt wurde, in einem Repository `main` mit dem Namen `MyDemoRepo`.

```
aws codecommit get-merge-commit \  
  --source-commit-specifier bugfix-bug1234 \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE \  
  --repository-name MyDemoRepo
```

Ausgabe:

```
{  
  "sourceCommitId": "c5709475EXAMPLE",  
  "destinationCommitId": "317f8570EXAMPLE",  
  "baseCommitId": "fb12a539EXAMPLE",  
  "mergeCommitId": "ffc4d608eEXAMPLE"  
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Commit-Details anzeigen](#).

- API-Einheiten finden Sie [GetMergeCommit](#) in der AWS CLI Befehlsreferenz.

get-merge-conflicts

Das folgende Codebeispiel zeigt die Verwendung `get-merge-conflicts`.

AWS CLI

Um zu sehen, ob es bei einer Pull-Anfrage Zusammenführungskonflikte gibt

Das folgende `get-merge-conflicts` Beispiel zeigt, ob es irgendwelche Zusammenführungskonflikte zwischen der Spitze eines Quell-Branche mit dem Namen `feature-randomizationfeature` und einem Ziel-Branch namens `'main'` in einem Repository mit dem Namen `gibtMyDemoRepo`.

```
aws codecommit get-merge-conflicts \  
  --repository-name MyDemoRepo \  
  --source-commit-specifier feature-randomizationfeature \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE
```

Ausgabe:

```
{  
  "mergeable": false,  
  "destinationCommitId": "86958e0aEXAMPLE",  
  "sourceCommitId": "6ccd57fdEXAMPLE",  
  "baseCommitId": "767b6958EXAMPLE",  
  "conflictMetadataList": [  
    {  
      "filePath": "readme.md",  
      "fileSizes": {  
        "source": 139,  
        "destination": 230,  
        "base": 85  
      },  
      "fileModes": {  
        "source": "NORMAL",  
        "destination": "NORMAL",  
        "base": "NORMAL"  
      },  
      "objectTypes": {  
        "source": "FILE",  
        "destination": "FILE",  
        "base": "FILE"  
      },  
      "numberOfConflicts": 1,  
      "isBinaryFile": {  
        "source": false,  
        "destination": false,  
        "base": false  
      },  
      "contentConflict": true,  
      "fileModeConflict": false,  
      "objectTypeConflict": false,  
      "mergeOperations": {  
        "source": "M",  
        "destination": "M"  
      }  
    }  
  ]  
}
```

```

    }
  }
]
}

```

- API-Einheiten finden Sie [GetMergeConflicts](#) in der AWS CLI Befehlsreferenz.

get-merge-options

Das folgende Codebeispiel zeigt die Verwendung `get-merge-options`.

AWS CLI

Um Informationen zu den Zusammenführungsoptionen zu erhalten, die für das Zusammenführen zweier bestimmter Zweige verfügbar sind

Im folgenden `get-merge-options` Beispiel werden die verfügbaren Zusammenführungsoptionen für das Zusammenführen eines Quellzweigs `bugfix-bug1234` mit einem Zielzweig bestimmt, der `main` in einem Repository mit dem Namen `MyDemoRepo` benannt ist.

```

aws codecommit get-merge-options \
  --source-commit-specifier bugfix-bug1234 \
  --destination-commit-specifier main \
  --repository-name MyDemoRepo

```

Ausgabe:

```

{
  "mergeOptions": [
    "FAST_FORWARD_MERGE",
    "SQUASH_MERGE",
    "THREE_WAY_MERGE"
  ],
  "sourceCommitId": "18059494EXAMPLE",
  "destinationCommitId": "ffd3311dEXAMPLE",
  "baseCommitId": "ffd3311dEXAMPLE"
}

```

Weitere Informationen finden Sie unter [Konflikte in einer Pull-Anfrage lösen](#) im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [GetMergeOptions](#) in der AWS CLI Befehlsreferenz.

get-pull-request-approval-states

Das folgende Codebeispiel zeigt die Verwendung `get-pull-request-approval-states`.

AWS CLI

Um Genehmigungen für eine Pull-Anfrage einzusehen

Das folgende `get-pull-request-approval-states` Beispiel gibt Genehmigungen für den angegebenen Pull-Request zurück.

```
aws codecommit get-pull-request-approval-states \  
  --pull-request-id 8 \  
  --revision-id 9f29d167EXAMPLE
```

Ausgabe:

```
{  
  "approvals": [  
    {  
      "userArn": "arn:aws:iam::123456789012:user/Mary_Major",  
      "approvalState": "APPROVE"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Pull Requests anzeigen](#).

- API-Einheiten finden Sie [GetPullRequestApprovalStates](#) unter AWS CLI Befehlsreferenz.

get-pull-request-override-state

Das folgende Codebeispiel zeigt die Verwendung `get-pull-request-override-state`.

AWS CLI

Um Informationen über den Override-Status einer Pull-Anfrage abzurufen

Das folgende `get-pull-request-override-state` Beispiel gibt den Override-Status für den angegebenen Pull-Request zurück. In diesem Beispiel wurden die Genehmigungsregeln für die Pull-Anfrage von einem Benutzer namens Mary Major außer Kraft gesetzt, sodass die Ausgabe den Wert zurückgibt. `true` :

```
aws codecommit get-pull-request-override-state \  
  --pull-request-id 34 \  
  --revision-id 9f29d167EXAMPLE
```

Ausgabe:

```
{  
  "overridden": true,  
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"  
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Überschreiben von Genehmigungsregeln für einen Pull Request](#).

- API-Einheiten finden Sie [GetPullRequestOverrideState](#) unter AWS CLI Befehlsreferenz.

get-pull-request

Das folgende Codebeispiel zeigt die Verwendung `get-pull-request`.

AWS CLI

Um Details einer Pull-Anfrage anzuzeigen

Dieses Beispiel zeigt, wie Informationen zu einer Pull-Anfrage mit der ID von angezeigt 27 werden.

```
aws codecommit get-pull-request \  
  --pull-request-id 27
```

Ausgabe:

```
{  
  "pullRequest": {
```

```

    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":
[{\": \"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}}]",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "lastActivityDate": 1562619583.565,
    "pullRequestTargets": [
      {
        "sourceCommit": "ca45e279EXAMPLE",
        "sourceReference": "refs/heads/bugfix-1234",
        "mergeBase": "a99f5ddbEXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": false
        },
        "destinationCommit": "2abfc6beEXAMPLE",
        "repositoryName": "MyDemoRepo"
      }
    ],
    "revisionId": "e47def21EXAMPLE",
    "title": "Quick fix for bug 1234",
    "authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
    "clientRequestToken": "d8d7612e-EXAMPLE",
    "creationDate": 1562619583.565,
    "pullRequestId": "27",
    "pullRequestStatus": "OPEN"
  }
}

```

- API-Einheiten finden Sie [GetPullRequest](#) unter AWS CLI Befehlsreferenz.

get-repository-triggers

Das folgende Codebeispiel zeigt die Verwendung `get-repository-triggers`.

AWS CLI

Um Informationen über Trigger in einem Repository abzurufen

Dieses Beispiel zeigt Details zu Triggern, die für ein AWS CodeCommit Repository mit dem Namen konfiguriert wurden `MyDemoRepo`.

```
aws codecommit get-repository-triggers \  
  --repository-name MyDemoRepo
```

Ausgabe:

```
{  
  "configurationId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
  "triggers": [  
    {  
      "destinationArn": "arn:aws:sns:us-  
east-1:111111111111:MyCodeCommitTopic",  
      "branches": [  
        "main",  
        "preprod"  
      ],  
      "name": "MyFirstTrigger",  
      "customData": "",  
      "events": [  
        "all"  
      ]  
    },  
    {  
      "destinationArn": "arn:aws:lambda:us-  
east-1:111111111111:function:MyCodeCommitPythonFunction",  
      "branches": [],  
      "name": "MySecondTrigger",  
      "customData": "EXAMPLE",  
      "events": [  
        "all"  
      ]  
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetRepositoryTriggers](#) in der AWS CLI Befehlsreferenz.

get-repository

Das folgende Codebeispiel zeigt die Verwendung `get-repository`.

AWS CLI

Um Informationen über ein Repository zu erhalten

Dieses Beispiel zeigt Details zu einem AWS CodeCommit Repository.

```
aws codecommit get-repository \  
  --repository-name MyDemoRepo
```

Ausgabe:

```
{  
  "repositoryMetadata": {  
    "creationDate": 1429203623.625,  
    "defaultBranch": "main",  
    "repositoryName": "MyDemoRepo",  
    "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/v1/  
repos/MyDemoRepo",  
    "lastModifiedDate": 1430783812.0869999,  
    "repositoryDescription": "My demonstration repository",  
    "cloneUrlHttp": "https://codecommit.us-east-1.amazonaws.com/v1/repos/  
MyDemoRepo",  
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
    "Arn": "arn:aws:codecommit:us-east-1:80398EXAMPLE:MyDemoRepo",  
    "accountId": "111111111111"  
  }  
}
```

- API-Einheiten finden Sie [GetRepository](#) in der AWS CLI Befehlsreferenz.

list-approval-rule-templates

Das folgende Codebeispiel zeigt die Verwendung `list-approval-rule-templates`.

AWS CLI

Um alle Vorlagen für Genehmigungsregeln in einer AWS Region aufzulisten

Im folgenden `list-approval-rule-templates` Beispiel werden alle Vorlagen für Genehmigungsregeln in der angegebenen Region aufgeführt. Wenn keine AWS Region als Parameter angegeben ist, gibt der Befehl Genehmigungsregelvorlagen für die Region zurück, die in dem AWS CLI Profil angegeben ist, das zur Ausführung des Befehls verwendet wurde.

```
aws codecommit list-approval-rule-templates \  
  --region us-east-2
```

Ausgabe:

```
{  
  "approvalRuleTemplateName": [  
    "2-approver-rule-for-main",  
    "1-approver-rule-for-all-pull-requests"  
  ]  
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#).

- API-Einheiten finden Sie [ListApprovalRuleTemplates](#) in der AWS CLI Befehlsreferenz.

list-associated-approval-rule-templates-for-repository

Das folgende Codebeispiel zeigt die Verwendung `list-associated-approval-rule-templates-for-repository`.

AWS CLI

Um alle Vorlagen aufzulisten, die einem Repository zugeordnet sind

Im folgenden `list-associated-approval-rule-templates-for-repository` Beispiel werden alle Vorlagen für Genehmigungsregeln aufgeführt, die einem Repository mit dem Namen `MyDemoRepo` zugeordnet sind.

```
aws codecommit list-associated-approval-rule-templates-for-repository \  
  --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "approvalRuleTemplateNames": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#).

- API-Einheiten finden Sie [ListAssociatedApprovalRuleTemplatesForRepository](#) in der AWS CLI Befehlsreferenz.

list-branches

Das folgende Codebeispiel zeigt die Verwendung `list-branches`.

AWS CLI

Um eine Liste von Zweignamen anzuzeigen

In diesem Beispiel werden alle Zweignamen in einem AWS CodeCommit Repository aufgeführt.

```
aws codecommit list-branches \
  --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "branches": [
    "MyNewBranch",
    "main"
  ]
}
```

- API-Einheiten finden Sie [ListBranches](#) in der AWS CLI Befehlsreferenz.

list-pull-requests

Das folgende Codebeispiel zeigt die Verwendung `list-pull-requests`.

AWS CLI

Um eine Liste von Pull-Requests in einem Repository anzuzeigen

Dieses Beispiel zeigt, wie Pull-Requests, die von einem IAM Benutzer mit dem Namen ARN 'arn:aws:iam: :111111111111:user/li_Juan' und dem Status " erstellt wurden, in einem Repository mit dem Namen " aufgelistet werden: CLOSED AWS CodeCommit MyDemoRepo

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "nextToken": "",
  "pullRequestIds": ["2", "12", "16", "22", "23", "35", "30", "39", "47"]
}
```

- API-Einheiten [ListPullRequests AWS CLI](#) finden Sie in der Befehlsreferenz.

list-repositories-for-approval-rule-template

Das folgende Codebeispiel zeigt die Verwendung `list-repositories-for-approval-rule-template`.

AWS CLI

Um alle Repositories aufzulisten, die einer Vorlage zugeordnet sind

Das folgende `list-repositories-for-approval-rule-template` Beispiel listet alle Repositories auf, die mit der angegebenen Genehmigungsregelvorlage verknüpft sind.

```
aws codecommit list-repositories-for-approval-rule-template \
  --approval-rule-template-name 2-approver-rule-for-main
```

Ausgabe:

```
{
  "repositoryNames": [
    "MyDemoRepo",
  ]
}
```

```
    "MyClonedRepo"  
  ]  
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#).

- API-Einheiten finden Sie [ListRepositoriesForApprovalRuleTemplate](#) in der AWS CLI Befehlsreferenz.

list-repositories

Das folgende Codebeispiel zeigt die Verwendung `list-repositories`.

AWS CLI

Um eine Liste von Repositories anzuzeigen

In diesem Beispiel werden alle AWS CodeCommit Repositories aufgelistet, die dem Konto des Benutzers AWS zugeordnet sind.

Befehl:

```
aws codecommit list-repositories
```

Ausgabe:

```
{  
  "repositories": [  
    {  
      "repositoryName": "MyDemoRepo"  
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
    },  
    {  
      "repositoryName": "MyOtherDemoRepo"  
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListRepositories](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die AWS Tags für ein Repository anzuzeigen

Das folgende `list-tags-for-resource` Beispiel listet Tag-Schlüssel und Tag-Werte für das angegebene Repository auf.

```
aws codecommit list-tags-for-resource \
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

Ausgabe:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "Saanvi"
  }
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Tags für ein Repository anzeigen](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

merge-branches-by-fast-forward

Das folgende Codebeispiel zeigt die Verwendung `merge-branches-by-fast-forward`.

AWS CLI

Um zwei Zweige mithilfe der Fast-Forward-Merge-Strategie zusammenzuführen

Im folgenden `merge-branches-by-fast-forward` Beispiel wird der angegebene Quellzweig mit dem angegebenen Zielzweig in einem Repository mit dem Namen `MyDemoRepo`

`MyDemoRepo`

```
aws codecommit merge-branches-by-fast-forward \
```

```
--source-commit-specifier bugfix-bug1234 \  
--destination-commit-specifier bugfix-bug1233 \  
--repository-name MyDemoRepo
```

Ausgabe:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Branches vergleichen und zusammenführen](#).AWS CodeCommit

- API-Einheiten finden Sie [MergeBranchesByFastForward](#) in der AWS CLI Befehlsreferenz.

merge-branches-by-squash

Das folgende Codebeispiel zeigt die Verwendung `merge-branches-by-squash`.

AWS CLI

Um zwei Zweige mithilfe der Squash-Merge-Strategie zusammenzuführen

Im folgenden `merge-branches-by-squash` Beispiel wird der angegebene Quellzweig mit dem angegebenen Zielzweig in einem Repository mit dem Namen `zusammengeführt`. `MyDemoRepo`

```
aws codecommit merge-branches-by-squash \  
--source-commit-specifier bugfix-bug1234 \  
--destination-commit-specifier bugfix-bug1233 \  
--author-name "Maria Garcia" \  
--email "maria_garcia@example.com" \  
--commit-message "Merging two fix branches to prepare for a general patch." \  
--repository-name MyDemoRepo
```

Ausgabe:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Branches vergleichen und zusammenführen](#).AWS CodeCommit

- API-Einheiten finden Sie [MergeBranchesBySquash](#) in der AWS CLI Befehlsreferenz.

merge-branches-by-three-way

Das folgende Codebeispiel zeigt die Verwendung `merge-branches-by-three-way`.

AWS CLI

Um zwei Zweige mithilfe der Drei-Wege-Merge-Strategie zusammenzuführen

Im folgenden `merge-branches-by-three-way` Beispiel wird der angegebene Quellzweig mit dem angegebenen Zielzweig in einem Repository mit dem Namen `zusammengeführt`.

`MyDemoRepo`

```
aws codecommit merge-branches-by-three-way \
  --source-commit-specifier main \
  --destination-commit-specifier bugfix-bug1234 \
  --author-name "Jorge Souza" --email "jorge_souza@example.com" \
  --commit-message "Merging changes from main to bugfix branch before additional testing." \
  --repository-name MyDemoRepo
```

Ausgabe:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Branches vergleichen und zusammenführen](#).AWS CodeCommit

- API-Einheiten finden Sie [MergeBranchesByThreeWay](#) in der AWS CLI Befehlsreferenz.

merge-pull-request-by-fast-forward

Das folgende Codebeispiel zeigt die Verwendung `merge-pull-request-by-fast-forward`.

AWS CLI

Um eine Pull-Anfrage zusammenzuführen und zu schließen

Dieses Beispiel zeigt, wie eine Pull-Anfrage mit der ID '47' und der Quell-Commit-ID '99132ab0EXAMPLE' in einem Repository mit dem Namen zusammengeführt und geschlossen wird. MyDemoRepo

```
aws codecommit merge-pull-request-by-fast-forward \  
  --pull-request-id 47 \  
  --source-commit-id 99132ab0EXAMPLE \  
  --repository-name MyDemoRepo
```

Ausgabe:

```
{  
  "pullRequest": {  
    "approvalRules": [  
      {  
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":  
[{\n\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":  
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",  
        "approvalRuleId": "dd8b17fe-EXAMPLE",  
        "approvalRuleName": "I want one approver for this pull request",  
        "creationDate": 1571356106.936,  
        "lastModifiedDate": 571356106.936,  
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
        "ruleContentSha256": "4711b576EXAMPLE"  
      }  
    ],  
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",  
    "clientRequestToken": "",  
    "creationDate": 1508530823.142,  
    "description": "Review the latest changes and updates to the global  
variables",  
    "lastActivityDate": 1508887223.155,  
    "pullRequestId": "47",  
    "pullRequestStatus": "CLOSED",  
    "pullRequestTargets": [  
      {  
        "destinationCommit": "9f31c968EXAMPLE",  
        "destinationReference": "refs/heads/main",  
        "mergeMetadata": {
```

```

        "isMerged": true,
        "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
    },
    "repositoryName": "MyDemoRepo",
    "sourceCommit": "99132ab0EXAMPLE",
    "sourceReference": "refs/heads/variables-branch"
  }
],
"title": "Consolidation of global variables"
}
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Einen Pull-Request zusammenführen](#).AWS CodeCommit

- API-Einheiten finden Sie [MergePullRequestByFastForward](#) in der AWS CLI Befehlsreferenz.

merge-pull-request-by-squash

Das folgende Codebeispiel zeigt die Verwendung `merge-pull-request-by-squash`.

AWS CLI

Um eine Pull-Anfrage mithilfe der Squash-Merge-Strategie zusammenzuführen

Im folgenden `merge-pull-request-by-squash` Beispiel wird die angegebene Pull-Anfrage unter Verwendung der Konfliktlösungsstrategie `ACCEPT_SOURCE` in einem Repository mit dem Namen `MyDemoRepo` zusammengeführt und geschlossen.

```

aws codecommit merge-pull-request-by-squash \
  --pull-request-id 47 \
  --source-commit-id 99132ab0EXAMPLE \
  --repository-name MyDemoRepo \
  --conflict-detail-level LINE_LEVEL \
  --conflict-resolution-strategy ACCEPT_SOURCE \
  --name "Jorge Souza" --email "jorge_souza@example.com" \
  --commit-message "Merging pull request 47 by squash and accepting source in
merge conflicts"

```

Ausgabe:

```

{
  "pullRequest": {

```

```

    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\\\"DestinationReferences\": [\\\"refs/heads/main\\\"],\\\"Statements\": [{\\\"Type
\\\": \\\"Approvers\\\",\\\"NumberOfApprovalsNeeded\": 2,\\\"ApprovalPoolMembers\":
[\\\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\\\"]}]}\",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
          "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
          "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.142,
    "description": "Review the latest changes and updates to the global
variables",
    "lastActivityDate": 1508887223.155,
    "pullRequestId": "47",
    "pullRequestStatus": "CLOSED",
    "pullRequestTargets": [
      {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": true,
          "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
      }
    ],
    "title": "Consolidation of global variables"
  }
}

```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Einen Pull-Request zusammenführen](#).

- API-Einheiten finden Sie [MergePullRequestBySquash](#) in der AWS CLI Befehlsreferenz.

merge-pull-request-by-three-way

Das folgende Codebeispiel zeigt die Verwendung `merge-pull-request-by-three-way`.

AWS CLI

Um eine Pull-Anfrage mithilfe der Drei-Wege-Merge-Strategie zusammenzuführen

Im folgenden `merge-pull-request-by-three-way` Beispiel wird die angegebene Pull-Anfrage unter Verwendung der Standardoptionen für Konfliktdetails und Konfliktlösungsstrategie in einem Repository mit dem Namen `zusammengeführt` und geschlossen. `MyDemoRepo`

```
aws codecommit merge-pull-request-by-three-way \
  --pull-request-id 47 \
  --source-commit-id 99132ab0EXAMPLE \
  --repository-name MyDemoRepo \
  --name "Maria Garcia" \
  --email "maria_garcia@example.com" \
  --commit-message "Merging pull request 47 by three-way with default options"
```

Ausgabe:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
        \"DestinationReferences\": [\"refs/heads/main\"], \"Statements\": [{\"Type
        \": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
        [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
          "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
```

```

        "approvalRuleTemplateName": "2-approver-rule-for-main"
      },
      "ruleContentSha256": "4711b576EXAMPLE"
    }
  ],
  "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
  "clientRequestToken": "",
  "creationDate": 1508530823.142,
  "description": "Review the latest changes and updates to the global
variables",
  "lastActivityDate": 1508887223.155,
  "pullRequestId": "47",
  "pullRequestStatus": "CLOSED",
  "pullRequestTargets": [
    {
      "destinationCommit": "9f31c968EXAMPLE",
      "destinationReference": "refs/heads/main",
      "mergeMetadata": {
        "isMerged": true,
        "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
      },
      "repositoryName": "MyDemoRepo",
      "sourceCommit": "99132ab0EXAMPLE",
      "sourceReference": "refs/heads/variables-branch"
    }
  ],
  "title": "Consolidation of global variables"
}

```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Einen Pull-Request zusammenführen](#).

- API-Einheiten finden Sie [MergePullRequestByThreeWay](#) in der AWS CLI Befehlsreferenz.

override-pull-request-approval-rules

Das folgende Codebeispiel zeigt die Verwendung `override-pull-request-approval-rules`.

AWS CLI

Um die Anforderungen der Genehmigungsregeln für eine Pull-Anfrage zu überschreiben

Im folgenden `override-pull-request-approval-rules` Beispiel werden die Genehmigungsregeln für die angegebene Pull-Anfrage außer Kraft gesetzt. Um stattdessen eine Überschreibung zu widerrufen, setzen Sie den `--override-status` Parameterwert auf `REVOKE`.

```
aws codecommit override-pull-request-approval-rules \  
  --pull-request-id 34 \  
  --revision-id 927df8d8EXAMPLE \  
  --override-status OVERRIDE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Überschreiben von Genehmigungsregeln bei einem Pull Request](#).

- API-Einheiten finden Sie [OverridePullRequestApprovalRules](#) unter AWS CLI Befehlsreferenz.

post-comment-for-compared-commit

Das folgende Codebeispiel zeigt die Verwendung `post-comment-for-compared-commit`.

AWS CLI

Um einen Kommentar zu einem Commit zu erstellen

Dieses Beispiel zeigt, wie der Kommentar zur Änderung "Can you add a test case for this?" zur `cl_sample.js` Datei hinzugefügt wird, wenn zwei Commits in einem Repository mit dem Namen `MyDemoRepo` verglichen werden.

```
aws codecommit post-comment-for-compared-commit \  
  --repository-name MyDemoRepo \  
  --before-commit-id 317f8570EXAMPLE \  
  --after-commit-id 5d036259EXAMPLE \  
  --client-request-token 123Example \  
  --content "Can you add a test case for this?" \  
  --location filePath=cl_sample.js,filePosition=1232,relativeFileVersion=AFTER
```

Ausgabe:

```
{  
  "afterBlobId": "1f330709EXAMPLE",  
  "afterCommitId": "317f8570EXAMPLE",  
  "beforeBlobId": "80906a4cEXAMPLE",
```

```

"beforeCommitId": "6e147360EXAMPLE",
"comment": {
  "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
  "clientRequestToken": "",
  "commentId": "553b509bEXAMPLE56198325",
  "content": "Can you add a test case for this?",
  "creationDate": 1508369612.203,
  "deleted": false,
  "commentId": "abc123-EXAMPLE",
  "lastModifiedDate": 1508369612.203,
  "callerReactions": [],
  "reactionCounts": []
},
"location": {
  "filePath": "cl_sample.js",
  "filePosition": 1232,
  "relativeFileVersion": "AFTER"
,
"repositoryName": "MyDemoRepo"
}
}

```

- API-Einheiten finden Sie [PostCommentForComparedCommit](#) in der AWS CLI Befehlsreferenz.

post-comment-for-pull-request

Das folgende Codebeispiel zeigt die Verwendung `post-comment-for-pull-request`.

AWS CLI

Um einen Kommentar zu einer Pull-Anfrage hinzuzufügen

Das folgende `post-comment-for-pull-request` Beispiel fügt den Kommentar hinzu: „Diese scheinen nirgends verwendet zu werden. Können wir sie entfernen?“ über die Änderung an der `ahs_count.py` Datei in einem Pull-Request mit der ID von 47 in einem Repository namens `MyDemoRepo`.

```

aws codecommit post-comment-for-pull-request \
  --pull-request-id "47" \
  --repository-name MyDemoRepo \
  --before-commit-id 317f8570EXAMPLE \
  --after-commit-id 5d036259EXAMPLE \

```

```
--client-request-token 123Example \  
--content "These don't appear to be used anywhere. Can we remove them?" \  
--location filePath=ahs_count.py,filePosition=367,relativeFileVersion=AFTER
```

Ausgabe:

```
{  
  "afterBlobId": "1f330709EXAMPLE",  
  "afterCommitId": "5d036259EXAMPLE",  
  "beforeBlobId": "80906a4cEXAMPLE",  
  "beforeCommitId": "317f8570EXAMPLE",  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",  
    "clientRequestToken": "123Example",  
    "commentId": "abcd1234EXAMPLEeb5678efgh",  
    "content": "These don't appear to be used anywhere. Can we remove  
them?",  
    "creationDate": 1508369622.123,  
    "deleted": false,  
    "CommentId": "",  
    "lastModifiedDate": 1508369622.123,  
    "callerReactions": [],  
    "reactionCounts": []  
  },  
  "location": {  
    "filePath": "ahs_count.py",  
    "filePosition": 367,  
    "relativeFileVersion": "AFTER"  
  },  
  "repositoryName": "MyDemoRepo",  
  "pullRequestId": "47"  
}
```

- API-Einheiten finden Sie [PostCommentForPullRequest](#) in der AWS CLI Befehlsreferenz.

post-comment-reply

Das folgende Codebeispiel zeigt die Verwendung `post-comment-reply`.

AWS CLI

Um auf einen Kommentar zu einem Commit oder in einer Pull-Anfrage zu antworten

Dieses Beispiel zeigt, wie die Antwort "Good catch. I'll remove them." auf den Kommentar mit der vom System generierten ID von hinzugefügt wird.
abcd1234EXAMPLEb5678efgh

```
aws codecommit post-comment-reply \  
  --in-reply-to abcd1234EXAMPLEb5678efgh \  
  --content "Good catch. I'll remove them." \  
  --client-request-token 123Example
```

Ausgabe:

```
{  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "123Example",  
    "commentId": "442b498bEXAMPLE5756813",  
    "content": "Good catch. I'll remove them.",  
    "creationDate": 1508369829.136,  
    "deleted": false,  
    "CommentId": "abcd1234EXAMPLEb5678efgh",  
    "lastModifiedDate": 150836912.221,  
    "callerReactions": [],  
    "reactionCounts": []  
  }  
}
```

- API-Einheiten finden Sie unter [PostCommentReply AWS CLI Befehlsreferenz](#).

put-comment-reaction

Das folgende Codebeispiel zeigt die Verwendung `put-comment-reaction`.

AWS CLI

Um auf einen Kommentar zu einem Commit mit einem Emoji zu antworten

Das folgende `put-comment-reaction` Beispiel antwortet auf einen Kommentar mit der ID von `abcd1234EXAMPLEb5678efgh` mit dem Emoji-Reaktionswert von `:thumbsup:`

```
aws codecommit put-comment-reaction \  
  --comment-id abcd1234EXAMPLEb5678efgh \  
  --reaction :thumbsup:
```

```
--comment-id abcd1234EXAMPLEb5678efgh \  
--reaction-value :thumbsup:
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch [unter Kommentieren zu einem Commit AWS CodeCommit in](#).

- API-Einheiten finden Sie [PutCommentReaction](#) in der AWS CLI Befehlsreferenz.

put-file

Das folgende Codebeispiel zeigt die Verwendung `put-file`.

AWS CLI

Um eine Datei zu einem Repository hinzuzufügen

Im folgenden `put-file` Beispiel wird eine Datei mit dem Namen 'ExampleSolution.py' zu einem Repository namens " zu einem Branch namens MyDemoRepo 'feature-randomizationfeature' hinzugefügt, dessen letzter Commit die ID '4c925148' hat. EXAMPLE

```
aws codecommit put-file \  
  --repository-name MyDemoRepo \  
  --branch-name feature-randomizationfeature \  
  --file-content file://MyDirectory/ExampleSolution.py \  
  --file-path /solutions/ExampleSolution.py \  
  --parent-commit-id 4c925148EXAMPLE \  
  --name "Maria Garcia" \  
  --email "maria_garcia@example.com" \  
  --commit-message "I added a third randomization routine."
```

Ausgabe:

```
{  
  "blobId": "2eb4af3bEXAMPLE",  
  "commitId": "317f8570EXAMPLE",  
  "treeId": "347a3408EXAMPLE"  
}
```

- [PutFile](#) Einheiten API AWS CLI finden Sie unter Befehlsreferenz.

put-repository-triggers

Das folgende Codebeispiel zeigt die Verwendung `put-repository-triggers`.

AWS CLI

Um einen Trigger in einem Repository hinzuzufügen oder zu aktualisieren

Dieses Beispiel zeigt, wie Trigger mit den Namen " " und `MyFirstTrigger` `'MySecondTrigger'` mithilfe einer bereits erstellten JSON Datei (hier mit dem Namen „ `MyTriggers.json`“) aktualisiert werden, die die Struktur aller Trigger für ein Repository mit dem Namen enthält. `MyDemoRepo` Informationen zum Abrufen der JSON vier vorhandenen Trigger finden Sie im Befehl. `get-repository-triggers`

```
aws codecommit put-repository-triggers \  
  --repository-name MyDemoRepo file://MyTriggers.json
```

Inhalt von `MyTriggers.json`:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": [  
    {  
      "destinationArn": "arn:aws:sns:us-  
east-1:80398EXAMPLE:MyCodeCommitTopic",  
      "branches": [  
        "main",  
        "preprod"  
      ],  
      "name": "MyFirstTrigger",  
      "customData": "",  
      "events": [  
        "all"  
      ]  
    },  
    {  
      "destinationArn": "arn:aws:lambda:us-  
east-1:111111111111:function:MyCodeCommitPythonFunction",  
      "branches": [],  
      "name": "MySecondTrigger",  
      "customData": "EXAMPLE",  
      "events": [  

```

```

    "all"
  ]
}

```

Ausgabe:

```

{
  "configurationId": "6fa51cd8-35c1-EXAMPLE"
}

```

- API-Einheiten finden Sie [PutRepositoryTriggers](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um AWS Tags zu einem vorhandenen Repository hinzuzufügen

Im folgenden `tag-resource` Beispiel wird das angegebene Repository mit zwei Tags versehen.

```

aws codecommit tag-resource \
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
  --tags Status=Secret,Team=Saanvi

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen eines Tags zu einem Repository](#) im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

test-repository-triggers

Das folgende Codebeispiel zeigt die Verwendung `test-repository-triggers`.

AWS CLI

Um Trigger in einem Repository zu testen

Dieses Beispiel zeigt, wie ein Trigger mit dem Namen 'MyFirstTrigger' in einem AWS CodeCommit Repository mit dem Namen getestet wird MyDemoRepo. In diesem Beispiel lösen Ereignisse im Repository Benachrichtigungen von einem Amazon Simple Notification Service (AmazonSNS) - Thema aus.

Befehl:

```
aws codecommit test-repository-triggers --repository-name MyDemoRepo
--triggers name=MyFirstTrigger,destinationArn=arn:aws:sns:us-
east-1:111111111111:MyCodeCommitTopic,branches=mainline,preprod,events=all
```

Ausgabe:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- API-Einheiten finden Sie [TestRepositoryTriggers](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um AWS Tags aus einem Repository zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem angegebenen Schlüssel aus dem Repository mit dem Namen entfernt MyDemoRepo.

```
aws codecommit untag-resource \
--resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
--tag-keys Status
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Entfernen eines Tags aus einem Repository](#) im AWS CodeCommit Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-approval-rule-template-content

Das folgende Codebeispiel zeigt die Verwendung `update-approval-rule-template-content`.

AWS CLI

Um den Inhalt einer Vorlage für Genehmigungsregeln zu aktualisieren

Im folgenden `update-approval-rule-template-content` Beispiel wird der Inhalt der angegebenen Genehmigungsregelvorlage geändert, um den Genehmigungsprozess für Benutzer neu zu definieren, die die Rolle von `CodeCommitReview` übernehmen.

```
aws codecommit update-approval-rule-template-content \
  --approval-rule-template-name 1-approver-rule \
  --new-rule-content "{\"Version\": \"2018-11-08\", \"DestinationReferences\": [\"refs/heads/main\"], \"Statements\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

Ausgabe:

```
{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull requests from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#).

- API-Einheiten finden Sie [UpdateApprovalRuleTemplateContent](#) in der AWS CLI Befehlsreferenz.

update-approval-rule-template-description

Das folgende Codebeispiel zeigt die Verwendung `update-approval-rule-template-description`.

AWS CLI

Um die Beschreibung einer Vorlage für Genehmigungsregeln zu aktualisieren

Im folgenden `update-approval-rule-template-description` Beispiel wird die Beschreibung der angegebenen Vorlage für Genehmigungsregeln in geändert `Requires 1 approval for all pull requests from the CodeCommitReview pool`.

```
aws codecommit update-approval-rule-template-description \
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests \
  --approval-rule-template-description "Requires 1 approval for all pull requests
  from the CodeCommitReview pool"
```

Ausgabe:

```
{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull requests
    from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\":
    [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":
    [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}
```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#).

- API-Einheiten finden Sie [UpdateApprovalRuleTemplateDescription](#) in der AWS CLI Befehlsreferenz.

update-approval-rule-template-name

Das folgende Codebeispiel zeigt die Verwendung `update-approval-rule-template-name`.

AWS CLI

Um den Namen einer Vorlage für Genehmigungsregeln zu aktualisieren

Im folgenden `update-approval-rule-template-name` Beispiel wird der Name einer Vorlage für Genehmigungsregeln von `1-1-approver-rule` `approver-rule-for-all -Pull-Requests` geändert.

```
aws codecommit update-approval-rule-template-name \
  --old-approval-rule-template-name 1-approver-rule \
  --new-approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Ausgabe:

```
{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedDate": 1571358241.619,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\": [
    {\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\": [
    \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "creationDate": 1571352720.773,
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "approvalRuleTemplateDescription": "All pull requests must be approved by one
    developer on the team.",
    "ruleContentSha256": "2f6c21a5cEXAMPLE"
  }
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Vorlagen für Genehmigungsregeln verwalten](#). AWS CodeCommit

- API-Einheiten finden Sie [UpdateApprovalRuleTemplateName](#) in der AWS CLI Befehlsreferenz.

update-comment

Das folgende Codebeispiel zeigt die Verwendung `update-comment`.

AWS CLI

Um einen Kommentar zu einem Commit zu aktualisieren

Dieses Beispiel zeigt, wie der Inhalt `"Fixed as requested. I'll update the pull request."` zu einem Kommentar mit der ID hinzugefügt wird `442b498bEXAMPLE5756813`.

```
aws codecommit update-comment \  
  --comment-id 442b498bEXAMPLE5756813 \  
  --content "Fixed as requested. I'll update the pull request."
```

Ausgabe:

```
{  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "",  
    "commentId": "442b498bEXAMPLE5756813",  
    "content": "Fixed as requested. I'll update the pull request.",  
    "creationDate": 1508369929.783,  
    "deleted": false,  
    "lastModifiedDate": 1508369929.287,  
    "callerReactions": [],  
    "reactionCounts":  
      {  
        "THUMBSUP" : 2  
      }  
  }  
}
```

- API-Einheiten finden Sie [UpdateComment](#) in der AWS CLI Befehlsreferenz.

update-default-branch

Das folgende Codebeispiel zeigt die Verwendung `update-default-branch`.

AWS CLI

Um den Standardzweig für ein Repository zu ändern

In diesem Beispiel wird der Standardzweig für ein AWS CodeCommit Repository geändert. Dieser Befehl liefert nur eine Ausgabe, wenn Fehler aufgetreten sind.

Befehl:

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [UpdateDefaultBranch](#) in der AWS CLI Befehlsreferenz.

update-pull-request-approval-rule-content

Das folgende Codebeispiel zeigt die Verwendung `update-pull-request-approval-rule-content`.

AWS CLI

Um eine Genehmigungsregel für eine Pull-Anfrage zu bearbeiten

Im folgenden `update-pull-request-approval-rule-content` Beispiel wird die von ihr angegebene Genehmigungsregel dahingehend aktualisiert, dass eine Benutzergenehmigung aus einem Genehmigungs-pool erforderlich ist, der alle IAM Benutzer des 123456789012 AWS Kontos umfasst.

```
aws codecommit update-pull-request-approval-rule-content \  
  --pull-request-id 27 \  
  --approval-rule-name "Require two approved approvers" \  
  --approval-rule-content "{Version: 2018-11-08, Statements: [{Type: \  
  \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers: \  
  [\"CodeCommitApprovers:123456789012:user/*\"]}]}"
```

Ausgabe:

```
{  
  "approvalRule": {
```

```

    "approvalRuleContent": "{Version: 2018-11-08, Statements:
  [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
  [\"CodeCommitApprovers:123456789012:user/*\"]}]}",
    "approvalRuleId": "aac33506-EXAMPLE",
    "originApprovalRuleTemplate": {},
    "creationDate": 1570752871.932,
    "lastModifiedDate": 1570754058.333,
    "approvalRuleName": "Require two approved approvers",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "ruleContentSha256": "cd93921cEXAMPLE",
  }
}

```

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Bearbeiten oder Löschen einer Genehmigungsregel](#).

- API-Einheiten finden Sie [UpdatePullRequestApprovalRuleContent](#) in der AWS CLI Befehlsreferenz.

update-pull-request-approval-state

Das folgende Codebeispiel zeigt die Verwendung `update-pull-request-approval-state`.

AWS CLI

Um die Genehmigung für einen Pull-Request zu genehmigen oder zu widerrufen

Im folgenden `update-pull-request-approval-state` Beispiel wird eine Pull-Anfrage mit der ID von 27 und der Revisions-ID von 9f29d167EXAMPLE genehmigt. Wenn Sie stattdessen die Genehmigung widerrufen möchten, setzen Sie den `--approval-state` Parameterwert auf `REVOKE`.

```

aws codecommit update-pull-request-approval-state \
  --pull-request-id 27 \
  --revision-id 9f29d167EXAMPLE \
  --approval-state "APPROVE"

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Einen Pull-Request überprüfen](#).

- API-Einheiten finden Sie [UpdatePullRequestApprovalState](#) in der AWS CLI Befehlsreferenz.

update-pull-request-description

Das folgende Codebeispiel zeigt die Verwendung `update-pull-request-description`.

AWS CLI

Um die Beschreibung einer Pull-Anfrage zu ändern

Dieses Beispiel zeigt, wie die Beschreibung einer Pull-Anfrage mit der ID von geändert wird 47.

```
aws codecommit update-pull-request-description \  
  --pull-request-id 47 \  
  --description "Updated the pull request to remove unused global variable."
```

Ausgabe:

```
{  
  "pullRequest": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "",  
    "creationDate": 1508530823.155,  
    "description": "Updated the pull request to remove unused global variable.",  
    "lastActivityDate": 1508372423.204,  
    "pullRequestId": "47",  
    "pullRequestStatus": "OPEN",  
    "pullRequestTargets": [  
      {  
        "destinationCommit": "9f31c968EXAMPLE",  
        "destinationReference": "refs/heads/main",  
        "mergeMetadata": {  
          "isMerged": false,  
        },  
        "repositoryName": "MyDemoRepo",  
        "sourceCommit": "99132ab0EXAMPLE",  
        "sourceReference": "refs/heads/variables-branch"  
      }  
    ],  
    "title": "Consolidation of global variables"  
  }  
}
```

- API-Einheiten finden Sie [UpdatePullRequestDescription](#) unter AWS CLI Befehlsreferenz.

update-pull-request-status

Das folgende Codebeispiel zeigt die Verwendung `update-pull-request-status`.

AWS CLI

Um den Status einer Pull-Anfrage zu ändern

Dieses Beispiel zeigt, wie der Status einer Pull-Anfrage mit der ID von 42 in den Status `CLOSED` in einem AWS CodeCommit Repository mit dem Namen geändert werden kann `MyDemoRepo`.

```
aws codecommit update-pull-request-status \  
  --pull-request-id 42 \  
  --pull-request-status CLOSED
```

Ausgabe:

```
{  
  "pullRequest": {  
    "approvalRules": [  
      {  
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":  
[{\n\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":  
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",  
        "approvalRuleId": "dd8b17fe-EXAMPLE",  
        "approvalRuleName": "2-approvers-needed-for-this-change",  
        "creationDate": 1571356106.936,  
        "lastModifiedDate": 571356106.936,  
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
        "ruleContentSha256": "4711b576EXAMPLE"  
      }  
    ],  
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",  
    "clientRequestToken": "",  
    "creationDate": 1508530823.165,  
    "description": "Updated the pull request to remove unused global variable.",  
    "lastActivityDate": 1508372423.12,  
    "pullRequestId": "47",  
    "pullRequestStatus": "CLOSED",  
    "pullRequestTargets": [  
      {
```



```

        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
            "isMerged": false,
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
    }
],
    "title": "Consolidation of global variables"
}
}

```

- API-Einheiten finden Sie [UpdatePullRequestStatus](#) in der AWS CLI Befehlsreferenz.

update-pull-request-title

Das folgende Codebeispiel zeigt die Verwendung `update-pull-request-title`.

AWS CLI

Um den Titel einer Pull-Anfrage zu ändern

Dieses Beispiel zeigt, wie der Titel einer Pull-Anfrage mit der ID von geändert wird⁴⁷.

```

aws codecommit update-pull-request-title \
  --pull-request-id 47 \
  --title "Consolidation of global variables - updated review"

```

Ausgabe:

```

{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
        \"DestinationReferences\": [\"refs/heads/main\"], \"Statements\": [{\"Type
        \": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
        [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,

```

```

        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
            "approvalRuleTemplateId": "dd8b26gr-EXAMPLE",
            "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
    },
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.12,
    "description": "Review the latest changes and updates to the global
variables. I have updated this request with some changes, including removing some
unused variables.",
    "lastActivityDate": 1508372657.188,
    "pullRequestId": "47",
    "pullRequestStatus": "OPEN",
    "pullRequestTargets": [
        {
            "destinationCommit": "9f31c968EXAMPLE",
            "destinationReference": "refs/heads/main",
            "mergeMetadata": {
                "isMerged": false,
            },
            "repositoryName": "MyDemoRepo",
            "sourceCommit": "99132ab0EXAMPLE",
            "sourceReference": "refs/heads/variables-branch"
        }
    ],
    "title": "Consolidation of global variables - updated review"
}
}

```

- API-Einheiten finden Sie [UpdatePullRequestTitle](#) unter AWS CLI Befehlsreferenz.

update-repository-description

Das folgende Codebeispiel zeigt die Verwendung `update-repository-description`.

AWS CLI

Um die Beschreibung für ein Repository zu ändern

In diesem Beispiel wird die Beschreibung für ein AWS CodeCommit Repository geändert. Dieser Befehl liefert nur eine Ausgabe, wenn Fehler aufgetreten sind.

Befehl:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --  
repository-description "This description was changed"
```

Ausgabe:

```
None .
```

- API-Einheiten finden Sie [UpdateRepositoryDescription](#) in der AWS CLI Befehlsreferenz.

update-repository-name

Das folgende Codebeispiel zeigt die Verwendung `update-repository-name`.

AWS CLI

Um den Namen eines Repositories zu ändern

In diesem Beispiel wird der Name eines AWS CodeCommit Repositories geändert. Dieser Befehl liefert nur eine Ausgabe, wenn Fehler aufgetreten sind. Wenn Sie den Namen des AWS CodeCommit Repositories ändern, ändert sich auch der SSH und HTTPS URLs der Benutzer, um eine Verbindung mit dem Repository herzustellen, benötigen. Benutzer können erst eine Verbindung mit diesem Repository herstellen, wenn sie ihre Verbindungseinstellungen aktualisiert haben. Da sich der Name des Repositories ändern wird, werden durch die Änderung des Repository-Namens auch alle IAM Benutzerrichtlinien ungültig, die sich auf die Richtlinien dieses Repositories ARN stützen.

Befehl:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-  
name MyRenamedDemoRepo
```

Ausgabe:

```
None .
```

- API-Einheiten finden Sie [UpdateRepositoryName](#) in der AWS CLI Befehlsreferenz.

CodeDeploy Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CodeDeploy.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-on-premises-instances

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-on-premises-instances`.

AWS CLI

Um Tags zu lokalen Instanzen hinzuzufügen

Im folgenden `add-tags-to-on-premises-instances` Beispiel wird AWS CodeDeploy dasselbe lokale Instanz-Tag zwei lokalen Instanzen zugeordnet. Die lokalen Instanzen werden nicht bei registriert. AWS CodeDeploy

```
aws deploy add-tags-to-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AddTagsToOnPremisesInstances](#) in der AWS CLI Befehlsreferenz.

batch-get-application-revisions

Das folgende Codebeispiel zeigt die Verwendung `batch-get-application-revisions`.

AWS CLI

Um Informationen über Anwendungsrevisionen abzurufen

Im folgenden `batch-get-application-revisions` Beispiel werden Informationen über die angegebene Version abgerufen, die in einem GitHub Repository gespeichert ist.

```
aws deploy batch-get-application-revisions \
  --application-name my-codedeploy-application \
  --revisions "[{\\"githubLocation\\": {\\"commitId\\":
  \\"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\\",\\"repository\\": \\"my-github-token/my-
  repository\\"},\\"revisionType\\": \\"GitHub\\"}]"
```

Ausgabe:

```
{
  "revisions": [
    {
      "genericRevisionInfo": {
        "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
        "lastUsedTime": 1556912355.884,
        "registerTime": 1556912355.884,
        "firstUsedTime": 1556912355.884,
        "deploymentGroups": []
      },
      "revisionLocation": {
        "revisionType": "GitHub",
        "githubLocation": {
          "commitId": "fa85936EXAMPLEa31736c051f10d77297EXAMPLE",
          "repository": "my-github-token/my-repository"
        }
      }
    }
  ],
  "applicationName": "my-codedeploy-application",
  "errorMessage": ""
}
```

Weitere Informationen finden Sie [BatchGetApplicationRevisions](#) in der AWS CodeDeploy APIReferenz.

- APIEinzelheiten finden Sie [BatchGetApplicationRevisions](#) in der AWS CLI Befehlsreferenz.

batch-get-applications

Das folgende Codebeispiel zeigt die Verwendung `batch-get-applications`.

AWS CLI

Um Informationen über mehrere Anwendungen zu erhalten

Im folgenden `batch-get-applications` Beispiel werden Informationen zu mehreren Anwendungen angezeigt, die dem AWS Konto des Benutzers zugeordnet sind.

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

Ausgabe:

```
{
  "applicationsInfo": [
    {
      "applicationName": "WordPress_App",
      "applicationId": "d9dd6993-f171-44fa-a811-211e4EXAMPLE",
      "createTime": 1407878168.078,
      "linkedToGitHub": false
    },
    {
      "applicationName": "MyOther_App",
      "applicationId": "8ca57519-31da-42b2-9194-8bb16EXAMPLE",
      "createTime": 1407453571.63,
      "linkedToGitHub": false
    }
  ]
}
```

- APIEinzelheiten finden Sie [BatchGetApplications](#) unter AWS CLI Befehlsreferenz.

batch-get-deployment-groups

Das folgende Codebeispiel zeigt die Verwendung `batch-get-deployment-groups`.

AWS CLI

Um Informationen über eine oder mehrere Bereitstellungsgruppen abzurufen

Im folgenden `batch-get-deployment-groups` Beispiel werden Informationen zu zwei der Bereitstellungsgruppen abgerufen, die der angegebenen CodeDeploy Anwendung zugeordnet sind.

```
aws deploy batch-get-deployment-groups \  
  --application-name my-codedeploy-application \  
  --deployment-group-names ["my-deployment-group-1","my-deployment-group-2"]
```

Ausgabe:

```
{  
  "deploymentGroupsInfo": [  
    {  
      "deploymentStyle": {  
        "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",  
        "deploymentType": "IN_PLACE"  
      },  
      "autoRollbackConfiguration": {  
        "enabled": false  
      },  
      "onPremisesTagSet": {  
        "onPremisesTagSetList": []  
      },  
      "serviceRoleArn": "arn:aws:iam::123456789012:role/  
CodeDeployServiceRole",  
      "lastAttemptedDeployment": {  
        "endTime": 1556912366.415,  
        "status": "Failed",  
        "createTime": 1556912355.884,  
        "deploymentId": "d-A1B2C3111"  
      },  
      "autoScalingGroups": [],  
      "deploymentGroupName": "my-deployment-group-1",  
      "ec2TagSet": {  
        "ec2TagSetList": [  
          [  
            {  
              "Type": "KEY_AND_VALUE",  
              "Value": "my-EC2-instance",
```

```

        "Key": "Name"
      }
    ]
  ],
  "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111example",
  "triggerConfigurations": [],
  "applicationName": "my-codedeploy-application",
  "computePlatform": "Server",
  "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
},
{
  "deploymentStyle": {
    "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
    "deploymentType": "IN_PLACE"
  },
  "autoRollbackConfiguration": {
    "enabled": false
  },
  "onPremisesTagSet": {
    "onPremisesTagSetList": []
  },
  "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
  "autoScalingGroups": [],
  "deploymentGroupName": "my-deployment-group-2",
  "ec2TagSet": {
    "ec2TagSetList": [
      [
        {
          "Type": "KEY_AND_VALUE",
          "Value": "my-EC2-instance",
          "Key": "Name"
        }
      ]
    ]
  },
  "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-22222example",
  "triggerConfigurations": [],
  "applicationName": "my-codedeploy-application",
  "computePlatform": "Server",
  "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
}
],

```



```
"errorMessage": ""
}
```

Weitere Informationen finden Sie [BatchGetDeploymentGroups](#) in der AWS CodeDeploy APIReferenz.

- APIEinzelheiten finden Sie [BatchGetDeploymentGroups](#) in der AWS CLI Befehlsreferenz.

batch-get-deployment-targets

Das folgende Codebeispiel zeigt die Verwendung `batch-get-deployment-targets`.

AWS CLI

Um die mit einer Bereitstellung verknüpften Ziele abzurufen

Das folgende `batch-get-deployment-targets` Beispiel gibt Informationen über eines der Ziele zurück, die mit der angegebenen Bereitstellung verknüpft sind.

```
aws deploy batch-get-deployment-targets \
  --deployment-id "d-1A2B3C4D5" \
  --target-ids "i-01a2b3c4d5e6f1111"
```

Ausgabe:

```
{
  "deploymentTargets": [
    {
      "deploymentTargetType": "InstanceTarget",
      "instanceTarget": {
        "lifecycleEvents": [
          {
            "startTime": 1556918592.162,
            "lifecycleEventName": "ApplicationStop",
            "status": "Succeeded",
            "endTime": 1556918592.247,
            "diagnostics": {
              "scriptName": "",
              "errorCode": "Success",
              "logTail": "",
              "message": "Succeeded"
            }
          }
        ]
      }
    }
  ]
}
```

```
    }
  },
  {
    "startTime": 1556918593.193,
    "lifecycleEventName": "DownloadBundle",
    "status": "Succeeded",
    "endTime": 1556918593.981,
    "diagnostics": {
      "scriptName": "",
      "errorCode": "Success",
      "logTail": "",
      "message": "Succeeded"
    }
  },
  {
    "startTime": 1556918594.805,
    "lifecycleEventName": "BeforeInstall",
    "status": "Succeeded",
    "endTime": 1556918681.807,
    "diagnostics": {
      "scriptName": "",
      "errorCode": "Success",
      "logTail": "",
      "message": "Succeeded"
    }
  }
],
"targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-01a2b3c4d5e6f1111",
"deploymentId": "d-1A2B3C4D5",
"lastUpdatedAt": 1556918687.504,
"targetId": "i-01a2b3c4d5e6f1111",
"status": "Succeeded"
}
]
}
```

Weitere Informationen finden Sie [BatchGetDeploymentTargets](#) in der AWS CodeDeploy APIReferenz.

- APIEinzelheiten finden Sie [BatchGetDeploymentTargets](#) in der AWS CLI Befehlsreferenz.

batch-get-deployments

Das folgende Codebeispiel zeigt die Verwendung `batch-get-deployments`.

AWS CLI

Um Informationen über mehrere Bereitstellungen zu erhalten

Im folgenden `batch-get-deployments` Beispiel werden Informationen zu mehreren Bereitstellungen angezeigt, die dem Konto des Benutzers AWS zugeordnet sind.

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

Ausgabe:

```
{
  "deploymentsInfo": [
    {
      "applicationName": "WordPress_App",
      "status": "Failed",
      "deploymentOverview": {
        "Failed": 0,
        "InProgress": 0,
        "Skipped": 0,
        "Succeeded": 1,
        "Pending": 0
      },
      "deploymentConfigName": "CodeDeployDefault.OneAtATime",
      "creator": "user",
      "deploymentGroupName": "WordPress_DG",
      "revision": {
        "revisionType": "S3",
        "s3Location": {
          "bundleType": "zip",
          "version": "uTecLusEXAMPLEFXtfUcyfV8bEXAMPLE",
          "bucket": "CodeDeployDemoBucket",
          "key": "WordPressApp.zip"
        }
      },
      "deploymentId": "d-A1B2C3111",
      "createTime": 1408480721.9,
      "completeTime": 1408480741.822
    },
  ],
}
```

```

    {
      "applicationName": "MyOther_App",
      "status": "Failed",
      "deploymentOverview": {
        "Failed": 1,
        "InProgress": 0,
        "Skipped": 0,
        "Succeeded": 0,
        "Pending": 0
      },
      "deploymentConfigName": "CodeDeployDefault.OneAtATime",
      "creator": "user",
      "errorInformation": {
        "message": "Deployment failed: Constraint default violated: No hosts
succeeded.",
        "code": "HEALTH_CONSTRAINTS"
      },
      "deploymentGroupName": "MyOther_DG",
      "revision": {
        "revisionType": "S3",
        "s3Location": {
          "bundleType": "zip",
          "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
          "bucket": "CodeDeployDemoBucket",
          "key": "MyOtherApp.zip"
        }
      },
      "deploymentId": "d-A1B2C3222",
      "createTime": 1409764576.589,
      "completeTime": 1409764596.101
    }
  ]
}

```

- API-Einheiten finden Sie unter [BatchGetDeployments AWS CLI-Befehlsreferenz](#).

batch-get-on-premises-instances

Das folgende Codebeispiel zeigt die Verwendung `batch-get-on-premises-instances`.

AWS CLI

Um Informationen über eine oder mehrere lokale Instanzen zu erhalten

Im folgenden `batch-get-on-premises-instances` Beispiel werden Informationen zu zwei lokalen Instanzen abgerufen.

```
aws deploy batch-get-on-premises-instances --instance-  
names AssetTag12010298EX AssetTag23121309EX
```

Ausgabe:

```
{  
  "instanceInfos": [  
    {  
      "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/  
AssetTag12010298EX",  
      "tags": [  
        {  
          "Value": "CodeDeployDemo-OnPrem",  
          "Key": "Name"  
        }  
      ],  
      "instanceName": "AssetTag12010298EX",  
      "registerTime": 1425579465.228,  
      "instanceArn": "arn:aws:codedeploy:us-west-2:123456789012:instance/  
AssetTag12010298EX_4IwLNI2Alh"  
    },  
    {  
      "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/  
AssetTag23121309EX",  
      "tags": [  
        {  
          "Value": "CodeDeployDemo-OnPrem",  
          "Key": "Name"  
        }  
      ],  
      "instanceName": "AssetTag23121309EX",  
      "registerTime": 1425595585.988,  
      "instanceArn": "arn:aws:codedeploy:us-west-2:80398EXAMPLE:instance/  
AssetTag23121309EX_PomUy64Was"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [BatchGetOnPremisesInstances AWS CLI Befehlsreferenz](#).

continue-deployment

Das folgende Codebeispiel zeigt die Verwendung `continue-deployment`.

AWS CLI

Um mit der Umleitung des Datenverkehrs zu beginnen, ohne auf den Ablauf einer bestimmten Wartezeit zu warten.

Im folgenden `continue-deployment` Beispiel wird mit der Umleitung des Datenverkehrs von Instances in der ursprünglichen Umgebung begonnen, die bereit sind, den Verkehr auf Instances in der Ersatzumgebung zu verlagern.

```
aws deploy continue-deployment \  
  --deployment-id "d-A1B2C3111" \  
  --deployment-wait-type "READY_WAIT"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [ContinueDeployment](#) in der AWS CodeDeploy APIReferenz.

- APIEinzelheiten finden Sie [ContinueDeployment](#) in der AWS CLI Befehlsreferenz.

create-application

Das folgende Codebeispiel zeigt die Verwendung `create-application`.

AWS CLI

Um eine Anwendung zu erstellen

Das folgende `create-application` Beispiel erstellt eine Anwendung und ordnet sie dem AWS Konto des Benutzers zu.

```
aws deploy create-application --application-name MyOther_App
```

Ausgabe:

```
{  
  "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- API-Einheiten finden Sie [CreateApplication](#) unter AWS CLI Befehlsreferenz.

create-deployment-config

Das folgende Codebeispiel zeigt die Verwendung `create-deployment-config`.

AWS CLI

Um eine benutzerdefinierte Bereitstellungs-Konfiguration zu erstellen

Im folgenden `create-deployment-config` Beispiel wird eine benutzerdefinierte Bereitstellungs-Konfiguration erstellt und sie dem AWS Benutzerkonto zugeordnet.

```
aws deploy create-deployment-config \  
  --deployment-config-name ThreeQuartersHealthy \  
  --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

Ausgabe:

```
{  
  "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- API-Einheiten finden Sie [CreateDeploymentConfig](#) unter AWS CLI Befehlsreferenz.

create-deployment-group

Das folgende Codebeispiel zeigt die Verwendung `create-deployment-group`.

AWS CLI

Um eine Bereitstellungsgruppe zu erstellen

Im folgenden `create-deployment-group` Beispiel wird eine Bereitstellungsgruppe erstellt und sie der angegebenen Anwendung und dem AWS Benutzerkonto zugeordnet.

```
aws deploy create-deployment-group \  
  --application-name WordPress_App \  
  --auto-scaling-groups CodeDeployDemo-ASG \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DG \  
  --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \  
  --
```

```
--service-role-arn arn:aws:iam::123456789012:role/CodeDeployDemoRole
```

Ausgabe:

```
{
  "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- API-Einheiten finden Sie [CreateDeploymentGroup](#) unter AWS CLI Befehlsreferenz.

create-deployment

Das folgende Codebeispiel zeigt die Verwendung `create-deployment`.

AWS CLI

Beispiel 1: Um eine CodeDeploy Bereitstellung mit der EC2 /On-Premises-Compute-Plattform zu erstellen

Das folgende `create-deployment` Beispiel erstellt eine Bereitstellung und ordnet sie dem Konto des Benutzers AWS zu.

```
aws deploy create-deployment \
  --application-name WordPress_App \
  --deployment-config-name CodeDeployDefault.OneAtATime \
  --deployment-group-name WordPress_DG \
  --description "My demo deployment" \
  --s3-
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

Ausgabe:

```
{
  "deploymentId": "d-A1B2C3111"
}
```

Beispiel 2: So erstellen Sie eine CodeDeploy Bereitstellung mithilfe der ECS Amazon-Rechenplattform

Das folgende `create-deployment` Beispiel verwendet die folgenden zwei Dateien, um einen ECS Amazon-Service bereitzustellen.

Inhalt der `create-deployment.json` Datei:

```
{
  "applicationName": "ecs-deployment",
  "deploymentGroupName": "ecs-deployment-dg",
  "revision": {
    "revisionType": "S3",
    "s3Location": {
      "bucket": "ecs-deployment-bucket",
      "key": "appspec.yaml",
      "bundleType": "YAML"
    }
  }
}
```

Diese Datei wiederum ruft die folgende Datei `appspec.yaml` aus einem S3-Bucket mit dem Namen `ecs-deployment-bucket` ab.

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:region:123456789012:task-definition/ecs-task-def:2"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80
          PlatformVersion: "LATEST"
```

Befehl:

```
aws deploy create-deployment \
  --cli-input-json file://create-deployment.json \
  --region us-east-1
```

Ausgabe:

```
{
  "deploymentId": "d-1234ABCDE"
}
```

Weitere Informationen finden Sie [CreateDeployment](#) in der AWS CodeDeploy APIReferenz.

- APIEinzelheiten finden Sie [CreateDeployment](#) in der AWS CLI Befehlsreferenz.

delete-application

Das folgende Codebeispiel zeigt die Verwendung `delete-application`.

AWS CLI

So löschen Sie eine Anwendung

Im folgenden `delete-application` Beispiel wird die angegebene Anwendung gelöscht, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy delete-application --application-name WordPress_App
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- APIEinzelheiten finden Sie unter [DeleteApplication AWS CLI](#) Befehlsreferenz.

delete-deployment-config

Das folgende Codebeispiel zeigt die Verwendung `delete-deployment-config`.

AWS CLI

Um eine Bereitstellungskonfiguration zu löschen

Im folgenden `delete-deployment-config` Beispiel wird eine benutzerdefinierte Bereitstellungskonfiguration gelöscht, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- APIEinzelheiten finden Sie unter [DeleteDeploymentConfig AWS CLI](#) Befehlsreferenz.

delete-deployment-group

Das folgende Codebeispiel zeigt die Verwendung `delete-deployment-group`.

AWS CLI

Um eine Bereitstellungsgruppe zu löschen

Im folgenden `delete-deployment-group` Beispiel wird eine Bereitstellungsgruppe gelöscht, die der angegebenen Anwendung zugeordnet ist.

```
aws deploy delete-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DG
```

Ausgabe:

```
{  
  "hooksNotCleanedUp": []  
}
```

- API-Einheiten finden Sie unter [DeleteDeploymentGroup AWS CLI Befehlsreferenz](#).

`delete-git-hub-account-token`

Das folgende Codebeispiel zeigt die Verwendung `delete-git-hub-account-token`.

AWS CLI

Um eine GitHub Kontoverbindung zu löschen

Im folgenden `delete-git-hub-account-token` Beispiel wird die Verbindung des angegebenen GitHub Kontos gelöscht.

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

Ausgabe:

```
{  
  "tokenName": "my-github-account"  
}
```

Weitere Informationen finden Sie [DeleteGitHubAccountToken](#) in der AWS CodeDeploy API-Referenz.

- API-Einheiten finden Sie [DeleteGitHubAccountToken](#) in der AWS CLI Befehlsreferenz.

deregister-on-premises-instance

Das folgende Codebeispiel zeigt die Verwendung `deregister-on-premises-instance`.

AWS CLI

Um die Registrierung einer lokalen Instanz aufzuheben

Im folgenden `deregister-on-premises-instance` Beispiel wird die Registrierung einer lokalen Instanz aufgehoben AWS CodeDeploy, der mit der Instanz verknüpfte IAM Benutzer wird jedoch nicht gelöscht, und es wird auch nicht die Zuordnung der Tags für die lokale Instanz von AWS CodeDeploy der Instanz getrennt. Außerdem wird weder der AWS CodeDeploy Agent von der Instanz deinstalliert noch die lokale Konfigurationsdatei aus der Instanz entfernt.

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeregisterOnPremisesInstance AWS CLI](#) Befehlsreferenz.

deregister

Das folgende Codebeispiel zeigt die Verwendung `deregister`.

AWS CLI

Um die Registrierung einer lokalen Instanz aufzuheben

Im folgenden `deregister` Beispiel wird die Registrierung einer lokalen Instanz mit aufgehoben. AWS CodeDeploy Der IAM Benutzer, der der Instanz zugeordnet ist, wird nicht gelöscht. Es trennt die Verknüpfung AWS CodeDeploy der lokalen Tags von der Instanz. Dabei wird weder der AWS CodeDeploy Agent von der Instanz deinstalliert noch die lokale Konfigurationsdatei aus der Instanz entfernt.

```
aws deploy deregister \  
  --instance-name AssetTag12010298EX \  
  --no-delete-iam-user \  
  --region us-west-2
```

Ausgabe:

```
Retrieving on-premises instance information... DONE
```

```
IamUserArn: arn:aws:iam::80398EXAMPLE:user/AWS/CodeDeploy/AssetTag12010298EX
Tags: Key=Name,Value=CodeDeployDemo-OnPrem
Removing tags from the on-premises instance... DONE
Deregistering the on-premises instance... DONE
Run the following command on the on-premises instance to uninstall the codedeploy-
agent:
aws deploy uninstall
```

- API-Einheiten finden Sie unter [Deregister](#) in AWS CLI der Befehlsreferenz.

get-application-revision

Das folgende Codebeispiel zeigt die Verwendung. `get-application-revision`

AWS CLI

Um Informationen über eine Anwendungsrevision zu erhalten

Im folgenden `get-application-revision` Beispiel werden Informationen zu einer Anwendungsrevision angezeigt, die der angegebenen Anwendung zugeordnet ist.

```
aws deploy get-application-revision \
  --application-name WordPress_App \
  --s3-
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

Ausgabe:

```
{
  "applicationName": "WordPress_App",
  "revisionInfo": {
    "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
    "registerTime": 1411076520.009,
    "deploymentGroups": "WordPress_DG",
    "lastUsedTime": 1411076520.009,
    "firstUsedTime": 1411076520.009
  },
  "revision": {
    "revisionType": "S3",
    "s3Location": {
      "bundleType": "zip",
```

```
    "eTag": "dd56cfdEXAMPLE8e768f9d77fEXAMPLE",
    "bucket": "CodeDeployDemoBucket",
    "key": "WordPressApp.zip"
  }
}
```

- API-Einheiten finden Sie [GetApplicationRevision](#) unter AWS CLI Befehlsreferenz.

get-application

Das folgende Codebeispiel zeigt die Verwendung `get-application`.

AWS CLI

Um Informationen über eine Anwendung zu erhalten

Im folgenden `get-application` Beispiel werden Informationen zu einer Anwendung angezeigt, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy get-application --application-name WordPress_App
```

Ausgabe:

```
{
  "application": {
    "applicationName": "WordPress_App",
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "createTime": 1407878168.078,
    "linkedToGitHub": false
  }
}
```

- API-Einheiten finden Sie [GetApplication](#) unter AWS CLI Befehlsreferenz.

get-deployment-config

Das folgende Codebeispiel zeigt die Verwendung `get-deployment-config`.

AWS CLI

Um Informationen über eine Bereitstellungskonfiguration abzurufen

Im folgenden `get-deployment-config` Beispiel werden Informationen zu einer Bereitstellungsconfiguration angezeigt, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Ausgabe:

```
{
  "deploymentConfigInfo": {
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "minimumHealthyHosts": {
      "type": "FLEET_PERCENT",
      "value": 75
    },
    "createTime": 1411081164.379,
    "deploymentConfigName": "ThreeQuartersHealthy"
  }
}
```

- API-Einheiten finden Sie [GetDeploymentConfig](#) unter AWS CLI Befehlsreferenz.

get-deployment-group

Das folgende Codebeispiel zeigt die Verwendung `get-deployment-group`.

AWS CLI

Um Informationen zu einer Bereitstellungsgruppe anzuzeigen

Im folgenden `get-deployment-group` Beispiel werden Informationen zu einer Bereitstellungsgruppe angezeigt, die der angegebenen Anwendung zugeordnet ist.

```
aws deploy get-deployment-group \
  --application-name WordPress_App \
  --deployment-group-name WordPress_DG
```

Ausgabe:

```
{
  "deploymentGroupInfo": {
    "applicationName": "WordPress_App",
    "autoScalingGroups": [
```

```

        "CodeDeployDemo-ASG"
    ],
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "ec2TagFilters": [
        {
            "Type": "KEY_AND_VALUE",
            "Value": "CodeDeployDemo",
            "Key": "Name"
        }
    ],
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "serviceRoleArn": "arn:aws:iam::123456789012:role/CodeDeployDemoRole",
    "deploymentGroupName": "WordPress_DG"
}
}

```

- API-Einheiten finden Sie [GetDeploymentGroup](#) unter AWS CLI Befehlsreferenz.

get-deployment-instance

Das folgende Codebeispiel zeigt die Verwendung `get-deployment-instance`.

AWS CLI

Um Informationen über eine Bereitstellungsinstanz abzurufen

Im folgenden `get-deployment-instance` Beispiel werden Informationen zu einer Bereitstellungsinstanz angezeigt, die der angegebenen Bereitstellung zugeordnet ist.

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-id i-902e9fEX
```

Ausgabe:

```

{
  "instanceSummary": {
    "instanceId": "arn:aws:ec2:us-east-1:80398EXAMPLE:instance/i-902e9fEX",
    "lifecycleEvents": [
      {
        "status": "Succeeded",
        "endTime": 1408480726.569,
        "startTime": 1408480726.437,

```



```
    "lifecycleEventName": "ApplicationStop"
  },
  {
    "status": "Succeeded",
    "endTime": 1408480728.016,
    "startTime": 1408480727.665,
    "lifecycleEventName": "DownloadBundle"
  },
  {
    "status": "Succeeded",
    "endTime": 1408480729.744,
    "startTime": 1408480729.125,
    "lifecycleEventName": "BeforeInstall"
  },
  {
    "status": "Succeeded",
    "endTime": 1408480730.979,
    "startTime": 1408480730.844,
    "lifecycleEventName": "Install"
  },
  {
    "status": "Failed",
    "endTime": 1408480732.603,
    "startTime": 1408480732.1,
    "lifecycleEventName": "AfterInstall"
  },
  {
    "status": "Skipped",
    "endTime": 1408480732.606,
    "lifecycleEventName": "ApplicationStart"
  },
  {
    "status": "Skipped",
    "endTime": 1408480732.606,
    "lifecycleEventName": "ValidateService"
  }
],
"deploymentId": "d-QA4G4F9EX",
"lastUpdatedAt": 1408480733.152,
"status": "Failed"
}
}
```

- API-Einheiten finden Sie [GetDeploymentInstance](#) unter AWS CLI Befehlsreferenz.

get-deployment-target

Das folgende Codebeispiel zeigt die Verwendung `get-deployment-target`.

AWS CLI

Um Informationen über ein Bereitstellungsziel zurückzugeben

Im folgenden `get-deployment-target` Beispiel werden Informationen zu einem Bereitstellungsziel zurückgegeben, das der angegebenen Bereitstellung zugeordnet ist.

```
aws deploy get-deployment-target \
  --deployment-id "d-A1B2C3111" \
  --target-id "i-a1b2c3d4e5f611111"
```

Ausgabe:

```
{
  "deploymentTarget": {
    "deploymentTargetType": "InstanceTarget",
    "instanceTarget": {
      "lastUpdatedAt": 1556918687.504,
      "targetId": "i-a1b2c3d4e5f611111",
      "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-
a1b2c3d4e5f611111",
      "status": "Succeeded",
      "lifecycleEvents": [
        {
          "status": "Succeeded",
          "diagnostics": {
            "errorCode": "Success",
            "message": "Succeeded",
            "logTail": "",
            "scriptName": ""
          },
          "lifecycleEventName": "ApplicationStop",
          "startTime": 1556918592.162,
          "endTime": 1556918592.247
        },
        {
          "status": "Succeeded",
          "diagnostics": {
            "errorCode": "Success",
```

```
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "DownloadBundle",
    "startTime": 1556918593.193,
    "endTime": 1556918593.981
},
{
    "status": "Succeeded",
    "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "BeforeInstall",
    "startTime": 1556918594.805,
    "endTime": 1556918681.807
},
{
    "status": "Succeeded",
    "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "Install",
    "startTime": 1556918682.696,
    "endTime": 1556918683.005
},
{
    "status": "Succeeded",
    "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "AfterInstall",
    "startTime": 1556918684.135,
    "endTime": 1556918684.216
},
```

```
    {
      "status": "Succeeded",
      "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
      },
      "lifecycleEventName": "ApplicationStart",
      "startTime": 1556918685.211,
      "endTime": 1556918685.295
    },
    {
      "status": "Succeeded",
      "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
      },
      "lifecycleEventName": "ValidateService",
      "startTime": 1556918686.65,
      "endTime": 1556918686.747
    }
  ],
  "deploymentId": "d-A1B2C3111"
}
}
```

Weitere Informationen finden Sie [GetDeploymentTarget](#) in der AWS CodeDeploy API Referenz.

- API Einheiten finden Sie [GetDeploymentTarget](#) in der AWS CLI Befehlsreferenz.

get-deployment

Das folgende Codebeispiel zeigt die Verwendung `get-deployment`.

AWS CLI

Um Informationen über eine Bereitstellung zu erhalten

Im folgenden `get-deployment` Beispiel werden Informationen zu einer Bereitstellung angezeigt, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

Ausgabe:

```
{
  "deploymentInfo": {
    "applicationName": "WordPress_App",
    "status": "Succeeded",
    "deploymentOverview": {
      "Failed": 0,
      "InProgress": 0,
      "Skipped": 0,
      "Succeeded": 1,
      "Pending": 0
    },
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "creator": "user",
    "description": "My WordPress app deployment",
    "revision": {
      "revisionType": "S3",
      "s3Location": {
        "bundleType": "zip",
        "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
        "bucket": "CodeDeployDemoBucket",
        "key": "WordPressApp.zip"
      }
    },
    "deploymentId": "d-A1B2C3123",
    "deploymentGroupName": "WordPress_DG",
    "createTime": 1409764576.589,
    "completeTime": 1409764596.101,
    "ignoreApplicationStopFailures": false
  }
}
```

- API-Einheiten finden Sie [GetDeployment](#) unter AWS CLI Befehlsreferenz.

get-on-premises-instance

Das folgende Codebeispiel zeigt die Verwendung `get-on-premises-instance`.

AWS CLI

Um Informationen über eine lokale Instanz abzurufen

Im folgenden `get-on-premises-instance` Beispiel werden Informationen über die angegebene lokale Instanz abgerufen.

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Ausgabe:

```
{
  "instanceInfo": {
    "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag12010298EX",
    "tags": [
      {
        "Value": "CodeDeployDemo-OnPrem",
        "Key": "Name"
      }
    ],
    "instanceName": "AssetTag12010298EX",
    "registerTime": 1425579465.228,
    "instanceArn": "arn:aws:codedeploy:us-east-1:123456789012:instance/
AssetTag12010298EX_4IwLNI2Alh"
  }
}
```

- API-Einheiten finden Sie unter [GetOnPremisesInstance AWS CLI](#) Befehlsreferenz.

install

Das folgende Codebeispiel zeigt die Verwendung `install`.

AWS CLI

Um eine lokale Instanz zu installieren

Im folgenden `install` Beispiel wird die lokale Konfigurationsdatei vom angegebenen Speicherort auf der Instanz an den Speicherort auf der Instanz kopiert, von dem der AWS CodeDeploy Agent erwartet, dass er sie findet. Außerdem wird der AWS CodeDeploy Agent auf der Instanz installiert. Es erstellt keinen IAM Benutzer, registriert die lokale Instanz nicht bei AWS CodeDeploy und ordnet der Instanz auch keine lokalen Instanz-Tags AWS CodeDeploy zu.

```
aws deploy install \  
  --override-config \  
  --config-file C:\temp\codedeploy.onpremises.yml \  
  --region us-west-2 \  
  --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-agent.msi
```

Ausgabe:

```
Creating the on-premises instance configuration file... DONE  
Installing the AWS CodeDeploy Agent... DONE
```

- API-Einheiten finden Sie unter [Installation](#) in der AWS CLI Befehlsreferenz.

list-application-revisions

Das folgende Codebeispiel zeigt die Verwendung `list-application-revisions`.

AWS CLI

Um Informationen über Anwendungsrevisionen zu erhalten

Im folgenden `list-application-revisions` Beispiel werden Informationen zu allen Anwendungsrevisionen angezeigt, die der angegebenen Anwendung zugeordnet sind.

```
aws deploy list-application-revisions \  
  --application-name WordPress_App \  
  --s-3-bucket CodeDeployDemoBucket \  
  --deployed exclude \  
  --s-3-key-prefix WordPress_ \  
  --sort-by LastUsedTime \  
  --sort-order descending
```

Ausgabe:

```
{
  "revisions": [
    {
      "revisionType": "S3",
      "s3Location": {
        "version": "uTecLusvCB_JqHFxtfUcyfV8bEXAMPLE",
        "bucket": "CodeDeployDemoBucket",
        "key": "WordPress_App.zip",
        "bundleType": "zip"
      }
    },
    {
      "revisionType": "S3",
      "s3Location": {
        "version": "tMk.UxgDpMEVb7V187ZM6wVAWEXAMPLE",
        "bucket": "CodeDeployDemoBucket",
        "key": "WordPress_App_2-0.zip",
        "bundleType": "zip"
      }
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListApplicationRevisions AWS CLI](#) Befehlsreferenz.

list-applications

Das folgende Codebeispiel zeigt die Verwendung `list-applications`.

AWS CLI

Um Informationen über Anwendungen zu erhalten

Im folgenden `list-applications` Beispiel werden Informationen zu allen Anwendungen angezeigt, die dem AWS Konto des Benutzers zugeordnet sind.

```
aws deploy list-applications
```

Ausgabe:

```
{
```



```
"applications": [  
  "WordPress_App",  
  "MyOther_App"  
]  
}
```

- API-Einheiten finden Sie [ListApplications](#) unter AWS CLI Befehlsreferenz.

list-deployment-configs

Das folgende Codebeispiel zeigt die Verwendung `list-deployment-configs`.

AWS CLI

Um Informationen zu Bereitstellungs-konfigurationen zu erhalten

Im folgenden `list-deployment-configs` Beispiel werden Informationen zu allen Bereitstellungs-konfigurationen angezeigt, die dem AWS Benutzerkonto zugeordnet sind.

```
aws deploy list-deployment-configs
```

Ausgabe:

```
{  
  "deploymentConfigsList": [  
    "ThreeQuartersHealthy",  
    "CodeDeployDefault.AllAtOnce",  
    "CodeDeployDefault.HalfAtATime",  
    "CodeDeployDefault.OneAtATime"  
  ]  
}
```

- API-Einheiten finden Sie [ListDeploymentConfigs](#) unter AWS CLI Befehlsreferenz.

list-deployment-groups

Das folgende Codebeispiel zeigt die Verwendung `list-deployment-groups`.

AWS CLI

Um Informationen über Bereitstellungsgruppen zu erhalten

Im folgenden `list-deployment-groups` Beispiel werden Informationen zu allen Bereitstellungsgruppen angezeigt, die der angegebenen Anwendung zugeordnet sind.

```
aws deploy list-deployment-groups --application-name WordPress_App
```

Ausgabe:

```
{
  "applicationName": "WordPress_App",
  "deploymentGroups": [
    "WordPress_DG",
    "WordPress_Beta_DG"
  ]
}
```

- API-Einheiten finden Sie [ListDeploymentGroups](#) unter AWS CLI Befehlsreferenz.

list-deployment-instances

Das folgende Codebeispiel zeigt die Verwendung `list-deployment-instances`.

AWS CLI

Um Informationen über Bereitstellungsinstanzen zu erhalten

Im folgenden `list-deployment-instances` Beispiel werden Informationen zu allen Bereitstellungsinstanzen angezeigt, die der angegebenen Bereitstellung zugeordnet sind.

```
aws deploy list-deployment-instances \
  --deployment-id d-A1B2C3111 \
  --instance-status-filter Succeeded
```

Ausgabe:

```
{
  "instancesList": [
    "i-EXAMPLE11",
    "i-EXAMPLE22"
  ]
}
```

- API-Einheiten finden Sie [ListDeploymentInstances](#) unter AWS CLI Befehlsreferenz.

list-deployment-targets

Das folgende Codebeispiel zeigt die Verwendung `list-deployment-targets`.

AWS CLI

Um eine Liste von Zielen abzurufen IDs, die einer Bereitstellung zugeordnet sind

Im folgenden `list-deployment-targets` Beispiel wird eine Liste von Zielen abgerufen, die Bereitstellungen IDs zugeordnet sind, die den Status „Fehlgeschlagen“ oder „InProgress“ haben.

```
aws deploy list-deployment-targets \
  --deployment-id "d-A1B2C3111" \
  --target-filters "{\"TargetStatus\": [\"Failed\", \"InProgress\"]}"
```

Ausgabe:

```
{
  "targetIds": [
    "i-0f1558aaf90e5f1f9"
  ]
}
```

Weitere Informationen finden Sie [ListDeploymentTargets](#) in der AWS CodeDeploy API Referenz.

- API-Einheiten finden Sie [ListDeploymentTargets](#) in der AWS CLI Befehlsreferenz.

list-deployments

Das folgende Codebeispiel zeigt die Verwendung `list-deployments`.

AWS CLI

Um Informationen über Bereitstellungen zu erhalten

Im folgenden `list-deployments` Beispiel werden Informationen zu allen Bereitstellungen angezeigt, die der angegebenen Anwendung und Bereitstellungsgruppe zugeordnet sind.

```
aws deploy list-deployments \
  --application-name WordPress_App \
```

```
--create-time-range start=2014-08-19T00:00:00,end=2014-08-20T00:00:00 \  
--deployment-group-name WordPress_DG \  
--include-only-statuses Failed
```

Ausgabe:

```
{  
  "deployments": [  
    "d-EXAMPLE11",  
    "d-EXAMPLE22",  
    "d-EXAMPLE33"  
  ]  
}
```

- API-Einheiten finden Sie unter [ListDeployments AWS CLI](#) Befehlsreferenz.

list-git-hub-account-token-names

Das folgende Codebeispiel zeigt die Verwendung `list-git-hub-account-token-names`.

AWS CLI

Um die Namen der gespeicherten Verbindungen zu GitHub-Konten aufzulisten

Im folgenden `list-git-hub-account-token-names` Beispiel werden die Namen der gespeicherten Verbindungen zu GitHub-Konten für den aktuellen AWS-Benutzer aufgeführt.

```
aws deploy list-git-hub-account-token-names
```

Ausgabe:

```
{  
  "tokenNameList": [  
    "my-first-token",  
    "my-second-token",  
    "my-third-token"  
  ]  
}
```

Weitere Informationen finden Sie [ListGitHubAccountTokenNames](#) in der AWS CodeDeploy API-Referenz.

- API-Einheiten finden Sie [ListGitHubAccountTokenNames](#) in der AWS CLI Befehlsreferenz.

list-on-premises-instances

Das folgende Codebeispiel zeigt die Verwendung `list-on-premises-instances`.

AWS CLI

Um Informationen über eine oder mehrere lokale Instanzen zu erhalten

Im folgenden `list-on-premises-instances` Beispiel wird eine Liste verfügbarer lokaler Instanznamen für Instanzen abgerufen, die in der Instanz registriert sind AWS CodeDeploy und denen auch das angegebene lokale Instanz-Tag zugeordnet ist. AWS CodeDeploy

```
aws deploy list-on-premises-instances \
  --registration-status Registered \
  --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

Ausgabe:

```
{
  "instanceNames": [
    "AssetTag12010298EX"
  ]
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [ListOnPremisesInstances](#).AWS CLI

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource (Anwendung) aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags auf, die auf eine Anwendung mit dem Namen `testApp` in angewendet wurden CodeDeploy.

```
aws deploy list-tags-for-resource \
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "Type",
      "Value": "testType"
    },
    {
      "Key": "Name",
      "Value": "testName"
    }
  ]
}
```

Weitere Informationen finden Sie [im CodeDeploy AWS CodeDeploy Benutzerhandbuch unter Tagging Instances für Deployment-Gruppen](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

push

Das folgende Codebeispiel zeigt die Verwendung `push`.

AWS CLI

Um eine AWS CodeDeploy kompatible Anwendungsrevision für Amazon S3 zu bündeln und bereitzustellen

Das folgende `push` Beispiel bündelt und stellt eine Anwendungsrevision für Amazon S3 bereit und ordnet dann die Anwendungsrevision der angegebenen Anwendung zu.

```
aws deploy push \
  --application-name WordPress_App \
  --description "This is my deployment" \
  --ignore-hidden-files \
  --s3-location s3://CodeDeployDemoBucket/WordPressApp.zip \
  --source /tmp/MyLocalDeploymentFolder/
```

In der Ausgabe wird beschrieben, wie der `create-deployment` Befehl verwendet wird, um eine Bereitstellung zu erstellen, die die hochgeladene Anwendungsrevision verwendet.

To deploy with this revision, run:

```
aws deploy create-deployment --application-name WordPress_App
--deployment-config-name <deployment-config-name> --
deployment-group-name <deployment-group-name> --s3-location
bucket=CodeDeployDemoBucket,key=WordPressApp.zip,bundleType=zip,eTag="cecc9b8EXAMPLE50a6e71
```

- API-Einheiten finden Sie unter [Push](#) in AWS CLI Command Reference.

register-application-revision

Das folgende Codebeispiel zeigt die Verwendung `register-application-revision`.

AWS CLI

Um Informationen über eine bereits hochgeladene Anwendungsrevision zu registrieren

Das folgende `register-application-revision` Beispiel registriert Informationen über eine bereits hochgeladene Anwendungsrevision, die in Amazon S3 gespeichert ist, mit AWS CodeDeploy

```
aws deploy register-application-revision \
--application-name WordPress_App \
--description "Revised WordPress application" \
--s3-
location bucket=CodeDeployDemoBucket,key=RevisedWordPressApp.zip,bundleType=zip,eTag=cecc9b8
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [RegisterApplicationRevision AWS CLI](#) Befehlsreferenz.

register-on-premises-instance

Das folgende Codebeispiel zeigt die Verwendung `register-on-premises-instance`.

AWS CLI

Um eine lokale Instanz zu registrieren

Im folgenden `register-on-premises-instance` Beispiel wird eine lokale Instanz bei registriert. AWS CodeDeploy Es erstellt weder den angegebenen IAM Benutzer noch ordnet es der registrierten Instanz in AWS CodeDeploy lokalen Instanzen Tags zu.

```
aws deploy register-on-premises-instance \  
  --instance-name AssetTag12010298EX \  
  --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [RegisterOnPremisesInstance](#) in der AWS CLI Befehlsreferenz.

register

Das folgende Codebeispiel zeigt die Verwendung `register`.

AWS CLI

Um eine lokale Instanz zu registrieren

Im folgenden `register` Beispiel wird eine lokale Instanz bei registriert AWS CodeDeploy, AWS CodeDeploy das angegebene lokale Instanz-Tag der registrierten Instanz zugeordnet und eine lokale Konfigurationsdatei erstellt, die in die Instanz kopiert werden kann. Es erstellt weder den IAM Benutzer noch installiert es den AWS CodeDeploy Agenten auf der Instanz.

```
aws deploy register \  
  --instance-name AssetTag12010298EX \  
  --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployUser-OnPrem \  
  --tags Key=Name,Value=CodeDeployDemo-OnPrem \  
  --region us-west-2
```

Ausgabe:

```
Registering the on-premises instance... DONE  
Adding tags to the on-premises instance... DONE  
Copy the on-premises configuration file named codedeploy.onpremises.yml to the on-  
premises instance, and run the following command on the on-premises instance to  
install and configure the AWS CodeDeploy Agent:  
aws deploy install --config-file codedeploy.onpremises.yml
```

- API-Einheiten finden Sie unter [Registrieren](#) in der AWS CLI Befehlsreferenz.

remove-tags-from-on-premises-instances

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-on-premises-instances`.

AWS CLI

Um Tags aus einer oder mehreren lokalen Instanzen zu entfernen

Im folgenden `remove-tags-from-on-premises-instances` Beispiel wird die Zuordnung der angegebenen lokalen Tags zu lokalen Instanzen aufgehoben AWS CodeDeploy . Es wird weder die Registrierung der lokalen Instanzen in AWS CodeDeploy der Instanz aufgehoben noch der AWS CodeDeploy Agent von der Instanz deinstalliert, noch wird die lokale Konfigurationsdatei aus den Instanzen entfernt, noch werden die IAM Benutzer gelöscht, die den Instanzen zugeordnet sind.

```
aws deploy remove-tags-from-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie in der Befehlsreferenz [RemoveTagsFromOnPremisesInstances](#).AWS CLI

stop-deployment

Das folgende Codebeispiel zeigt die Verwendung `stop-deployment`.

AWS CLI

Um zu versuchen, eine Bereitstellung zu beenden

Im folgenden `stop-deployment` Beispiel wird versucht, eine laufende Bereitstellung zu beenden, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy stop-deployment --deployment-id D-a1b2c3111
```

Ausgabe:

```
{  
  "status": "Succeeded",  
  "statusMessage": "No more commands will be scheduled for execution in the  
  deployment instances"  
}
```

- API-Einheiten finden Sie in der [StopDeployment](#)-Befehlsreferenz AWS CLI .

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource (Anwendung) zu taggen

Im folgenden `tag-resource` Beispiel werden zwei Tags mit den Schlüsseln `Name` und `Type` sowie Werten `testName` und `testType` zu einer Anwendung mit dem Namen `testApp` in hinzugefügt. `CodeDeploy` :

```
aws deploy tag-resource \  
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp \  
  --tags Key=Name,Value=testName Key=Type,Value=testType
```

Bei Erfolg erzeugt dieser Befehl keine Ausgabe.

Weitere Informationen finden Sie [im CodeDeploy AWS CodeDeploy Benutzerhandbuch unter Tagging Instances für Deployment-Gruppen](#).

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

uninstall

Das folgende Codebeispiel zeigt die Verwendung `uninstall`.

AWS CLI

Um eine lokale Instanz zu deinstallieren

Im folgenden `uninstall` Beispiel wird der AWS CodeDeploy Agent von der lokalen Instanz deinstalliert und die lokale Konfigurationsdatei aus der Instanz entfernt. Es wird weder die Instanz in der Instance deregistriert AWS CodeDeploy, noch werden irgendwelche On-Premises-Instanz-Tags AWS CodeDeploy von der Instanz getrennt, noch wird der IAM Benutzer gelöscht, der der Instanz zugeordnet ist.

```
aws deploy uninstall
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [Deinstallation](#) in der Befehlsreferenz.AWS CLI

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressource (Anwendung) zu entfernen

Im folgenden `untag-resource` Beispiel werden zwei Tags mit den Schlüsseln `Name` und `Type` aus einer Anwendung mit dem Namen `testApp` in entfernt CodeDeploy.

```
aws deploy untag-resource \  
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp \  
  --tag-keys Name Type
```

Bei Erfolg erzeugt dieser Befehl keine Ausgabe.

Weitere Informationen finden Sie [im CodeDeploy AWS CodeDeploy Benutzerhandbuch unter Tagging Instances für Deployment-Gruppen](#).

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-application

Das folgende Codebeispiel zeigt die Verwendung `update-application`.

AWS CLI

Um die Details einer Anwendung zu ändern

Im folgenden `update-application` Beispiel wird der Name einer Anwendung geändert, die dem AWS Konto des Benutzers zugeordnet ist.

```
aws deploy update-application \  
  --application-name WordPress_App \  
  --new-application-name My_WordPress_App
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- APIEinzelheiten finden Sie [UpdateApplication](#) unter AWS CLI Befehlsreferenz.

update-deployment-group

Das folgende Codebeispiel zeigt die Verwendung `update-deployment-group`.

AWS CLI

Um Informationen zu einer Bereitstellungsgruppe zu ändern

Im folgenden `update-deployment-group` Beispiel werden die Einstellungen einer Bereitstellungsgruppe geändert, die der angegebenen Anwendung zugeordnet ist.

```
aws deploy update-deployment-group \
  --application-name WordPress_App \
  --auto-scaling-groups My_CodeDeployDemo_ASG \
  --current-deployment-group-name WordPress_DG \
  --deployment-config-name CodeDeployDefault.AllAtOnce \
  --ec2-tag-filters Key=Name,Type=KEY_AND_VALUE,Value=My_CodeDeployDemo \
  --new-deployment-group-name My_WordPress_DepGroup \
  --service-role-arn arn:aws:iam::80398EXAMPLE:role/CodeDeployDemo-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- APIEinzelheiten finden Sie [UpdateDeploymentGroup](#) unter AWS CLI Befehlsreferenz.

CodeGuru Beispiele für Gutachter mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with CodeGuru Reviewer Aktionen ausführen und gängige Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-repository

Das folgende Codebeispiel zeigt die Verwendung `associate-repository`.

AWS CLI

Beispiel 1: So erstellen Sie eine Bitbucket-Repository-Verknüpfung

Im folgenden `associate-repository` Beispiel wird mithilfe eines vorhandenen Bitbucket-Repositorys eine Repository-Verknüpfung erstellt.

```
aws codeguru-reviewer associate-repository \  
  --repository 'Bitbucket={Owner=sample-owner, Name=mySampleRepo,  
  ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/  
  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

Ausgabe:

```
{  
  "RepositoryAssociation": {  
    "ProviderType": "Bitbucket",  
    "Name": "mySampleRepo",  
    "LastUpdatedTimeStamp": 1596216896.979,  
    "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "CreatedTimeStamp": 1596216896.979,  
    "ConnectionArn": "arn:aws:codestar-connections:us-  
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "State": "Associating",  
    "StateReason": "Pending Repository Association",  
    "AssociationArn": "arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "Owner": "sample-owner"  
  }  
}
```

Weitere Informationen finden Sie unter [Eine Bitbucket-Repository-Verknüpfung in Amazon CodeGuru Reviewer erstellen](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

Beispiel 2: So erstellen Sie eine GitHub Enterprise-Repository-Zuordnung

Das folgende `associate-repository` Beispiel erstellt eine Repository-Zuordnung unter Verwendung eines vorhandenen GitHub Enterprise-Repositorys.

```
aws codeguru-reviewer associate-repository \
  --repository 'GitHubEnterpriseServer={Owner=sample-owner, Name=mySampleRepo,
  ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/
  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

Ausgabe:

```
{
  "RepositoryAssociation": {
    "ProviderType": "GitHubEnterpriseServer",
    "Name": "mySampleRepo",
    "LastUpdatedTimeStamp": 1596216896.979,
    "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "CreatedTimeStamp": 1596216896.979,
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "State": "Associating",
    "StateReason": "Pending Repository Association",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Owner": "sample-owner"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen einer GitHub Enterprise Server-Repository-Zuordnung in Amazon CodeGuru Reviewer](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

Beispiel 3: So erstellen Sie eine AWS CodeCommit Repository-Zuordnung

Das folgende `associate-repository` Beispiel erstellt eine Repository-Zuordnung unter Verwendung eines vorhandenen AWS CodeCommit Repositorys.

```
aws codeguru-reviewer associate-repository \
  --repository CodeCommit={Name=mySampleRepo}
```

Ausgabe:

```
{
  "RepositoryAssociation": {
```

```

    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "My-ecs-beta-repo",
    "LastUpdatedTimeStamp": 1595634764.029,
    "ProviderType": "CodeCommit",
    "CreatedTimeStamp": 1595634764.029,
    "Owner": "544120495673",
    "State": "Associating",
    "StateReason": "Pending Repository Association",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:544120495673:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen einer AWS CodeCommit Repository-Verknüpfung in Amazon CodeGuru Reviewer](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [AssociateRepository](#) in der AWS CLI Befehlsreferenz.

create-code-review

Das folgende Codebeispiel zeigt die Verwendung `create-code-review`.

AWS CLI

Um einen Code-Review zu erstellen.

Im Folgenden `create-code-review` wird eine Überprüfung des Codes im `mainline` Branch eines AWS CodeCommit Repositories erstellt, der benannt ist `my-repository-name`.

```

aws codeguru-reviewer create-code-review \
  --name my-code-review \
  --repository-association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --type '{"RepositoryAnalysis": {"RepositoryHead": {"BranchName": "mainline"}}}'

```

Ausgabe:

```

{
  "CodeReview": {
    "Name": "my-code-review",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-code-review",

```

```

    "RepositoryName": "my-repository-name",
    "Owner": "123456789012",
    "ProviderType": "CodeCommit",
    "State": "Pending",
    "StateReason": "CodeGuru Reviewer has received the request, and a code
review is scheduled.",
    "CreatedTimeStamp": 1618873489.195,
    "LastUpdatedTimeStamp": 1618873489.195,
    "Type": "RepositoryAnalysis",
    "SourceCodeType": {
      "RepositoryHead": {
        "BranchName": "mainline"
      }
    },
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

Weitere Informationen finden Sie unter [Code-Rezensionen in Amazon CodeGuru Reviewer erstellen](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCodeReview](#) in der AWS CLI Befehlsreferenz.

describe-code-review

Das folgende Codebeispiel zeigt die Verwendung `describe-code-review`.

AWS CLI

Listet Details zu einer Code-Überprüfung auf.

Im Folgenden `describe-code-review` werden Informationen zu einer Überprüfung von Code im Zweig „main line“ eines AWS CodeCommit Repositorys mit dem Namen "my-repo-name" aufgeführt.

```

aws codeguru-reviewer put-recommendation-feedback \
  --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefg12345678 \
  --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \

```


--reactions *ThumbsUp*

Output

```
{
  "CodeReview": {
    "Name": "My-ecs-beta-repo-master-xs6di4kfd4j269dz",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-repo-name",
    "RepositoryName": "My-ecs-beta-repo",
    "Owner": "123456789012",
    "ProviderType": "CodeCommit",
    "State": "Pending",
    "StateReason": "CodeGuru Reviewer is reviewing the source code.",
    "CreatedTimeStamp": 1618874226.226,
    "LastUpdatedTimeStamp": 1618874233.689,
    "Type": "RepositoryAnalysis",
    "SourceCodeType": {
      "RepositoryHead": {
        "BranchName": "mainline"
      }
    },
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}
```

Weitere Informationen finden Sie unter [Details zur Codeüberprüfung anzeigen](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCodeReview](#) in der AWS CLI Befehlsreferenz.

describe-recommendation-feedback

Das folgende Codebeispiel zeigt die Verwendung `describe-recommendation-feedback`.

AWS CLI

Um Informationen über Feedback zu einer Empfehlung anzuzeigen

Im Folgenden `describe-recommendation-feedback` werden Informationen zu Feedback zu einer Empfehlung angezeigt. Diese Empfehlung hat eine ThumbsUp Reaktion.

```
aws codeguru-reviewer describe-recommendation-feedback \
  --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \
  --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb
```

Ausgabe:

```
{
  "RecommendationFeedback": {
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678",
    "RecommendationId":
    "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
    "Reactions": [
      "ThumbsUp"
    ],
    "UserId": "aws-user-id",
    "CreatedTimeStamp": 1618877070.313,
    "LastUpdatedTimeStamp": 1618877948.881
  }
}
```

Weitere Informationen finden Sie unter [Empfehlungen anzeigen und Feedback geben](#) und [Schritt 4: Feedback geben](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeRecommendationFeedback](#) in der AWS CLI Befehlsreferenz.

describe-repository-association

Das folgende Codebeispiel zeigt die Verwendung `describe-repository-association`.

AWS CLI

Beispiel 1: Um Informationen über eine GitHub Repository-Zuordnung zurückzugeben

Das folgende `describe-repository-association` Beispiel gibt Informationen über eine Repository-Zuordnung zurück, die ein GitHub Enterprise-Repository verwendet und sich im `Associated` Status befindet.

```
aws codeguru-reviewer describe-repository-association \  
  --association-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "RepositoryAssociation": {  
    "AssociationId": "b822717e-0711-4e8a-bada-0e738289c75e",  
    "Name": "mySampleRepo",  
    "LastUpdatedTimeStamp": 1588102637.649,  
    "ProviderType": "GitHub",  
    "CreatedTimeStamp": 1588102615.636,  
    "Owner": "sample-owner",  
    "State": "Associated",  
    "StateReason": "Pull Request Notification configuration successful",  
    "AssociationArn": "arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer GitHub Enterprise Server-Repository-Zuordnung in Amazon CodeGuru Reviewer](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

Beispiel 2: Um Informationen über eine fehlgeschlagene Repository-Zuordnung zurückzugeben

Das folgende `describe-repository-association` Beispiel gibt Informationen über eine Repository-Zuordnung zurück, die ein GitHub Enterprise-Repository verwendet und sich im Failed Status befindet.

```
aws codeguru-reviewer describe-repository-association \  
  --association-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "RepositoryAssociation": {  
    "ProviderType": "GitHubEnterpriseServer",  
    "Name": "mySampleRepo",  
    "LastUpdatedTimeStamp": 1596217036.892,  
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```

    "CreatedTimeStamp": 1596216896.979,
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "State": "Failed",
    "StateReason": "Failed, Please retry.",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "Owner": "sample-owner"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen einer GitHub Enterprise Server-Repository-Zuordnung in Amazon CodeGuru Reviewer](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

Beispiel 3: Um Informationen über eine aufgelöste Repository-Zuordnung zurückzugeben

Das folgende `describe-repository-association` Beispiel gibt Informationen über eine Repository-Zuordnung zurück, die ein GitHub Enterprise-Repository verwendet und sich im `Disassociating` Status befindet.

```

aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "RepositoryAssociation": {
    "ProviderType": "GitHubEnterpriseServer",
    "Name": "mySampleRepo",
    "LastUpdatedTimeStamp": 1596217036.892,
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "CreatedTimeStamp": 1596216896.979,
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "State": "Disassociating",
    "StateReason": "Source code access removal in progress",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "Owner": "sample-owner"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen einer GitHub Enterprise Server-Repository-Zuordnung in Amazon CodeGuru Reviewer](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeRepositoryAssociation](#) in der AWS CLI Befehlsreferenz.

disassociate-repository

Das folgende Codebeispiel zeigt die Verwendung `disassociate-repository`.

AWS CLI

Um die Zuordnung einer Repository-Verknüpfung aufzuheben

Im Folgenden wird die `disassociate-repository` Zuordnung einer Repository-Zuordnung aufgehoben, die ein AWS CodeCommit Repository verwendet.

```
aws codeguru-reviewer disassociate-repository \  
  --association-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "RepositoryAssociation": {  
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "AssociationArn": "arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Name": "my-repository",  
    "Owner": "123456789012",  
    "ProviderType": "CodeCommit",  
    "State": "Disassociating",  
    "LastUpdatedTimeStamp": 1618939174.759,  
    "CreatedTimeStamp": 1595636947.096  
  },  
  "Tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

Weitere Informationen finden Sie unter [Zuordnung eines Repositories in CodeGuru Reviewer](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateRepository](#) in der AWS CLI Befehlsreferenz.

list-code-reviews

Das folgende Codebeispiel zeigt die Verwendung `list-code-reviews`.

AWS CLI

Um Code-Rezensionen aufzulisten, die in den letzten 90 Tagen in Ihrem AWS Konto erstellt wurden.

Das folgende `list-code-reviews` Beispiel listet die Code-Reviews auf, die in den letzten 90 Tagen mithilfe von Pull-Requests erstellt wurden.

```
aws codeguru-reviewer list-code-reviews \  
  --type PullRequest
```

Ausgabe:

```
{  
  "CodeReviewSummaries": [  
    {  
      "LastUpdatedTimeStamp": 1588897288.054,  
      "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ProviderType": "GitHub",  
      "PullRequestId": "5",  
      "MetricsSummary": {  
        "MeteredLinesOfCodeCount": 24,  
        "FindingsCount": 1  
      },  
      "CreatedTimeStamp": 1588897068.512,  
      "State": "Completed",  
      "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-  
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Owner": "sample-owner",  
      "RepositoryName": "sample-repository-name",  
      "Type": "PullRequest"  
    },  
    {  
      "LastUpdatedTimeStamp": 1588869793.263,  
      "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "ProviderType": "GitHub",  
      "State": "Completed",  
      "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-  
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "Owner": "sample-owner",  
      "RepositoryName": "sample-repository-name",  
      "Type": "PullRequest"  
    }  
  ]  
}
```

```
    "PullRequestId": "4",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 29,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588869575.949,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588870511.211,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "ProviderType": "GitHub",
    "PullRequestId": "4",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 2,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588870292.425,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588118522.452,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
    "ProviderType": "GitHub",
    "PullRequestId": "3",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 29,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588118301.131,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
    "Owner": "sample-owner",
```

```

    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588112205.207,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
    "ProviderType": "GitHub",
    "PullRequestId": "2",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 25,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588111987.443,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588104489.981,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
    "ProviderType": "GitHub",
    "PullRequestId": "1",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 25,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588104270.223,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  }
]
}

```

Weitere Informationen finden Sie unter [Alle Code-Rezensionen anzeigen](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListCodeReviews](#) in der AWS CLI Befehlsreferenz.

list-recommendation-feedback

Das folgende Codebeispiel zeigt die Verwendung `list-recommendation-feedback`.

AWS CLI

Um das Feedback von Kunden zu einer Empfehlung in einem zugehörigen Repository aufzulisten

Im Folgenden wird Kundenfeedback zu allen Empfehlungen im Rahmen einer Code-Überprüfung `list-recommendation-feedback` aufgeführt. Dieser Code-Review enthält ein Feedback, ein "ThumbsUp,, von einem Kunden.

```
aws codeguru-reviewer list-recommendation-feedback \
  --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefg12345678
```

Ausgabe:

```
{
  "RecommendationFeedbackSummaries": [
    {
      "RecommendationId":
"3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
      "Reactions": [
        "ThumbsUp"
      ],
      "UserId": "aws-user-id"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Schritt 4: Feedback geben](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRecommendationFeedback](#) in der AWS CLI Befehlsreferenz.

list-recommendations

Das folgende Codebeispiel zeigt die Verwendung `list-recommendations`.

AWS CLI

Um die Empfehlungen für einen abgeschlossenen Code-Review aufzulisten

Im folgenden `list-recommendations` Beispiel sind die Empfehlungen für einen abgeschlossenen Code-Review aufgeführt. Dieser Code-Review enthält eine Empfehlung.

```
aws codeguru-reviewer list-recommendations \  
  --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-  
  review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "RecommendationSummaries": [  
    {  
      "Description": "\n\nProblem  \n You are using a `ConcurrentHashMap`,  
but your usage of `containsKey()` and `get()` may not be thread-safe at lines: **63  
and 64**. In between the check and the `get()` another thread can remove the key  
and the `get()` will return `null`. The remove that can remove the key is at line:  
**59**.\n\nFix  \n Consider calling `get()`, checking instead of your current  
check if the returned object is `null`, and then using that object only, without  
calling `get()` again.\n\nMore info  \n [View an example on GitHub](https://  
github.com/apache/hadoop/blob/f16cf877e565084c66bc63605659b157c4394dc8/hadoop-tools/  
hadoop-aws/src/main/java/org/apache/hadoop/fs/s3a/s3guard/S3Guard.java#L302-L304)  
(external link).",  
      "RecommendationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "StartLine": 63,  
      "EndLine": 64,  
      "FilePath": "src/main/java/com/company/sample/application/  
CreateOrderThread.java"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Schritt 4: Feedback geben](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRecommendations](#) in der AWS CLI Befehlsreferenz.

list-repository-associations

Das folgende Codebeispiel zeigt die Verwendung `list-repository-associations`.

AWS CLI

Um die Repository-Verknüpfungen in Ihrem AWS Konto aufzulisten

Das folgende `list-repository-associations` Beispiel gibt eine Liste der Objekte mit der Zusammenfassung der Repository-Verknüpfungen in Ihrem Konto zurück. Sie können die zurückgegebene Liste nach `ProviderType`, `NameState`, und `filternOwner`.

```
aws codeguru-reviewer list-repository-associations
```

Ausgabe:

```
{
  "RepositoryAssociationSummaries": [
    {
      "LastUpdatedTimeStamp": 1595886609.616,
      "Name": "test",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Owner": "sample-owner",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProviderType": "Bitbucket"
    },
    {
      "LastUpdatedTimeStamp": 1595636969.035,
      "Name": "CodeDeploy-CodePipeline-ECS-Tutorial",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "Owner": "123456789012",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "ProviderType": "CodeCommit"
    },
    {
      "LastUpdatedTimeStamp": 1595634785.983,
      "Name": "My-ecs-beta-repo",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "Owner": "123456789012",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "ProviderType": "CodeCommit"
    }
  ]
}
```

```
    },
    {
      "LastUpdatedTimeStamp": 1590712811.77,
      "Name": "MyTestCodeCommit",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
      "Owner": "123456789012",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
      "ProviderType": "CodeCommit"
    },
    {
      "LastUpdatedTimeStamp": 1588102637.649,
      "Name": "aws-codeguru-profiler-sample-application",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
      "Owner": "sample-owner",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
      "ProviderType": "GitHub"
    },
    {
      "LastUpdatedTimeStamp": 1588028233.995,
      "Name": "codeguru-profiler-demo-app",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
      "Owner": "sample-owner",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
      "ProviderType": "GitHub"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Alle Repository-Verknüpfungen in CodeGuru Reviewer](#) anzeigen im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRepositoryAssociations](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags in einem zugehörigen Repository aufzulisten

Im Folgenden `list-tags-for-resource` werden die Tags in einem zugehörigen Repository aufgeführt. Dieses zugehörige Repository hat zwei Tags.

```
aws codeguru-reviewer list-tags-for-resource \  
  --resource-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "Tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

Weitere Informationen finden Sie unter [Tags für ein mit CodeGuru Reviewer verknüpftes Repository anzeigen \(AWS CLI\)](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-recommendation-feedback

Das folgende Codebeispiel zeigt die Verwendung `put-recommendation-feedback`.

AWS CLI

Um eine Empfehlung zu einem Code-Review hinzuzufügen

Im Folgenden finden `put-recommendation-feedback` Sie eine ThumbsUp Empfehlung zu einem Code-Review.

```
aws codeguru-reviewer put-recommendation-feedback \  
  --code-review-arn \\arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-  
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \  
  --recommendation-  
  id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \  
  --
```

```
--reactions ThumbsUp
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 4: Feedback geben](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [PutRecommendationFeedback](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Tag zu einem zugehörigen Repository hinzuzufügen

Im Folgenden werden einem zugehörigen Repository zwei Tags `tag-resource` hinzugefügt

```
aws codeguru-reviewer tag-resource \  
  --resource-arn arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tags Status=Secret,Team=Saanvi
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen eines Tags zu einem mit CodeGuru Reviewer verknüpften Repository \(AWS CLI\)](#) und [Hinzufügen oder Aktualisieren von Tags für ein mit CodeGuru Reviewer verbundenes Repository \(AWS CLI\)](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung eines zugehörigen Repositories aufzuheben

Im Folgenden `untag-resource` werden zwei Tags mit den Schlüsseln „Secret“ und „Team“ aus einem zugehörigen Repository entfernt.

```
aws codeguru-reviewer untag-resource \  
  --resource-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tag-keys Status Team
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tags aus einem mit CodeGuru Reviewer verknüpften Repository entfernen \(AWS CLI\)](#) im Amazon CodeGuru Reviewer-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

CodePipeline Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CodePipeline.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

acknowledge-job

Das folgende Codebeispiel zeigt die Verwendung `acknowledge-job`.

AWS CLI

Um Informationen zu einem bestimmten Job abzurufen

In diesem Beispiel werden Informationen zu einem angegebenen Auftrag zurückgegeben, einschließlich des Status dieses Auftrags, falls dieser vorhanden ist. Dies wird nur für Jobworker und benutzerdefinierte Aktionen verwendet. Verwenden Sie `aws poll-for-jobs codepipeline`, um den Wert von `nonce` und die Job-ID zu ermitteln.

Befehl:

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

Ausgabe:

```
{
  "status": "InProgress"
}
```

- API-Einheiten finden Sie [AcknowledgeJob](#) in der AWS CLI Befehlsreferenz.

create-custom-action-type

Das folgende Codebeispiel zeigt die Verwendung `create-custom-action-type`.

AWS CLI

Um eine benutzerdefinierte Aktion zu erstellen

In diesem Beispiel wird eine benutzerdefinierte Aktion für die AWS CodePipeline-Verwendung einer bereits erstellten JSON-Datei (hier `MyCustomAction.json` genannt) erstellt, die die Struktur der benutzerdefinierten Aktion enthält. Weitere Informationen zu den Anforderungen für die Erstellung einer benutzerdefinierten Aktion, einschließlich der Struktur der Datei, finden Sie im [AWS CodePipeline Benutzerhandbuch](#).

```
aws codepipeline create-custom-action-type --cli-input-json file://  
MyCustomAction.json
```

Inhalt der JSON-Datei `MyCustomAction.json`:

```
{
  "category": "Build",
  "provider": "MyJenkinsProviderName",
  "version": "1",
  "settings": {
    "entityUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/",
    "executionUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/  
lastSuccessfulBuild/{ExternalExecutionId}/"
  },
}
```



```
"configurationProperties": [  
  {  
    "name": "MyJenkinsExampleBuildProject",  
    "required": true,  
    "key": true,  
    "secret": false,  
    "queryable": false,  
    "description": "The name of the build project must be provided when this  
action is added to the pipeline.",  
    "type": "String"  
  }  
],  
"inputArtifactDetails": {  
  "maximumCount": 1,  
  "minimumCount": 0  
},  
"outputArtifactDetails": {  
  "maximumCount": 1,  
  "minimumCount": 0  
}  
}
```

Dieser Befehl gibt die Struktur der benutzerdefinierten Aktion zurück.

- API-Einheiten finden Sie [CreateCustomActionType](#) in der AWS CLI Befehlsreferenz.

create-pipeline

Das folgende Codebeispiel zeigt die Verwendung `create-pipeline`.

AWS CLI

Um eine Pipeline zu erstellen

In diesem Beispiel wird eine Pipeline AWS CodePipeline unter Verwendung einer bereits erstellten JSON-Datei (hier `MySecondPipeline.json` genannt) erstellt, die die Struktur der Pipeline enthält. Weitere Informationen zu den Anforderungen für die Erstellung einer Pipeline, einschließlich der Struktur der Datei, finden Sie im AWS CodePipeline Benutzerhandbuch.

Befehl:

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

JSONInhalt der Beispieldatei:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
              "S3Bucket": "awscodepipeline-demo-bucket",
              "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
            },
            "runOrder": 1
          }
        ]
      },
      {
        "name": "Beta",
        "actions": [
          {
            "inputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "name": "CodePipelineDemoFleet",
            "actionTypeId": {
              "category": "Deploy",
              "owner": "AWS",

```

```
        "version": "1",
        "provider": "CodeDeploy"
    },
    "outputArtifacts": [],
    "configuration": {
        "ApplicationName": "CodePipelineDemoApplication",
        "DeploymentGroupName": "CodePipelineDemoFleet"
    },
    "runOrder": 1
}
]
}
],
"artifactStore": {
    "type": "S3",
    "location": "codepipeline-us-east-1-11EXAMPLE11"
},
"name": "MySecondPipeline",
"version": 1
}
}
```

Ausgabe:

This command returns the structure of the pipeline.

- API-Einheiten finden Sie [CreatePipeline](#) in der AWS CLI Befehlsreferenz.

delete-custom-action-type

Das folgende Codebeispiel zeigt die Verwendung `delete-custom-action-type`.

AWS CLI

Um eine benutzerdefinierte Aktion zu löschen

In diesem Beispiel AWS CodePipeline wird eine benutzerdefinierte Aktion mithilfe einer bereits erstellten JSON-Datei (hier mit dem Namen „DeleteMyCustomAction.json“) gelöscht, die den Aktionstyp, den Anbieternamen und die Versionsnummer der zu löschenden Aktion enthält. Verwenden Sie den `list-action-types` Befehl, um die richtigen Werte für Kategorie, Version und Anbieter anzuzeigen.

Befehl:

```
aws codepipeline delete-custom-action-type --cli-input-json file://DeleteMyCustomAction.json
```

JSONInhalt der Beispieldatei:

```
{
  "category": "Build",
  "version": "1",
  "provider": "MyJenkinsProviderName"
}
```

Ausgabe:

None.

- APIEinzelheiten finden Sie [DeleteCustomActionType](#) in der AWS CLI Befehlsreferenz.

delete-pipeline

Das folgende Codebeispiel zeigt die Verwendung `delete-pipeline`.

AWS CLI

Um eine Pipeline zu löschen

In diesem Beispiel wird eine Pipeline mit dem Namen `MySecondPipeline` von AWS CodePipeline gelöscht. Verwenden Sie den Befehl `list-pipelines`, um eine Liste der Pipelines anzuzeigen, die Ihrem Konto zugeordnet sind. AWS

Befehl:

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

Ausgabe:

None.

- APIEinzelheiten finden Sie unter [DeletePipeline](#) Befehlsreferenz. AWS CLI

delete-webhook

Das folgende Codebeispiel zeigt die Verwendung `delete-webhook`.

AWS CLI

Um einen Webhook zu löschen

Im folgenden `delete-webhook` Beispiel wird ein Webhook für eine Quellaktion der GitHub Version 1 gelöscht. Sie müssen den `deregister-webhook-with-third-party` Befehl verwenden, um den Webhook zu deregistrieren, bevor Sie ihn löschen.

```
aws codepipeline delete-webhook \  
  --name my-webhook
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Löschen des Webhooks für Ihre GitHub Quelle](#).AWS CodePipeline

- API-Einheiten finden Sie [DeleteWebhook](#) in der AWS CLI Befehlsreferenz.

deregister-webhook-with-third-party

Das folgende Codebeispiel zeigt die Verwendung `deregister-webhook-with-third-party`.

AWS CLI

Um einen Webhook abzumelden

Im folgenden `deregister-webhook-with-third-party` Beispiel wird ein Webhook für eine GitHub Quellaktion der Version 1 gelöscht. Sie müssen den Webhook abmelden, bevor Sie ihn löschen.

```
aws codepipeline deregister-webhook-with-third-party \  
  --webhook-name my-webhook
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [im Benutzerhandbuch unter Löschen des Webhooks für Ihre GitHub Quelle](#).AWS CodePipeline

- API-Einheiten finden Sie [DeregisterWebhookWithThirdParty](#) in der AWS CLI Befehlsreferenz.

disable-stage-transition

Das folgende Codebeispiel zeigt die Verwendung `disable-stage-transition`.

AWS CLI

Um einen Übergang zu einer Phase in einer Pipeline zu deaktivieren

In diesem Beispiel werden Übergänge in die Beta-Phase der `MyFirstPipeline` Pipeline in AWS CodePipeline deaktiviert.

Befehl:

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [DisableStageTransition](#) in der AWS CLI Befehlsreferenz.

enable-stage-transition

Das folgende Codebeispiel zeigt die Verwendung `enable-stage-transition`.

AWS CLI

Um einen Übergang zu einer Phase in einer Pipeline zu ermöglichen

Dieses Beispiel ermöglicht Übergänge in die Beta-Phase der `MyFirstPipeline` Pipeline in AWS CodePipeline.

Befehl:

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

Ausgabe:

None.

- API-Einheiten finden Sie [EnableStageTransition](#) in der AWS CLI Befehlsreferenz.

get-job-details

Das folgende Codebeispiel zeigt die Verwendung `get-job-details`.

AWS CLI

Um Details zu einem Job abzurufen

In diesem Beispiel werden Details zu einem Job zurückgegeben, dessen ID durch `f4f4ff82-2d11-` dargestellt wird. **EXAMPLE** Dieser Befehl wird nur für benutzerdefinierte Aktionen verwendet. Wenn dieser Befehl aufgerufen wird, werden temporäre Anmeldeinformationen für den Amazon S3 S3-Bucket AWS CodePipeline zurückgegeben, der zum Speichern von Artefakten für die Pipeline verwendet wird, falls dies für die benutzerdefinierte Aktion erforderlich ist. Dieser Befehl gibt auch alle geheimen Werte zurück, die für die Aktion definiert wurden, sofern welche definiert wurden.

Befehl:

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

Ausgabe:

```
{
  "jobDetails": {
    "accountId": "111111111111",
    "data": {
      "actionConfiguration": {
        "__type": "ActionConfiguration",
        "configuration": {
          "ProjectName": "MyJenkinsExampleTestProject"
        }
      },
      "actionTypeId": {
        "__type": "ActionTypeId",
        "category": "Test",
        "owner": "Custom",
```

```

    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "artifactCredentials": {
    "__type": "AWSSessionCredentials",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "sessionToken":
    "fICCQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs
  },
  "inputArtifacts": [
    {
      "__type": "Artifact",
      "location": {
        "s3Location": {
          "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
          "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
        },
        "type": "S3"
      },
      "name": "MyAppBuild"
    }
  ],
  "outputArtifacts": [],
  "pipelineContext": {
    "__type": "PipelineContext",
    "action": {
      "name": "MyJenkinsTest-Action"
    },
    "pipelineName": "MySecondPipeline",
    "stage": {
      "name": "Testing"
    }
  }
},
"id": "f4f4ff82-2d11-EXAMPLE"
}
}

```

- API-Einheiten finden Sie [GetJobDetails](#) in der AWS CLI Befehlsreferenz.

get-pipeline-state

Das folgende Codebeispiel zeigt die Verwendung `get-pipeline-state`.

AWS CLI

Um Informationen über den Status einer Pipeline zu erhalten

In diesem Beispiel wird der letzte Status einer Pipeline mit dem Namen zurückgegeben `MyFirstPipeline`.

Befehl:

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

Ausgabe:

```
{
  "created": 1446137312.204,
  "pipelineName": "MyFirstPipeline",
  "pipelineVersion": 1,
  "stageStates": [
    {
      "actionStates": [
        {
          "actionName": "Source",
          "entityUrl": "https://console.aws.amazon.com/s3/home?#",
          "latestExecution": {
            "lastStatusChange": 1446137358.328,
            "status": "Succeeded"
          }
        }
      ],
      "stageName": "Source"
    },
    {
      "actionStates": [
        {
          "actionName": "CodePipelineDemoFleet",
          "entityUrl": "https://console.aws.amazon.com/codedeploy/home?#/applications/CodePipelineDemoApplication/deployment-groups/CodePipelineDemoFleet",
          "latestExecution": {
            "externalExecutionId": "d-EXAMPLE",

```

```
        "externalExecutionUrl": "https://console.aws.amazon.com/codedeploy/home?#/
deployments/d-EXAMPLE",
        "lastStatusChange": 1446137493.131,
        "status": "Succeeded",
        "summary": "Deployment Succeeded"
    }
  ],
  "inboundTransitionState": {
    "enabled": true
  },
  "stageName": "Beta"
}
],
"updated": 1446137312.204
}
```

- API-Einheiten finden Sie [GetPipelineState](#) unter AWS CLI Befehlsreferenz.

get-pipeline

Das folgende Codebeispiel zeigt die Verwendung `get-pipeline`.

AWS CLI

Um die Struktur einer Pipeline anzuzeigen

In diesem Beispiel wird die Struktur einer Pipeline mit dem Namen zurückgegeben `MyFirstPipeline`.

Befehl:

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

Ausgabe:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "name": "Source",
        "actions": [
```

```
    {
      "inputArtifacts": [],
      "name": "Source",
      "actionTypeId": {
        "category": "Source",
        "owner": "AWS",
        "version": "1",
        "provider": "S3"
      },
      "outputArtifacts": [
        {
          "name": "MyApp"
        }
      ],
      "configuration": {
        "S3Bucket": "awscodepipeline-demo-bucket",
        "S3ObjectKey": "aws-codepipeline-s3-aws-
codedeploy_linux.zip"
      },
      "runOrder": 1
    }
  ],
},
{
  "name": "Beta",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "MyApp"
        }
      ],
      "name": "CodePipelineDemoFleet",
      "actionTypeId": {
        "category": "Deploy",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeDeploy"
      },
      "outputArtifacts": [],
      "configuration": {
        "ApplicationName": "CodePipelineDemoApplication",
        "DeploymentGroupName": "CodePipelineDemoFleet"
      },
    }
  ]
}
```

```

        "runOrder": 1
      }
    ]
  },
  "artifactStore": {
    "type": "S3",
    "location": "codepipeline-us-east-1-11EXAMPLE11"
  },
  "name": "MyFirstPipeline",
  "version": 1
}
}

```

- API-Einheiten finden Sie [GetPipeline](#) unter AWS CLI Befehlsreferenz.

list-action-executions

Das folgende Codebeispiel zeigt die Verwendung `list-action-executions`.

AWS CLI

Um die Ausführung von Aktionen aufzulisten

Im folgenden `list-action-executions` Beispiel werden Details zur Aktionsausführung für eine Pipeline angezeigt, z. B. die Aktionsausführungs-ID, Eingabeartefakte, Ausgabeartefakte, Ausführungsergebnis und Status.

```
aws codepipeline list-action-executions \
  --pipeline-name myPipeline
```

Ausgabe:

```
{
  "actionExecutionDetails": [
    {
      "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
      "actionExecutionId": "EXAMPLE4-2ee8-4853-bd6a-111111158148",
      "pipelineVersion": 12,
      "stageName": "Deploy",
      "actionName": "Deploy",
      "startTime": 1598572628.6,

```

```
"lastUpdateTime": 1598572661.255,
"status": "Succeeded",
"input": {
  "actionTypeId": {
    "category": "Deploy",
    "owner": "AWS",
    "provider": "CodeDeploy",
    "version": "1"
  },
  "configuration": {
    "ApplicationName": "my-application",
    "DeploymentGroupName": "my-deployment-group"
  },
  "resolvedConfiguration": {
    "ApplicationName": "my-application",
    "DeploymentGroupName": "my-deployment-group"
  },
  "region": "us-east-1",
  "inputArtifacts": [
    {
      "name": "SourceArtifact",
      "s3location": {
        "bucket": "artifact-bucket",
        "key": "myPipeline/SourceArti/key"
      }
    }
  ],
  "namespace": "DeployVariables"
},
"output": {
  "outputArtifacts": [],
  "executionResult": {
    "externalExecutionId": "d-EXAMPLEE5",
    "externalExecutionSummary": "Deployment Succeeded",
    "externalExecutionUrl": "https://myaddress.com"
  },
  "outputVariables": {}
}
},
{
  "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
  "actionExecutionId": "EXAMPLE5-abb4-4192-9031-11111113a7b0",
  "pipelineVersion": 12,
  "stageName": "Source",
```

```
"actionName": "Source",
"startTime": 1598572624.387,
"lastUpdateTime": 1598572628.16,
"status": "Succeeded",
"input": {
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "provider": "CodeCommit",
    "version": "1"
  },
  "configuration": {
    "BranchName": "production",
    "PollForSourceChanges": "false",
    "RepositoryName": "my-repo"
  },
  "resolvedConfiguration": {
    "BranchName": "production",
    "PollForSourceChanges": "false",
    "RepositoryName": "my-repo"
  },
  "region": "us-east-1",
  "inputArtifacts": [],
  "namespace": "SourceVariables"
},
"output": {
  "outputArtifacts": [
    {
      "name": "SourceArtifact",
      "s3location": {
        "bucket": "my-bucket",
        "key": "myPipeline/SourceArti/key"
      }
    }
  ],
  "executionResult": {
    "externalExecutionId":
"1111111ad99dcd35914c00b7fba13995EXAMPLE",
    "externalExecutionSummary": "Edited template.yml",
    "externalExecutionUrl": "https://myaddress.com"
  },
  "outputVariables": {
    "AuthorDate": "2020-05-08T17:45:43Z",
    "BranchName": "production",
```

```
        "CommitId": "EXAMPLEad99dcd35914c00b7f7bea139951111111",
        "CommitMessage": "Edited template.yml",
        "CommitterDate": "2020-05-08T17:45:43Z",
        "RepositoryName": "my-repo"
      }
    },
    . . . .
```

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Aktionausführungen anzeigen \(CLI\)](#).

- API-Einheiten finden Sie [ListActionExecutions](#) in der AWS CLI Befehlsreferenz.

list-action-types

Das folgende Codebeispiel zeigt die Verwendung `list-action-types`.

AWS CLI

Um die verfügbaren Aktionstypen anzuzeigen

Der `list-action-types` Befehl wird eigenständig verwendet und gibt die Struktur aller Aktionen zurück, die für Ihr AWS Konto verfügbar sind. In diesem Beispiel wird die `action-owner-filter` Option `--` verwendet, um nur benutzerdefinierte Aktionen zurückzugeben.

Befehl:

```
aws codepipeline list-action-types --action-owner-filter Custom
```

Ausgabe:

```
{
  "actionTypes": [
    {
      "inputArtifactDetails": {
        "maximumCount": 5,
        "minimumCount": 0
      },
      "actionConfigurationProperties": [
        {
          "secret": false,
          "required": true,
```

```
        "name": "MyJenkinsExampleBuildProject",
        "key": true,
        "queryable": true
    }
],
"outputArtifactDetails": {
    "maximumCount": 5,
    "minimumCount": 0
},
"id": {
    "category": "Build",
    "owner": "Custom",
    "version": "1",
    "provider": "MyJenkinsProviderName"
},
"settings": {
    "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
    "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
}
},
{
    "inputArtifactDetails": {
        "maximumCount": 5,
        "minimumCount": 0
    },
    "actionConfigurationProperties": [
        {
            "secret": false,
            "required": true,
            "name": "MyJenkinsExampleTestProject",
            "key": true,
            "queryable": true
        }
    ],
    "outputArtifactDetails": {
        "maximumCount": 5,
        "minimumCount": 0
    },
    "id": {
        "category": "Test",
        "owner": "Custom",
        "version": "1",
        "provider": "MyJenkinsProviderName"
    }
}
```



```

    },
    "settings": {
      "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
      "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
    }
  }
]
}

```

- API-Einheiten finden Sie [ListActionTypes](#) in der AWS CLI Befehlsreferenz.

list-pipeline-executions

Das folgende Codebeispiel zeigt die Verwendung `list-pipeline-executions`.

AWS CLI

Um den Verlauf der Pipeline-Ausführung anzuzeigen

Das folgende `list-pipeline-executions` Beispiel zeigt den Verlauf der Pipeline-Ausführung für eine Pipeline in Ihrem AWS Konto.

```

aws codepipeline list-pipeline-executions \
  --pipeline-name MyPipeline

```

Ausgabe:

```

{
  "pipelineExecutionSummaries": [
    {
      "lastUpdateTime": 1496380678.648,
      "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
      "startTime": 1496380258.243,
      "status": "Succeeded"
    },
    {
      "lastUpdateTime": 1496591045.634,
      "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d841-d7eu3EXAMPLE",
      "startTime": 1496590401.222,
      "status": "Succeeded"
    }
  ],
}

```

```
    {
      "lastUpdateTime": 1496946071.6456,
      "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
      "startTime": 1496945471.5645,
      "status": "Succeeded"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Ausführungsverlauf anzeigen](#).

- API-Einheiten finden Sie [ListPipelineExecutions](#) in der AWS CLI Befehlsreferenz.

list-pipelines

Das folgende Codebeispiel zeigt die Verwendung `list-pipelines`.

AWS CLI

Um eine Liste von Pipelines anzuzeigen

In diesem Beispiel werden alle AWS CodePipeline Pipelines aufgeführt, die dem Konto des Benutzers AWS zugeordnet sind.

Befehl:

```
aws codepipeline list-pipelines
```

Ausgabe:

```
{
  "pipelines": [
    {
      "updated": 1439504274.641,
      "version": 1,
      "name": "MyFirstPipeline",
      "created": 1439504274.641
    },
    {
      "updated": 1436461837.992,
      "version": 2,
      "name": "MySecondPipeline",

```

```
        "created": 1436460801.381
      }
    ]
  }
```

- API-Einheiten finden Sie unter [List Pipelines AWS CLI Befehlsreferenz](#).

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags aufzulisten

Im folgenden `list-tags-for-resource` Beispiel wird eine Liste aller Tags abgerufen, die an die angegebene Pipeline-Ressource angehängt sind.

```
aws codepipeline list-tags-for-resource \
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

Ausgabe:

```
{
  "tags": {
    "Project": "ProjectA",
    "IscontainerBased": "true"
  }
}
```

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Tags für eine Pipeline anzeigen \(CLI\)](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-webhooks

Das folgende Codebeispiel zeigt die Verwendung `list-webhooks`.

AWS CLI

Um Webhooks aufzulisten

Im folgenden `list-webhooks` Beispiel wird eine Liste aller Tags abgerufen, die an die angegebene Pipeline-Ressource angehängt sind.

```
aws codepipeline list-webhooks \
  --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \
  --region "eu-central-1"
```

Ausgabe:

```
{
  "webhooks": [
    {
      "url": "https://webhooks.domain.com/
trigger1111111111EXAMPLE1111111111111111111": {
      "authenticationConfiguration": {
        "SecretToken": "Secret"
      },
      "name": "my-webhook",
      "authentication": "GITHUB_HMAC",
      "targetPipeline": "my-Pipeline",
      "targetAction": "Source",
      "filters": [
        {
          "jsonPath": "$.ref",
          "matchEquals": "refs/heads/{Branch}"
        }
      ]
    },
    "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-
webhook"
  ]
}
```

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Auflisten von Webhooks in Ihrem Konto](#).

- API-Einheiten finden Sie [ListWebhooks](#) in der AWS CLI Befehlsreferenz.

poll-for-jobs

Das folgende Codebeispiel zeigt die Verwendung `poll-for-jobs`.

AWS CLI

Um alle verfügbaren Jobs anzuzeigen

In diesem Beispiel werden Informationen über alle Jobs zurückgegeben, auf die ein Jobarbeiter reagieren kann. In diesem Beispiel wird eine vordefinierte JSON Datei (`MyActionTypeInfo.json`) verwendet, um Informationen über den Aktionstyp bereitzustellen, für den der Jobworker Jobs verarbeitet. Dieser Befehl wird nur für benutzerdefinierte Aktionen verwendet. Wenn dieser Befehl aufgerufen wird, werden temporäre Anmeldeinformationen für den Amazon S3 S3-Bucket AWS CodePipeline zurückgegeben, der zum Speichern von Artefakten für die Pipeline verwendet wird. Dieser Befehl gibt auch alle geheimen Werte zurück, die für die Aktion definiert wurden, sofern welche definiert wurden.

Befehl:

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

JSONInhalt der Beispieldatei:

```
{
  "actionTypeId": {
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
    "ProjectName": "MyJenkinsTestProject"
  }
}
```

Ausgabe:

```
{
  "jobs": [
    {
      "accountId": "111111111111",
      "data": {
        "actionConfiguration": {
          "__type": "ActionConfiguration",
          "configuration": {
```

```

    "ProjectName": "MyJenkinsExampleTestProject"
  }
},
"actionTypeId": {
  "__type": "ActionTypeId",
  "category": "Test",
  "owner": "Custom",
  "provider": "MyJenkinsProviderName",
  "version": "1"
},
"artifactCredentials": {
  "__type": "AWSSessionCredentials",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "sessionToken":
    "fICCCQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRawDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkBFbJvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs
  },
"inputArtifacts": [
  {
    "__type": "Artifact",
    "location": {
      "s3Location": {
        "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
        "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
      },
      "type": "S3"
    },
    "name": "MyAppBuild"
  }
],
"outputArtifacts": [],
"pipelineContext": {
  "__type": "PipelineContext",
  "action": {
    "name": "MyJenkinsTest-Action"
  },
  "pipelineName": "MySecondPipeline",
  "stage": {
    "name": "Testing"
  }
}

```

```
    }
  },
  "id": "ef66c259-64f9-EXAMPLE",
  "nonce": "3"
}
]
```

- API-Einheiten finden Sie [PollForJobs](#) in der AWS CLI Befehlsreferenz.

put-webhook

Das folgende Codebeispiel zeigt die Verwendung `put-webhook`.

AWS CLI

Um einen Webhook zu erstellen

Im folgenden `put-webhook` Beispiel wird ein Webhook für eine Quellaktion der GitHub Version 1 erstellt. Nachdem Sie den Webhook erstellt haben, müssen Sie ihn mit dem Befehl `register-webhook-with-third-party` registrieren.

```
aws codepipeline put-webhook \
  --cli-input-json file://webhook_json.json \
  --region "eu-central-1"
```

Inhalt von `webhook_json.json`:

```
{
  "webhook": {
    "name": "my-webhook",
    "targetPipeline": "pipeline_name",
    "targetAction": "source_action_name",
    "filters": [
      {
        "jsonPath": "$.ref",
        "matchEquals": "refs/heads/{Branch}"
      }
    ],
    "authentication": "GITHUB_HMAC",
    "authenticationConfiguration": {
      "SecretToken": "secret"
    }
  }
}
```

```

    }
  }
}

```

Ausgabe:

```

{
  "webhook": {
    "url": "https://webhooks.domain.com/
trigger1111111111EXAMPLE1111111111111111111",
    "definition": {
      "authenticationConfiguration": {
        "SecretToken": "secret"
      },
      "name": "my-webhook",
      "authentication": "GITHUB_HMAC",
      "targetPipeline": "pipeline_name",
      "targetAction": "Source",
      "filters": [
        {
          "jsonPath": "$.ref",
          "matchEquals": "refs/heads/{Branch}"
        }
      ]
    },
    "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-webhook"
  },
  "tags": [
    {
      "key": "Project",
      "value": "ProjectA"
    }
  ]
}

```

Weitere Informationen finden Sie im [Benutzerhandbuch unter Erstellen eines Webhooks für eine GitHub Quelle](#).AWS CodePipeline

- API-Einheiten finden Sie [PutWebhook](#) in der AWS CLI Befehlsreferenz.

retry-stage-execution

Das folgende Codebeispiel zeigt die Verwendung `retry-stage-execution`.

AWS CLI

Um eine fehlgeschlagene Aktion erneut zu versuchen

Im folgenden `retry-stage-execution` Beispiel wird eine Phase wiederholt, in der eine Aktion fehlgeschlagen ist.

```
aws codepipeline retry-stage-execution \  
  --pipeline-name MyPipeline \  
  --stage-name Deploy \  
  --pipeline-execution-id b59babff-5f34-EXAMPLE \  
  --retry-mode FAILED_ACTIONS
```

Ausgabe:

```
{  
  "pipelineExecutionId": "b59babff-5f34-EXAMPLE"  
}
```

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Fehlgeschlagene Aktionen CLI wiederholen \(\)](#).

- API-Einheiten finden Sie [RetryStageExecution](#) in der AWS CLI Befehlsreferenz.

start-pipeline-execution

Das folgende Codebeispiel zeigt die Verwendung `start-pipeline-execution`.

AWS CLI

Um die neueste Revision über eine Pipeline auszuführen

In diesem Beispiel wird die neueste Version, die sich in der Quellphase einer Pipeline befindet, über die Pipeline mit dem Namen "MyFirstPipeline" ausgeführt.

Befehl:

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

Ausgabe:

```
{
```

```
"pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"
}
```

- API-Einheiten finden Sie [StartPipelineExecution](#) unter AWS CLI Befehlsreferenz.

stop-pipeline-execution

Das folgende Codebeispiel zeigt die Verwendung `stop-pipeline-execution`.

AWS CLI

Um die Ausführung einer Pipeline zu beenden

Im folgenden `stop-pipeline-execution` Beispiel wird standardmäßig gewartet, bis die laufenden Aktionen abgeschlossen sind, und dann die Pipeline gestoppt. Sie können sich nicht für das Anhalten und Warten entscheiden, wenn sich die Ausführung bereits in einem Stopping (Wird angehalten)-Status befindet. Sie können eine Ausführung, die sich bereits in einem Status Stopping (Wird angehalten) befindet, anhalten und beenden.

```
aws codepipeline stop-pipeline-execution \
  --pipeline-name MyFirstPipeline \
  --pipeline-execution-id d-EXAMPLE \
  --reason "Stopping pipeline after the build action is done"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Stoppen der Ausführung einer Pipeline \(CLI\)](#).

- API-Einheiten finden Sie [StopPipelineExecution](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Das folgende `tag-resource` Beispiel verknüpft einen Satz bereitgestellter Tags mit einer Pipeline. Verwenden Sie diesen Befehl, um Tags hinzuzufügen oder zu bearbeiten.

```
aws codepipeline tag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Hinzufügen von Tags zu einer Pipeline \(CLI\)](#) im AWS CodePipeline Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um AWS Tags aus einer Verbindungsressource zu entfernen

Im folgenden `untag-resource` Beispiel wird ein Tag aus der angegebenen Ressource entfernt.

```
aws codepipeline untag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tag-keys Project IscontainerBased
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CodePipeline Benutzerhandbuch unter [Tags aus einer Pipeline entfernen \(CLI\)](#).

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-pipeline

Das folgende Codebeispiel zeigt die Verwendung `update-pipeline`.

AWS CLI

Um die Struktur einer Pipeline zu aktualisieren

In diesem Beispiel wird der Befehl `update-pipeline` mit dem Argument `--cli-input-json` verwendet. In diesem Beispiel wird eine vordefinierte JSON-Datei (`MyFirstPipeline.json`) verwendet, um

die Struktur einer Pipeline zu aktualisieren. AWS CodePipeline erkennt den in der JSON Datei enthaltenen Pipeline-Namen und wendet dann alle Änderungen an geänderten Feldern in der Pipeline-Struktur an, um die Pipeline zu aktualisieren.

Beachten Sie beim Erstellen der vordefinierten JSON Datei die folgenden Richtlinien:

Wenn Sie mit einer Pipeline-Struktur arbeiten, die mit dem Befehl `get-pipeline` abgerufen wurde, müssen Sie den Metadatenabschnitt aus der Pipeline-Struktur in der JSON Datei entfernen (die Zeilen `„metadata“: {}` und die darin enthaltenen Felder `„created“`, `„ARN“`, `„pipeline“` und `„updated“`). Der Pipelinename kann nicht geändert werden.

Befehl:

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

Inhalt der JSON Beispieldatei:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
              "S3Bucket": "awscodepipeline-demo-bucket2",
              "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
            }
          }
        ]
      }
    ]
  }
}
```

```

        "runOrder": 1
      }
    ]
  },
  {
    "name": "Beta",
    "actions": [
      {
        "inputArtifacts": [
          {
            "name": "MyApp"
          }
        ],
        "name": "CodePipelineDemoFleet",
        "actionTypeId": {
          "category": "Deploy",
          "owner": "AWS",
          "version": "1",
          "provider": "CodeDeploy"
        },
        "outputArtifacts": [],
        "configuration": {
          "ApplicationName": "CodePipelineDemoApplication",
          "DeploymentGroupName": "CodePipelineDemoFleet"
        },
        "runOrder": 1
      }
    ]
  }
],
"artifactStore": {
  "type": "S3",
  "location": "codepipeline-us-east-1-11EXAMPLE11"
},
"name": "MyFirstPipeline",
"version": 1
}
}

```

Ausgabe:

```

{
  "pipeline": {

```

```
"artifactStore": {
  "location": "codepipeline-us-east-1-11EXAMPLE11",
  "type": "S3"
},
"name": "MyFirstPipeline",
"roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
"stages": [
  {
    "actions": [
      {
        "actionTypeId": {
          "__type": "ActionTypeId",
          "category": "Source",
          "owner": "AWS",
          "provider": "S3",
          "version": "1"
        },
        "configuration": {
          "S3Bucket": "awscodepipeline-demo-bucket2",
          "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
        },
        "inputArtifacts": [],
        "name": "Source",
        "outputArtifacts": [
          {
            "name": "MyApp"
          }
        ],
        "runOrder": 1
      }
    ],
    "name": "Source"
  },
  {
    "actions": [
      {
        "actionTypeId": {
          "__type": "ActionTypeId",
          "category": "Deploy",
          "owner": "AWS",
          "provider": "CodeDeploy",
          "version": "1"
        },
        "configuration": {
```

```
    "ApplicationName": "CodePipelineDemoApplication",
    "DeploymentGroupName": "CodePipelineDemoFleet"
  },
  "inputArtifacts": [
    {
      "name": "MyApp"
    }
  ],
  "name": "CodePipelineDemoFleet",
  "outputArtifacts": [],
  "runOrder": 1
}
],
"name": "Beta"
}
],
"version": 3
}
}
```

- API-Einheiten finden Sie [UpdatePipeline](#) in der AWS CLI Befehlsreferenz.

AWS CodeStar Beispiele für Benachrichtigungen mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with AWS CodeStar Notifications Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-notification-rule

Das folgende Codebeispiel zeigt die Verwendung `create-notification-rule`.

AWS CLI

Um eine Benachrichtigungsregel zu erstellen

Das folgende `create-notification-rule` Beispiel verwendet eine JSON Datei mit dem Namen `rule.json`, um eine Benachrichtigungsregel zu erstellen, `MyNotificationRule` die nach einem Repository benannt ist, das `MyDemoRepo` im angegebenen AWS Konto benannt ist. Benachrichtigungen mit dem FULL Detailtyp werden an das angegebene SNS Amazon-Zielthema gesendet, wenn Branches und Tags erstellt werden.

```
aws codestar-notifications create-notification-rule \  
--cli-input-json file://rule.json
```

Inhalt von `rule.json`:

```
{  
  "Name": "MyNotificationRule",  
  "EventTypeIds": [  
    "codecommit-repository-branches-and-tags-created"  
  ],  
  "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",  
  "Targets": [  
    {  
      "TargetType": "SNS",  
      "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"  
    }  
  ],  
  "Status": "ENABLED",  
  "DetailType": "FULL"  
}
```

Ausgabe:

```
{
```



```
"Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Eine Benachrichtigungsregel erstellen](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateNotificationRule](#) unter AWS CLI Befehlsreferenz.

delete-notification-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-notification-rule`.

AWS CLI

Um eine Benachrichtigungsregel zu löschen

Im folgenden `delete-notification-rule` Beispiel wird die angegebene Benachrichtigungsregel gelöscht.

```
aws codestar-notifications delete-notification-rule \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

Ausgabe:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Löschen einer Benachrichtigungsregel](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteNotificationRule](#) unter AWS CLI Befehlsreferenz.

delete-target

Das folgende Codebeispiel zeigt die Verwendung `delete-target`.

AWS CLI

Um ein Ziel für eine Benachrichtigungsregel zu löschen

Im folgenden `delete-target` Beispiel wird das angegebene Ziel aus allen Benachrichtigungsregeln entfernt, die so konfiguriert sind, dass es als Ziel verwendet wird. Anschließend wird das Ziel gelöscht.

```
aws codestar-notifications delete-target \  
  --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \  
  --force-unsubscribe-all
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Benachrichtigungsregelziels](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteTarget](#) unter AWS CLI Befehlsreferenz.

describe-notification-rule

Das folgende Codebeispiel zeigt die Verwendung `describe-notification-rule`.

AWS CLI

Um Details einer Benachrichtigungsregel abzurufen

Im folgenden `describe-notification-rule` Beispiel werden die Details der angegebenen Benachrichtigungsregel abgerufen.

```
aws codestar-notifications describe-notification-rule \  
  --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/dc82df7a-EXAMPLE
```

Ausgabe:

```
{  
  "LastModifiedTimestamp": 1569199844.857,  
  "EventTypes": [  
    {  
      "ServiceName": "CodeCommit",  
      "EventTypeName": "Branches and tags: Created",
```

```

        "ResourceType": "Repository",
        "EventTypeId": "codecommit-repository-branches-and-tags-created"
    }
],
"Status": "ENABLED",
"DetailType": "FULL",
"Resource": "arn:aws:codecommit:us-west-2:123456789012:MyDemoRepo",
"Arn": "arn:aws:codestar-notifications:us-west-w:123456789012:notificationrule/dc82df7a-EXAMPLE",
"Targets": [
    {
        "TargetStatus": "ACTIVE",
        "TargetAddress": "arn:aws:sns:us-west-2:123456789012:MyNotificationTopic",
        "TargetType": "SNS"
    }
],
"Name": "MyNotificationRule",
"CreatedTimestamp": 1569199844.857,
"CreatedBy": "arn:aws:iam::123456789012:user/Mary_Major"
}

```

Weitere Informationen finden Sie unter [Benachrichtigungsregeln anzeigen](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeNotificationRule](#) in der AWS CLI Befehlsreferenz.

list-event-types

Das folgende Codebeispiel zeigt die Verwendung `list-event-types`.

AWS CLI

Um eine Liste von Ereignistypen für eine Benachrichtigungsregel abzurufen

Im folgenden `list-event-types` Beispiel wird eine gefilterte Liste aller verfügbaren Benachrichtigungsereignistypen für CodeDeploy Anwendungen abgerufen. Wenn Sie stattdessen keinen Filter verwenden, gibt der Befehl alle Benachrichtigungsereignistypen für alle Ressourcentypen zurück.

```

aws codestar-notifications list-event-types \
  --filters Name=SERVICE_NAME,Value=CodeDeploy

```

Ausgabe:

```
{
  "EventTypes": [
    {
      "EventTypeId": "codedeploy-application-deployment-succeeded",
      "ServiceName": "CodeDeploy",
      "EventTypeName": "Deployment: Succeeded",
      "ResourceType": "Application"
    },
    {
      "EventTypeId": "codedeploy-application-deployment-failed",
      "ServiceName": "CodeDeploy",
      "EventTypeName": "Deployment: Failed",
      "ResourceType": "Application"
    },
    {
      "EventTypeId": "codedeploy-application-deployment-started",
      "ServiceName": "CodeDeploy",
      "EventTypeName": "Deployment: Started",
      "ResourceType": "Application"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erstellen einer Benachrichtigungsregel](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [ListEventTypes](#) unter AWS CLI Befehlsreferenz.

list-notification-rules

Das folgende Codebeispiel zeigt die Verwendung `list-notification-rules`.

AWS CLI

Um eine Liste von Benachrichtigungsregeln abzurufen

Im folgenden `list-notification-rules` Beispiel wird eine Liste aller Benachrichtigungsregeln in der angegebenen AWS Region abgerufen.

```
aws codestar-notifications list-notification-rules --region us-east-1
```

Ausgabe:

```
{
  "NotificationRules": [
    {
      "Id": "dc82df7a-EXAMPLE",
      "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
    },
    {
      "Id": "8d1f0983-EXAMPLE",
      "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/8d1f0983-EXAMPLE"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Benachrichtigungsregeln anzeigen](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [ListNotificationRules](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um eine Liste von Tags abzurufen, die an eine Benachrichtigungsregel angehängt sind

Im folgenden `list-tags-for-resource` Beispiel wird eine Liste aller Tags abgerufen, die der angegebenen Benachrichtigungsregel zugeordnet sind. In diesem Beispiel sind der Benachrichtigungsregel derzeit keine Tags zugeordnet.

```
aws codestar-notifications list-tags-for-resource \
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/fe1efd35-EXAMPLE
```

Ausgabe:

```
{
  "Tags": {}
}
```

```
}
```

Weitere Informationen finden Sie unter [Erstellen einer Benachrichtigungsregel](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

list-targets

Das folgende Codebeispiel zeigt die Verwendung `list-targets`.

AWS CLI

Um eine Liste von Zielen für Benachrichtigungsregeln abzurufen

Im folgenden `list-targets` Beispiel wird eine Liste aller Ziele für Benachrichtigungsregeln in der angegebenen AWS Region abgerufen.

```
aws codestar-notifications list-targets \  
  --region us-east-1
```

Ausgabe:

```
{  
  "Targets": [  
    {  
      "TargetAddress": "arn:aws:sns:us-  
east-1:123456789012:MySNSTopicForNotificationRules",  
      "TargetType": "SNS",  
      "TargetStatus": "ACTIVE"  
    },  
    {  
      "TargetAddress": "arn:aws:sns:us-  
east-1:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo",  
      "TargetType": "SNS",  
      "TargetStatus": "ACTIVE"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Ziele für Benachrichtigungsregeln anzeigen](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTargets](#) unter AWS CLI Befehlsreferenz.

subscribe

Das folgende Codebeispiel zeigt die Verwendung `subscribe`.

AWS CLI

Um ein Ziel zu einer Benachrichtigungsregel hinzuzufügen

Im folgenden `subscribe` Beispiel wird ein SNS Amazon-Thema als Ziel für die angegebene Benachrichtigungsregel hinzugefügt.

```
aws codestar-notifications subscribe \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE \  
  --target TargetType=SNS,TargetAddress=arn:aws:sns:us-  
east-1:123456789012:MyNotificationTopic
```

Ausgabe:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Ein SNS Amazon-Thema als Ziel für eine Benachrichtigungsregel hinzufügen oder entfernen](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie unter [Abonnieren](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer Benachrichtigungsregel ein Tag hinzuzufügen

Im folgenden `tag-resource` Beispiel wird der angegebenen Benachrichtigungsregel ein Tag mit dem Schlüsselnamen `Team` und `Li_Juan` dem Wert von hinzugefügt.

```
aws codestar-notifications tag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tags Team=Li_Juan
```

Ausgabe:

```
{  
  "Tags": {  
    "Team": "Li_Juan"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Benachrichtigungsregel](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

unsubscribe

Das folgende Codebeispiel zeigt die Verwendung von `unsubscribe`.

AWS CLI

Um ein Ziel aus einer Benachrichtigungsregel zu entfernen

Im folgenden `unsubscribe` Beispiel wird ein SNS Amazon-Thema als Ziel aus der angegebenen Benachrichtigungsregel entfernt.

```
aws codestar-notifications unsubscribe \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE \  
  --target TargetType=SNS,TargetAddress=arn:aws:sns:us-  
east-1:123456789012:MyNotificationTopic
```

Ausgabe:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
  "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"  
}
```



```
}
```

Weitere Informationen finden Sie unter [Ein SNS Amazon-Thema als Ziel für eine Benachrichtigungsregel hinzufügen oder entfernen](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie unter [Abbestellen](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Benachrichtigungsregel zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem Schlüsselnamen `Team` aus der angegebenen Benachrichtigungsregel entfernt.

```
aws codestar-notifications untag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tag-keys Team
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Bearbeiten einer Benachrichtigungsregel](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-notification-rule

Das folgende Codebeispiel zeigt die Verwendung `update-notification-rule`.

AWS CLI

Um eine Benachrichtigungsregel zu aktualisieren

Im folgenden `update-notification-rule` Beispiel wird eine `MyNotificationRule` im AWS Konto angegebene Benachrichtigungsregel `123456789012` mithilfe einer JSON-Datei mit dem Namen `aktualisiertupdate.json`.

```
aws codestar-notifications update-notification-rule \  
--cli-input-json file://update.json
```

Inhalt von `update.json`:

```
{  
  "Name": "MyUpdatedNotificationRule",  
  "EventTypeIds": [  
    "codecommit-repository-branches-and-tags-created"  
  ],  
  "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",  
  "Targets": [  
    {  
      "TargetType": "SNS",  
      "TargetAddress": "arn:aws:sns:us-  
east-1:123456789012:MyNotificationTopic"  
    }  
  ],  
  "Status": "ENABLED",  
  "DetailType": "FULL"  
}
```

Ausgabe:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Bearbeiten einer Benachrichtigungsregel](#) im AWS Developer Tools Console-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateNotificationRule](#) unter AWS CLI Befehlsreferenz.

CodeConnections Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren CodeConnections.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-connection

Das folgende Codebeispiel zeigt die Verwendung `create-connection`.

AWS CLI

Um eine Verbindung herzustellen

Das folgende `create-connection` Beispiel zeigt, wie eine Verbindung zu einem Repository eines Drittanbieters hergestellt wird. In diesem Beispiel wird eine Verbindung hergestellt, bei der der Drittanbieter Bitbucket ist.

Eine Verbindung, die über AWS CLI oder AWS CloudFormation erstellt wurde, hat standardmäßig den Status Ausstehend. Nachdem Sie eine Verbindung mit dem CLI oder hergestellt haben AWS CloudFormation, bearbeiten Sie die Verbindung in der Konsole, sodass ihr Status Verfügbar zugewiesen wird.

```
aws codestar-connections create-connection \  
  --provider-type Bitbucket \  
  --connection-name MyConnection
```

Ausgabe:

```
{  
  "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch für die Developer Tools-Konsole unter [Verbindung erstellen](#).

- API-Einheiten finden Sie [CreateConnection](#) in der AWS CLI Befehlsreferenz.

create-host

Das folgende Codebeispiel zeigt die Verwendung `create-host`.

AWS CLI

Um einen Host zu erstellen

Das folgende `create-host` Beispiel zeigt, wie Sie einen Host erstellen, der den Endpunkt für die Infrastruktur darstellt, in der Ihr Drittanbieter installiert ist. In diesem Beispiel wird ein Host erstellt, auf dem GitHub Enterprise Server als Drittanbieter installiert ist.

Ein über den AWS CLI erstellter Host hat standardmäßig den Status `Ausstehend`. Nachdem Sie einen Host mit dem `create-host` CLI erstellt haben, verwenden Sie die Konsole oder die `update-host` CLI, um den Host so einzurichten, dass er den Status `Verfügbar` erhält.

```
aws codestar-connections create-host \  
  --name MyHost \  
  --provider-type GitHubEnterpriseServer \  
  --provider-endpoint "https://my-instance.dev"
```

Ausgabe:

```
{  
  "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-  
Host-28aef605"  
}
```

Weitere Informationen finden Sie unter [Create a host \(CLI\)](#) im Developer Tools-Konsolen-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateHost](#) unter AWS CLI Befehlsreferenz.

delete-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-connection`.

AWS CLI

Um eine Verbindung zu löschen

Das folgende `delete-connection` Beispiel zeigt, wie eine Verbindung gelöscht wird.

```
aws codestar-connections delete-connection \  
  --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Verbindung \(CLI\)](#) im Benutzerhandbuch für die Developer Tools-Konsole.

- API-Einheiten finden Sie [DeleteConnection](#) unter AWS CLI Befehlsreferenz.

`delete-host`

Das folgende Codebeispiel zeigt die Verwendung `delete-host`.

AWS CLI

Um einen Host zu löschen

Das folgende `delete-host` Beispiel zeigt, wie ein Host gelöscht wird. Bevor Sie einen Host löschen können, müssen Sie alle Verbindungen löschen, die mit dem Host verknüpft sind.

```
aws codestar-connections delete-host \  
  --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-  
Host-28aef605"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Hosts \(CLI\)](#) im Benutzerhandbuch für die Developer Tools-Konsole.

- API-Einheiten finden Sie [DeleteHost](#) unter AWS CLI Befehlsreferenz.

`get-connection`

Das folgende Codebeispiel zeigt die Verwendung `get-connection`.

AWS CLI

Um Informationen über eine Verbindung zu erhalten

Das folgende `get-connection` Beispiel zeigt Details zu einer Verbindung.

```
aws codestar-connections get-connection \  
  --connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Ausgabe:

```
{  
  "Connection": {  
    "ConnectionName": "MyConnection",  
    "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",  
    "ProviderType": "Bitbucket",  
    "OwnerAccountId": "123456789012",  
    "ConnectionStatus": "AVAILABLE"  
  }  
}
```

Weitere Informationen finden Sie im Benutzerhandbuch für die Developer Tools Console unter [Verbindungsdetails anzeigen](#).

- API-Einheiten finden Sie [GetConnection](#) in der AWS CLI Befehlsreferenz.

get-host

Das folgende Codebeispiel zeigt die Verwendung `get-host`.

AWS CLI

Um Informationen über einen Host zu erhalten

Das folgende `get-host` Beispiel zeigt Details zu einem Host:

```
aws codestar-connections get-host \  
  --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/  
MyHost-28aef605
```

Ausgabe:

```
{
  "Name": "MyHost",
  "Status": "AVAILABLE",
  "ProviderType": "GitHubEnterpriseServer",
  "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

Weitere Informationen finden Sie unter [Host-Details anzeigen \(CLI\)](#) im Developer Tools-Konsolen-Benutzerhandbuch.

- API-Einheiten finden Sie [GetHost](#) unter AWS CLI Befehlsreferenz.

list-connections

Das folgende Codebeispiel zeigt die Verwendung `list-connections`.

AWS CLI

Um Verbindungen aufzulisten

Im folgenden `list-connections` Beispiel wird eine Liste aller Verbindungen in deinem Konto für den Bitbucket-Anbietertyp abgerufen. :

```
aws codestar-connections list-connections \
--provider-type Bitbucket \
--max-results 5 \
--next-token: next-token
```

Ausgabe:

```
{
  "Connections": [
    {
      "ConnectionName": "my-connection",
      "ProviderType": "Bitbucket",
      "Status": "PENDING",
      "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
      "OwnerAccountId": "123456789012"
    }
  ]
}
```

```
    },
    {
      "ConnectionName": "my-other-connection",
      "ProviderType": "Bitbucket",
      "Status": "AVAILABLE",
      "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
      "OwnerAccountId": "123456789012"
    },
  ],
  "NextToken": "next-token"
}
```

Weitere Informationen finden Sie unter [Verbindungen auflisten \(CLI\)](#) im Benutzerhandbuch für die Developer Tools Console.

- API-Einheiten finden Sie [ListConnections](#) in der AWS CLI Befehlsreferenz.

list-hosts

Das folgende Codebeispiel zeigt die Verwendung `list-hosts`.

AWS CLI

Um Hosts aufzulisten

Im folgenden `list-hosts` Beispiel wird eine Liste aller Hosts in Ihrem Konto abgerufen.

```
aws codestar-connections list-hosts
```

Ausgabe:

```
{
  "Hosts": [
    {
      "Name": "My-Host",
      "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605",
      "ProviderType": "GitHubEnterpriseServer",
      "ProviderEndpoint": "https://my-instance.test.dev",
      "Status": "AVAILABLE"
    }
  ]
}
```



```
]
}
```

Weitere Informationen finden Sie unter [Hosts auflisten \(CLI\)](#) im Benutzerhandbuch für die Developer Tools Console.

- API-Einheiten finden Sie [ListHosts](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags aufzulisten

Im folgenden `list-tags-for-resource` Beispiel wird eine Liste aller Tags abgerufen, die an die angegebene Verbindungsressource angehängt sind.

```
aws codestar-connections list-tags-for-resource \
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "Project",
      "Value": "ProjectA"
    },
    {
      "Key": "ReadOnly",
      "Value": "true"
    }
  ]
}
```

Weitere Informationen finden Sie im Benutzerhandbuch für die Developer Tools-Konsole [unter Tags für eine Verbindungsressource anzeigen](#).

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Das folgende `tag-resource` Beispiel verknüpft einen Satz bereitgestellter Tags mit einer Verbindung. Verwenden Sie diesen Befehl, um Tags hinzuzufügen oder zu bearbeiten.

```
aws codestar-connections tag-resource \  
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \  
  --tags Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie im Benutzerhandbuch für die Developer Tools-Konsole unter Hinzufügen von Tags zu einer Verbindungsressource](#).

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um AWS Tags aus einer Verbindungsressource zu entfernen

Im Folgenden `untag-resource` wird ein Tag aus der angegebenen Ressource entfernt.

```
aws codestar-connections untag-resource \  
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \  
  --tag-keys Project ReadOnly
```

Ausgabe:

```
{
```

```
"Tags": []  
}
```

Weitere Informationen finden [Sie unter Entfernen von Tags aus einer Verbindungsressource](#) im Benutzerhandbuch für die Developer Tools-Konsole.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

Beispiele für Amazon Cognito Identity mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Cognito Identity Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-identity-pool

Das folgende Codebeispiel zeigt die Verwendung `create-identity-pool`.

AWS CLI

So erstellen Sie einen Identitätspool mit dem Cognito-Identitätspool-Anbieter

In diesem Beispiel wird ein Identitätspool mit dem Namen `MyIdentityPool` erstellt. Der Pool hat einen Cognito-Identitätspool-Anbieter. Nicht authentifizierte Identitäten sind nicht zulässig.

Befehl:

```
aws cognito-identity create-identity-pool --identity-pool-  
name MyIdentityPool --no-allow-unauthenticated-identities --cognito-
```

```
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Ausgabe:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
    {
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_11111111",
      "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- API-Einheiten finden Sie [CreateIdentityPool](#) in der AWS CLI Befehlsreferenz.

delete-identities

Das folgende Codebeispiel zeigt die Verwendung `delete-identities`.

AWS CLI

Löschen eines Identitätspools

In diesem Beispiel wird ein Identitätspool gelöscht.

Befehl:

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-west-2:11111111-1111-1111-1111-111111111111"
```

Ausgabe:

```
{
  "UnprocessedIdentityIds": []
}
```

- API-Einheiten finden Sie [DeleteIdentities](#) in der AWS CLI Befehlsreferenz.

delete-identity-pool

Das folgende Codebeispiel zeigt die Verwendung `delete-identity-pool`.

AWS CLI

Löschen eines Identitätspools

Im folgenden `delete-identity-pool`-Beispiel wird der angegebene Identitätspool gelöscht.

Befehl:

```
aws cognito-identity delete-identity-pool \  
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteIdentityPool](#) in der AWS CLI Befehlsreferenz.

describe-identity-pool

Das folgende Codebeispiel zeigt die Verwendung `describe-identity-pool`.

AWS CLI

Um einen Identitätspool zu beschreiben

Dieses Beispiel beschreibt einen Identitätspool.

Befehl:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-  
west-2:11111111-1111-1111-1111-111111111111"
```

Ausgabe:

```
{  
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
```

```
"IdentityPoolName": "MyIdentityPool",
"AllowUnauthenticatedIdentities": false,
"CognitoIdentityProviders": [
  {
    "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_1111111111",
    "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
    "ServerSideTokenCheck": false
  }
]
}
```

- API-Einheiten finden Sie [Beschreibung von IdentityPool](#) in der AWS CLI Befehlsreferenz.

get-identity-pool-roles

Das folgende Codebeispiel zeigt die Verwendung von `get-identity-pool-roles`.

AWS CLI

Um Identitätspool-Rollen abzurufen

In diesem Beispiel werden Identitätspool-Rollen abgerufen.

Befehl:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-  
west-2:11111111-1111-1111-1111-111111111111"
```

Ausgabe:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
    "authenticated": "arn:aws:iam::111111111111:role/  
Cognito_MyIdentityPoolAuth_Role",
    "unauthenticated": "arn:aws:iam::111111111111:role/  
Cognito_MyIdentityPoolUnauth_Role"
  }
}
```

- API-Einheiten finden Sie [GetIdentityPoolRoles](#) in der AWS CLI Befehlsreferenz.

list-identity-pools

Das folgende Codebeispiel zeigt die Verwendung `list-identity-pools`.

AWS CLI

Auflisten von Identitätspools

In diesem Beispiel werden Identitätspools aufgeführt. Es werden maximal 20 Identitäten aufgeführt.

Befehl:

```
aws cognito-identity list-identity-pools --max-results 20
```

Ausgabe:

```
{
  "IdentityPools": [
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "MyIdentityPool"
    },
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "AnotherIdentityPool"
    },
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "IdentityPoolRegionA"
    }
  ]
}
```

- API-Einheiten finden Sie [ListIdentityPools](#) in der AWS CLI Befehlsreferenz.

set-identity-pool-roles

Das folgende Codebeispiel zeigt die Verwendung `set-identity-pool-roles`.

AWS CLI

So legen Sie Identitätspool-Rollen fest

Im folgenden `set-identity-pool-roles` Beispiel wird eine Identitätspool-Rolle festgelegt.

```
aws cognito-identity set-identity-pool-roles \  
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \  
  --roles authenticated="arn:aws:iam::111111111111:role/  
Cognito_MyIdentityPoolAuth_Role"
```

- API-Einheiten finden Sie [SetIdentityPoolRoles](#) in der AWS CLI Befehlsreferenz.

update-identity-pool

Das folgende Codebeispiel zeigt die Verwendung `update-identity-pool`.

AWS CLI

Um einen Identitätspool zu aktualisieren

In diesem Beispiel wird ein Identitätspool aktualisiert. Es setzt den Namen auf `MyIdentityPool`. Es fügt Cognito als Identitätsanbieter hinzu. Es verbietet nicht authentifizierte Identitäten.

Befehl:

```
aws cognito-identity update-identity-pool --identity-pool-id "us-  
west-2:11111111-1111-1111-1111-111111111111" --identity-pool-  
name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-  
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-  
west-2_11111111",ClientId="3n4b5urk1ft4f13mg5e62d9ado",ServerSideTokenCheck=false
```

Ausgabe:

```
{  
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",  
  "IdentityPoolName": "MyIdentityPool",  
  "AllowUnauthenticatedIdentities": false,  
  "CognitoIdentityProviders": [  
    {  
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_11111111",  
      "ClientId": "3n4b5urk1ft4f13mg5e62d9ado",  
      "ServerSideTokenCheck": false  
    }  
  ]  
}
```



```
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [UpdateIdentityPool](#).AWS CLI

Beispiele für Amazon Cognito Identity Provider mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Cognito Identity Provider Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-custom-attributes

Das folgende Codebeispiel zeigt die Verwendung `add-custom-attributes`.

AWS CLI

Um ein benutzerdefiniertes Attribut hinzuzufügen

In diesem Beispiel wird einem Benutzerpool das benutzerdefinierte Attribut `CustomAttr 1` hinzugefügt. Es ist vom Typ Zeichenfolge und erfordert mindestens 1 und maximal 15 Zeichen. Sie ist nicht erforderlich.

Befehl:

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaaaa --custom-attributes
  Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- APIEinzelheiten finden Sie [AddCustomAttributes](#) in der AWS CLI Befehlsreferenz.

admim-disable-user

Das folgende Codebeispiel zeigt die Verwendung `admim-disable-user`.

AWS CLI

Um einen Benutzer zu deaktivieren

In diesem Beispiel wird der Benutzer `jane@example.com` deaktiviert.

Befehl:

```
aws cognito-idp admin-disable-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- APIEinzelheiten finden Sie [AdminDisableUser](#) in der AWS CLI Befehlsreferenz.

admim-enable-user

Das folgende Codebeispiel zeigt die Verwendung `admim-enable-user`.

AWS CLI

Um einen Benutzer zu aktivieren

In diesem Beispiel wird der Benutzername `jane@example.com` aktiviert.

Befehl:

```
aws cognito-idp admin-enable-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- APIEinzelheiten finden Sie [AdminEnableUser](#) in der AWS CLI Befehlsreferenz.

admin-add-user-to-group

Das folgende Codebeispiel zeigt die Verwendung `admin-add-user-to-group`.

AWS CLI

Um einen Benutzer zu einer Gruppe hinzuzufügen

In diesem Beispiel wird der Benutzer Jane zur Gruppe hinzugefügt MyGroup.

Befehl:

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --  
username Jane --group-name MyGroup
```

- API-Einheiten finden Sie [AdminAddUserToGroup](#) in der AWS CLI Befehlsreferenz.

admin-confirm-sign-up

Das folgende Codebeispiel zeigt die Verwendung `admin-confirm-sign-up`.

AWS CLI

Um die Benutzerregistrierung zu bestätigen

In diesem Beispiel wird der Benutzer `jane@example.com` bestätigt.

Befehl:

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- API-Einheiten finden Sie [AdminConfirmSignUp](#) in der AWS CLI Befehlsreferenz.

admin-create-user

Das folgende Codebeispiel zeigt die Verwendung `admin-create-user`.

AWS CLI

Um einen Benutzer zu erstellen

Im folgenden `admin-create-user` Beispiel wird ein Benutzer mit den angegebenen Einstellungen E-Mail-Adresse und Telefonnummer erstellt.

```
aws cognito-idp admin-create-user \  
  --user-pool-id us-west-2_aaaaaaaaa \  
  --username diego \  
  --user-attributes Name=email,Value=diego@example.com  
  Name=phone_number,Value="+15555551212" \  
  --message-action SUPPRESS
```

Ausgabe:

```
{  
  "User": {  
    "Username": "diego",  
    "Attributes": [  
      {  
        "Name": "sub",  
        "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"  
      },  
      {  
        "Name": "phone_number",  
        "Value": "+15555551212"  
      },  
      {  
        "Name": "email",  
        "Value": "diego@example.com"  
      }  
    ],  
    "UserCreateDate": 1548099495.428,  
    "UserLastModifiedDate": 1548099495.428,  
    "Enabled": true,  
    "UserStatus": "FORCE_CHANGE_PASSWORD"  
  }  
}
```

- API-Einheiten finden Sie [AdminCreateUser](#) unter AWS CLI Befehlsreferenz.

admin-delete-user-attributes

Das folgende Codebeispiel zeigt die Verwendung `admin-delete-user-attributes`.

AWS CLI

Um ein Benutzerattribut zu löschen

In diesem Beispiel wird das benutzerdefinierte Attribut CustomAttr 1 für den Benutzer diego@example.com gelöscht.

Befehl:

```
aws cognito-idp admin-delete-user-attributes --user-pool-id us-west-2_aaaaaaaa --  
username diego@example.com --user-attribute-names "custom:CustomAttr1"
```

- API-Einheiten finden Sie [AdminDeleteUserAttributes](#) in der AWS CLI Befehlsreferenz.

admin-delete-user

Das folgende Codebeispiel zeigt die Verwendung `admin-delete-user`.

AWS CLI

Benutzer löschen

In diesem Beispiel wird ein Benutzer gelöscht.

Befehl:

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaa --  
username diego@example.com
```

- API-Einheiten finden Sie [AdminDeleteUser](#) in der AWS CLI Befehlsreferenz.

admin-forget-device

Das folgende Codebeispiel zeigt die Verwendung `admin-forget-device`.

AWS CLI

Um ein Gerät zu vergessen

In diesem Beispiel wird das Gerät für den Benutzernamen jane@example.com vergessen

Befehl:

```
aws cognito-idp admin-forget-device --user-pool-id us-west-2_aaaaaaaa --  
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- APIEinzelheiten finden Sie [AdminForgetDevice](#) in der AWS CLI Befehlsreferenz.

admin-get-device

Das folgende Codebeispiel zeigt die Verwendung `admin-get-device`.

AWS CLI

Um ein Gerät zu bekommen

In diesem Beispiel wird ein Gerät mit dem Benutzernamen `jane@example.com` abgerufen

Befehl:

```
aws cognito-idp admin-get-device --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- APIEinzelheiten finden Sie [AdminGetDevice](#) in der AWS CLI Befehlsreferenz.

admin-get-user

Das folgende Codebeispiel zeigt die Verwendung `admin-get-user`.

AWS CLI

Benutzer abrufen

In diesem Beispiel werden Informationen zum Benutzernamen `jane@example.com` abgerufen.

Befehl:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

Ausgabe:

```
{  
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",  
  "Enabled": true,
```

```
"UserStatus": "FORCE_CHANGE_PASSWORD",
"UserCreateDate": 1548108509.537,
"UserAttributes": [
  {
    "Name": "sub",
    "Value": "4320de44-2322-4620-999b-5e2e1c8df013"
  },
  {
    "Name": "email_verified",
    "Value": "true"
  },
  {
    "Name": "phone_number_verified",
    "Value": "true"
  },
  {
    "Name": "phone_number",
    "Value": "+01115551212"
  },
  {
    "Name": "email",
    "Value": "jane@example.com"
  }
],
"UserLastModifiedDate": 1548108509.537
}
```

- API-Einheiten finden Sie [AdminGetUser](#) in der AWS CLI Befehlsreferenz.

admin-initiate-auth

Das folgende Codebeispiel zeigt die Verwendung `admin-initiate-auth`.

AWS CLI

Authentifizierung initiieren

In diesem Beispiel wird die Autorisierung mithilfe des AUTH Flows `ADMIN_NO_SRP_` für den Benutzernamen `jane@example.com` initiiert.

Auf dem Client muss die Anmeldung API für die serverbasierte Authentifizierung (`ADMIN_NO_`) aktiviert sein. SRP AUTH

Verwenden Sie die Sitzungsinformationen im Rückgabewert, um `-challenge` aufzurufen. `admin-respond-to-auth`

Befehl:

```
aws cognito-idp admin-initiate-auth --user-pool-id us-west-2_aaaaaaaaa --  
client-id 3n4b5urk1ft4fL3mg5e62d9ado --auth-flow ADMIN_NO_SRP_AUTH --auth-  
parameters USERNAME=jane@example.com,PASSWORD=password
```

Ausgabe:

```
{  
  "ChallengeName": "NEW_PASSWORD_REQUIRED",  
  "Session": "SESSION",  
  "ChallengeParameters": {  
    "USER_ID_FOR_SRP": "84514837-dcbc-4af1-abff-f3c109334894",  
    "requiredAttributes": "[]",  
    "userAttributes": "{\"email_verified\": \"true\", \"phone_number_verified\":  
\"true\", \"phone_number\": \"+01xxx5550100\", \"email\": \"jane@example.com\"}"  
  }  
}
```

- API-Einheiten finden Sie [AdminInitiateAuth](#) in der AWS CLI Befehlsreferenz.

admin-list-devices

Das folgende Codebeispiel zeigt die Verwendung `admin-list-devices`.

AWS CLI

Um Geräte für einen Benutzer aufzulisten

In diesem Beispiel werden Geräte für den Benutzernamen `jane@example.com` aufgelistet.

Befehl:

```
aws cognito-idp admin-list-devices --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- API-Einheiten finden Sie [AdminListDevices](#) in der AWS CLI Befehlsreferenz.

admin-list-groups-for-user

Das folgende Codebeispiel zeigt die Verwendung `admin-list-groups-for-user`.

AWS CLI

Um Gruppen für einen Benutzer aufzulisten

In diesem Beispiel werden Gruppen für den Benutzernamen `jane@example.com` aufgeführt.

Befehl:

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaa --  
username diego@example.com
```

Ausgabe:

```
{  
  "Groups": [  
    {  
      "Description": "Sample group",  
      "Precedence": 1,  
      "LastModifiedDate": 1548097827.125,  
      "RoleArn": "arn:aws:iam::111111111111:role/SampleRole",  
      "GroupName": "SampleGroup",  
      "UserPoolId": "us-west-2_aaaaaaaa",  
      "CreationDate": 1548097827.125  
    }  
  ]  
}
```

- API-Einheiten finden Sie [AdminListGroupForUser](#) in der AWS CLI Befehlsreferenz.

admin-list-user-auth-events

Das folgende Codebeispiel zeigt die Verwendung `admin-list-user-auth-events`.

AWS CLI

Um Autorisierungsereignisse für einen Benutzer aufzulisten

In diesem Beispiel werden Autorisierungsereignisse für den Benutzernamen `diego@example.com` aufgeführt.

Befehl:

```
aws cognito-idp admin-list-user-auth-events --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- API-Einheiten finden Sie [AdminListUserAuthEvents](#) in der AWS CLI Befehlsreferenz.

admin-remove-user-from-group

Das folgende Codebeispiel zeigt die Verwendung `admin-remove-user-from-group`.

AWS CLI

Um einen Benutzer aus einer Gruppe zu entfernen

In diesem Beispiel wird `jane@example.com` von `SampleGroup` entfernt.

Befehl:

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com --group-name SampleGroup
```

- API-Einheiten finden Sie [AdminRemoveUserFromGroup](#) in der AWS CLI Befehlsreferenz.

admin-reset-user-password

Das folgende Codebeispiel zeigt die Verwendung `admin-reset-user-password`.

AWS CLI

Um ein Benutzerkennwort zurückzusetzen

In diesem Beispiel wird das Passwort für `diego@example.com` zurückgesetzt.

Befehl:

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- API-Einheiten finden Sie [AdminResetUserPassword](#) in der AWS CLI Befehlsreferenz.

admin-set-user-mfa-preference

Das folgende Codebeispiel zeigt die Verwendung `admin-set-user-mfa-preference`.

AWS CLI

Um die MFA Benutzerpräferenz festzulegen

In diesem Beispiel wird die SMS MFA Präferenz für den Benutzernamen `diego@example.com` festgelegt.

Befehl:

```
aws cognito-idp admin-set-user-mfa-preference --user-pool-id us-west-2_aaaaaaaaa --username diego@example.com --sms-mfa-settings Enabled=false,PreferredMfa=false
```

- API-Einheiten finden Sie [AdminSetUserMfaPreference](#) in der AWS CLI Befehlsreferenz.

admin-set-user-settings

Das folgende Codebeispiel zeigt die Verwendung `admin-set-user-settings`.

AWS CLI

Um Benutzereinstellungen festzulegen

In diesem Beispiel wird die MFA Versandpräferenz für den Benutzernamen `diego@example.com` auf `EMAIL` festgelegt.

Befehl:

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- API-Einheiten finden Sie [AdminSetUserSettings](#) in der AWS CLI Befehlsreferenz.

admin-update-auth-event-feedback

Das folgende Codebeispiel zeigt die Verwendung `admin-update-auth-event-feedback`.

AWS CLI

Um Feedback zu einem Autorisierungsereignis zu geben

In diesem Beispiel wird der Feedback-Wert für ein durch event-id identifiziertes Autorisierungsereignis auf Valid gesetzt.

Befehl:

```
aws cognito-idp admin-update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa
--username diego@example.com --event-id c2c2cf89-c0d3-482d-aba6-99d78a5b0bfe --
feedback-value Valid
```

- API-Einheiten finden Sie unter [AdminUpdateAuthEventFeedback AWS CLI](#) Befehlsreferenz.

admin-update-device-status

Das folgende Codebeispiel zeigt die Verwendung `admin-update-device-status`.

AWS CLI

Um den Gerätestatus zu aktualisieren

In diesem Beispiel wird der Status „Gerät gespeichert“ für das mit dem Geräteschlüssel identifizierte Gerät auf „not_remembered“ gesetzt.

Befehl:

```
aws cognito-idp admin-update-device-status --user-pool-id us-west-2_aaaaaaaaa
--username diego@example.com --device-key xxxx --device-remembered-
status not_remembered
```

- API-Einheiten finden Sie unter [AdminUpdateDeviceStatus](#) Befehlsreferenz. AWS CLI

admin-update-user-attributes

Das folgende Codebeispiel zeigt die Verwendung `admin-update-user-attributes`.

AWS CLI

Um Benutzerattribute zu aktualisieren

In diesem Beispiel wird ein benutzerdefiniertes Benutzerattribut CustomAttr 1 für den Benutzer `diego@example.com` aktualisiert.

Befehl:

```
aws cognito-idp admin-update-user-attributes --user-pool-id us-west-2_aaaaaaaaa --username diego@example.com --user-attributes Name="custom:CustomAttr1",Value="Purple"
```

- API-Einheiten finden Sie [AdminUpdateUserAttributes](#) in der AWS CLI Befehlsreferenz.

change-password

Das folgende Codebeispiel zeigt die Verwendung `change-password`.

AWS CLI

Um ein Passwort zu ändern

In diesem Beispiel wird ein Passwort geändert.

Befehl:

```
aws cognito-idp change-password --previous-password OldPassword --proposed-password NewPassword --access-token ACCESS_TOKEN
```

- API-Einheiten finden Sie [ChangePassword](#) in der AWS CLI Befehlsreferenz.

confirm-forgot-password

Das folgende Codebeispiel zeigt die Verwendung `confirm-forgot-password`.

AWS CLI

Um ein vergessenes Passwort zu bestätigen

Dieses Beispiel bestätigt ein vergessenes Passwort für den Benutzernamen `diego@example.com`.

Befehl:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --  
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- API-Einheiten finden Sie [ConfirmForgotPassword](#) in der AWS CLI Befehlsreferenz.

confirm-sign-up

Das folgende Codebeispiel zeigt die Verwendung `confirm-sign-up`.

AWS CLI

Registrierung bestätigen

In diesem Beispiel wird die Registrierung des Benutzernamens `diego@example.com` bestätigt.

Befehl:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --  
username=diego@example.com --confirmation-code CONF_CODE
```

- API-Einheiten finden Sie [ConfirmSignUp](#) in der AWS CLI Befehlsreferenz.

create-group

Das folgende Codebeispiel zeigt die Verwendung `create-group`.

AWS CLI

Um eine Gruppe zu erstellen

In diesem Beispiel wird eine Gruppe mit einer Beschreibung erstellt.

Befehl:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-  
name MyNewGroup --description "New group."
```

Ausgabe:

```
{  
  "Group": {
```

```
"GroupName": "MyNewGroup",
"UserPoolId": "us-west-2_aaaaaaaaa",
"Description": "New group.",
"LastModifiedDate": 1548270073.795,
"CreationDate": 1548270073.795
}
}
```

Um eine Gruppe mit einer Rolle und Priorität zu erstellen

In diesem Beispiel wird eine Gruppe mit einer Beschreibung erstellt. Es beinhaltet auch eine Rolle und eine Rangfolge.

Befehl:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroupWithRole --description "New group with a role." --role-
arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

Ausgabe:

```
{
  "Group": {
    "GroupName": "MyNewGroupWithRole",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "New group with a role.",
    "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
    "Precedence": 2,
    "LastModifiedDate": 1548270211.761,
    "CreationDate": 1548270211.761
  }
}
```

- API-Einheiten finden Sie [CreateGroup](#) in der AWS CLI Befehlsreferenz.

create-user-import-job

Das folgende Codebeispiel zeigt die Verwendung `create-user-import-job`.

AWS CLI

Um einen Benutzerimportauftrag zu erstellen

In diesem Beispiel wird ein Benutzerimportauftrag mit dem Namen erstellt `MyImportJob`.

Weitere Informationen zum Importieren von Benutzern finden Sie unter `Benutzer` aus einer CSV Datei in Benutzerpools importieren.

Befehl:

```
aws cognito-idp create-user-import-job --user-pool-id us-west-2_aaaaaaaaa --  
job-name MyImportJob --cloud-watch-logs-role-arn arn:aws:iam::111111111111:role/  
CognitoCloudWatchLogsRole
```

Ausgabe:

```
{  
  "UserImportJob": {  
    "JobName": "MyImportJob",  
    "JobId": "import-qQ0DCt2fRh",  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "PreSignedUrl": "PRE_SIGNED_URL",  
    "CreationDate": 1548271795.471,  
    "Status": "Created",  
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/  
CognitoCloudWatchLogsRole",  
    "ImportedUsers": 0,  
    "SkippedUsers": 0,  
    "FailedUsers": 0  
  }  
}
```

Laden Sie die CSV-Datei mit `curl` hoch und verwenden Sie dabei das vorsignierte: URL

Befehl:

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"  
"PRE_SIGNED_URL"
```

- API-Einheiten finden Sie [CreateUserImportJob](#) in AWS CLI der Befehlsreferenz.

create-user-pool-client

Das folgende Codebeispiel zeigt die Verwendung `create-user-pool-client`.

AWS CLI

Um einen Benutzerpool-Client zu erstellen

In diesem Beispiel wird ein neuer Benutzerpool-Client mit zwei expliziten Autorisierungsabläufen erstellt: `USER_PASSWORD_AUTH` und `ADMIN_NO_SRP_AUTH`.

Befehl:

```
aws cognito-idp create-user-pool-client --user-pool-id us-west-2_aaaaaaaaa
--client-name MyNewClient --no-generate-secret --explicit-auth-
flows "USER_PASSWORD_AUTH" "ADMIN_NO_SRP_AUTH"
```

Ausgabe:

```
{
  "UserPoolClient": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientName": "MyNewClient",
    "ClientId": "6p3bs000no6a4ue1idruvd05ad",
    "LastModifiedDate": 1548697449.497,
    "CreationDate": 1548697449.497,
    "RefreshTokenValidity": 30,
    "ExplicitAuthFlows": [
      "USER_PASSWORD_AUTH",
      "ADMIN_NO_SRP_AUTH"
    ],
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

- API-Einheiten finden Sie [CreateUserPoolClient](#) in der AWS CLI Befehlsreferenz.

create-user-pool-domain

Das folgende Codebeispiel zeigt die Verwendung `create-user-pool-domain`.

AWS CLI

Um eine Benutzerpool-Domäne zu erstellen

In diesem Beispiel wird eine neue Benutzerpool-Domäne mit zwei expliziten Autorisierungsabläufen erstellt: `USER_PASSWORD_AUTH` und `ADMIN_SRP_NO_SRP_AUTH`.

Befehl:

```
aws cognito-idp create-user-pool-domain --user-pool-id us-west-2_aaaaaaaaa --  
domain my-new-domain
```

- API-Einheiten finden Sie [CreateUserPoolDomain](#) in AWS CLI der Befehlsreferenz.

create-user-pool

Das folgende Codebeispiel zeigt die Verwendung `create-user-pool`.

AWS CLI

So erstellen Sie einen minimal konfigurierten Benutzerpool

In diesem Beispiel wird ein Benutzerpool erstellt, der `MyUserPool` mit Standardwerten benannt wird. Es gibt keine erforderlichen Attribute und keine Anwendungs-Clients. MFA und erweiterte Sicherheit ist deaktiviert.

Befehl:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Ausgabe:

```
{  
  "UserPool": {  
    "SchemaAttributes": [  
      {  
        "Name": "sub",  
        "StringAttributeConstraints": {  
          "MinLength": "1",  
          "MaxLength": "2048"  
        },  
        "DeveloperOnlyAttribute": false,  
        "Required": true,  
        "AttributeDataType": "String",  
        "Mutable": false  
      },  
      {  
        "Name": "name",  
        "StringAttributeConstraints": {
```

```
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "given_name",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "family_name",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "middle_name",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "nickname",
    "StringAttributeConstraints": {
```

```
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "preferred_username",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "profile",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "picture",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "website",
    "StringAttributeConstraints": {
```

```
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "email",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
},
{
    "Name": "gender",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "birthdate",
    "StringAttributeConstraints": {
        "MinLength": "10",
        "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
```

```
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "zoneinfo",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "locale",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "phone_number",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "phone_number_verified",
    "Mutable": true
  },
}
```

```
{
  "Name": "address",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "updated_at",
  "NumberAttributeConstraints": {
    "MinValue": "0"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "Number",
  "Mutable": true
}
],
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"LastModifiedDate": 1547833345.777,
"AdminCreateUserConfig": {
  "UnusedAccountValidityDays": 7,
  "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {},
"Policies": {
  "PasswordPolicy": {
    "RequireLowercase": true,
    "RequireSymbols": true,
    "RequireNumbers": true,
    "MinimumLength": 8,
    "RequireUppercase": true
  }
},
"CreationDate": 1547833345.777,
"EstimatedNumberOfUsers": 0,
"Id": "us-west-2_aaaaaaaaa",
"LambdaConfig": {}
}
```

```
}
```

So erstellen Sie einen Benutzerpool mit zwei erforderlichen Attributen

In diesem Beispiel wird ein Benutzerpool erstellt `MyUserPool`. Der Pool ist so konfiguriert, dass er E-Mail-Adressen als Benutzernamensattribut akzeptiert. Außerdem wird die E-Mail-Quelladresse mit Amazon Simple Email Service auf eine validierte Adresse gesetzt.

Befehl:

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-east-1:111111111111:identity/jane@example.com",ReplyToEmailAddress="jane@example.com"
```

Ausgabe:

```
{
  "UserPool": {
    "SchemaAttributes": [
      {
        "Name": "sub",
        "StringAttributeConstraints": {
          "MinLength": "1",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": true,
        "AttributeDataType": "String",
        "Mutable": false
      },
      {
        "Name": "name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
        "Mutable": true
      }
    ]
  }
}
```



```
    "Name": "given_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "family_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "middle_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "nickname",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
```

```
    "Name": "preferred_username",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "profile",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "picture",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "website",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
```

```
    "Name": "email",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
  },
  {
    "Name": "gender",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "birthdate",
    "StringAttributeConstraints": {
      "MinLength": "10",
      "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "zoneinfo",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    }
  }
}
```

```
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "locale",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "phone_number",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "phone_number_verified",
    "Mutable": true
  },
  {
    "Name": "address",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
```

```
        "Mutable": true
    },
    {
        "Name": "updated_at",
        "NumberAttributeConstraints": {
            "MinValue": "0"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "Number",
        "Mutable": true
    }
],
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"LastModifiedDate": 1547837788.189,
"AdminCreateUserConfig": {
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {
    "ReplyToEmailAddress": "jane@example.com",
    "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
},
"Policies": {
    "PasswordPolicy": {
        "RequireLowercase": true,
        "RequireSymbols": true,
        "RequireNumbers": true,
        "MinimumLength": 8,
        "RequireUppercase": true
    }
},
"UsernameAttributes": [
    "email"
],
"CreationDate": 1547837788.189,
"EstimatedNumberOfUsers": 0,
"Id": "us-west-2_aaaaaaaaa",
"LambdaConfig": {}
}
}
```

- API-Einheiten finden Sie [CreateUserPool](#) in der AWS CLI Befehlsreferenz.

delete-group

Das folgende Codebeispiel zeigt die Verwendung `delete-group`.

AWS CLI

Um eine Gruppe zu löschen

In diesem Beispiel wird eine Gruppe gelöscht.

Befehl:

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaa --group-name MyGroupName
```

- API-Einheiten finden Sie [DeleteGroup](#) in der AWS CLI Befehlsreferenz.

delete-identity-provider

Das folgende Codebeispiel zeigt die Verwendung `delete-identity-provider`.

AWS CLI

Um einen Identitätsanbieter zu löschen

In diesem Beispiel wird ein Identitätsanbieter gelöscht.

Befehl:

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaa --provider-name Facebook
```

- API-Einheiten finden Sie unter [DeleteIdentityProvider AWS CLI](#) Befehlsreferenz.

delete-resource-server

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-server`.

AWS CLI

Um einen Ressourcenserver zu löschen

In diesem Beispiel wird ein Ressourcenserver mit dem Namen `weather.example.com` gelöscht.

Befehl:

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --  
identifier weather.example.com
```

- API-Einheiten finden Sie unter [DeleteResourceServer](#) Befehlsreferenz.AWS CLI

delete-user-attributes

Das folgende Codebeispiel zeigt die Verwendung `delete-user-attributes`.

AWS CLI

Um Benutzerattribute zu löschen

In diesem Beispiel wird das Benutzerattribut "FAVORITE_ANIMAL" gelöscht.

Befehl:

```
aws cognito-idp delete-user-attributes --access-token ACCESS_TOKEN --user-attribute-  
names "FAVORITE_ANIMAL"
```

- API-Einheiten finden Sie [DeleteUserAttributes](#) in der AWS CLI Befehlsreferenz.

delete-user-pool-client

Das folgende Codebeispiel zeigt die Verwendung `delete-user-pool-client`.

AWS CLI

Um einen Benutzerpool-Client zu löschen

In diesem Beispiel wird ein Benutzerpool-Client gelöscht.

Befehl:

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kpbsnet7mpld0
```

- API-Einheiten finden Sie [DeleteUserPoolClient](#) in der AWS CLI Befehlsreferenz.

delete-user-pool-domain

Das folgende Codebeispiel zeigt die Verwendung `delete-user-pool-domain`.

AWS CLI

Um eine Benutzerpool-Domäne zu löschen

Im folgenden `delete-user-pool-domain` Beispiel wird eine Benutzerpool-Domäne mit dem Namen `my-domain` gelöscht.

```
aws cognito-idp delete-user-pool-domain \
  --user-pool-id us-west-2_aaaaaaaaa \
  --domain my-domain
```

- API-Einheiten finden Sie unter [DeleteUserPoolDomain AWS CLI](#) Befehlsreferenz.

delete-user-pool

Das folgende Codebeispiel zeigt die Verwendung `delete-user-pool`.

AWS CLI

Um einen Benutzerpool zu löschen

In diesem Beispiel wird ein Benutzerpool mit der Benutzerpool-ID `us-west-2_aaaaaaaaa` gelöscht.

Befehl:

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- Einzelheiten API finden Sie in der Befehlsreferenz. [DeleteUserPool](#) AWS CLI

delete-user

Das folgende Codebeispiel zeigt die Verwendung `delete-user`.

AWS CLI

Benutzer löschen

In diesem Beispiel wird ein Benutzer gelöscht.

Befehl:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- API-Einheiten finden Sie [DeleteUser](#) in der AWS CLI Befehlsreferenz.

describe-identity-provider

Das folgende Codebeispiel zeigt die Verwendung `describe-identity-provider`.

AWS CLI

Um einen Identitätsanbieter zu beschreiben

Dieses Beispiel beschreibt einen Identitätsanbieter namens Facebook.

Befehl:

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --  
provider-name Facebook
```

Ausgabe:

```
{  
  "IdentityProvider": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "ProviderName": "Facebook",  
    "ProviderType": "Facebook",  
    "ProviderDetails": {  
      "attributes_url": "https://graph.facebook.com/me?fields=",  
      "attributes_url_add_attributes": "true",  
      "authorize_scopes": "myscope",  
      "authorize_url": "https://www.facebook.com/v2.9/dialog/oauth",  
      "client_id": "11111",  
      "client_secret": "11111",  
      "token_request_method": "GET",  
      "token_url": "https://graph.facebook.com/v2.9/oauth/access_token"    }  
  }  
}
```

```
    },
    "AttributeMapping": {
      "username": "id"
    },
    },
    "IdpIdentifiers": [],
    "LastModifiedDate": 1548105901.736,
    "CreationDate": 1548105901.736
  }
}
```

- API-Einheiten finden Sie [DescribeIdentityProvider](#) in der AWS CLI Befehlsreferenz.

describe-resource-server

Das folgende Codebeispiel zeigt die Verwendung `describe-resource-server`.

AWS CLI

Um einen Ressourcenserver zu beschreiben

Dieses Beispiel beschreibt den Ressourcenserver `weather.example.com`.

Befehl:

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaa --
  identifier weather.example.com
```

Ausgabe:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Weather",
    "Scopes": [
      {
        "ScopeName": "weather.update",
        "ScopeDescription": "Update weather forecast"
      },
      {
        "ScopeName": "weather.read",
        "ScopeDescription": "Read weather forecasts"
      }
    ]
  }
}
```

```
    {
      "ScopeName": "weather.delete",
      "ScopeDescription": "Delete a weather forecast"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeResourceServer](#) in AWS CLI der Befehlsreferenz.

describe-risk-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-risk-configuration`.

AWS CLI

Um eine Risikokonfiguration zu beschreiben

Dieses Beispiel beschreibt die Risikokonfiguration im Zusammenhang mit Pool `us-west-2_aaaaaaaaa`.

Befehl:

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaa
```

Ausgabe:

```
{
  "RiskConfiguration": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "CompromisedCredentialsRiskConfiguration": {
      "EventFilter": [
        "SIGN_IN",
        "SIGN_UP",
        "PASSWORD_CHANGE"
      ],
      "Actions": {
        "EventAction": "BLOCK"
      }
    },
    "AccountTakeoverRiskConfiguration": {
      "NotifyConfiguration": {
        "From": "diego@example.com",
```

```

    "ReplyTo": "diego@example.com",
    "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
diego@example.com",
    "BlockEmail": {
        "Subject": "Blocked sign-in attempt",
        "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
{country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
link</a> to let us know</pre>\n</body>\n</html>",
        "TextBody": "We blocked an unrecognized sign-in to your account
with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
{city}, {country}\nIf this sign-in was not by you, you should change your password
and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
you can follow {one-click-link-valid} to let us know"
    },
    "NoActionEmail": {
        "Subject": "New sign-in attempt",
        "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
{country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
link</a> to let us know</pre>\n</body>\n</html>",
        "TextBody": "We observed an unrecognized sign-in to your account
with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
{city}, {country}\nIf this sign-in was not by you, you should change your password
and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
you can follow {one-click-link-valid} to let us know"
    },
    "MfaEmail": {
        "Subject": "New sign-in attempt",
        "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
{country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
password and notify us by clicking on <a href={one-click-link-invalid}>this link</

```



```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaa --job-id import-TZqNQvDRnW
```

Ausgabe:

```
{
  "UserImportJob": {
    "JobName": "import-Test1",
    "JobId": "import-TZqNQvDRnW",
    "UserPoolId": "us-west-2_aaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED URL",
    "CreationDate": 1548271708.512,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

- API-Einheiten finden Sie [DescribeUserImportJob](#) unter AWS CLI Befehlsreferenz.

describe-user-pool-client

Das folgende Codebeispiel zeigt die Verwendung `describe-user-pool-client`.

AWS CLI

Um einen Benutzerpool-Client zu beschreiben

Dieses Beispiel beschreibt einen Benutzerpool-Client.

Befehl:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaa --client-id 38fjsnc484p94kpbsnet7mpld0
```

Ausgabe:

```
{
  "UserPoolClient": {
```

```
"UserPoolId": "us-west-2_aaaaaaaaa",
"ClientName": "MyApp",
"ClientId": "38fjsnc484p94kpbsnet7mpld0",
"ClientSecret": "CLIENT_SECRET",
"LastModifiedDate": 1548108676.163,
"CreationDate": 1548108676.163,
"RefreshTokenValidity": 30,
"ReadAttributes": [
  "address",
  "birthdate",
  "custom:CustomAttr1",
  "custom:CustomAttr2",
  "email",
  "email_verified",
  "family_name",
  "gender",
  "given_name",
  "locale",
  "middle_name",
  "name",
  "nickname",
  "phone_number",
  "phone_number_verified",
  "picture",
  "preferred_username",
  "profile",
  "updated_at",
  "website",
  "zoneinfo"
],
"WriteAttributes": [
  "address",
  "birthdate",
  "custom:CustomAttr1",
  "custom:CustomAttr2",
  "email",
  "family_name",
  "gender",
  "given_name",
  "locale",
  "middle_name",
  "name",
  "nickname",
  "phone_number",
```

```

        "picture",
        "preferred_username",
        "profile",
        "updated_at",
        "website",
        "zoneinfo"
    ],
    "ExplicitAuthFlows": [
        "ADMIN_NO_SRP_AUTH",
        "USER_PASSWORD_AUTH"
    ],
    "AllowedOauthFlowsUserPoolClient": false
}
}

```

- API-Einheiten finden Sie [DescribeUserPoolClient](#) in der AWS CLI Befehlsreferenz.

describe-user-pool-domain

Das folgende Codebeispiel zeigt die Verwendung `describe-user-pool-domain`.

AWS CLI

Um einen Benutzerpool-Client zu beschreiben

Dieses Beispiel beschreibt eine Benutzerpool-Domain mit dem Namen `my-domain`.

Befehl:

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

Ausgabe:

```

{
  "DomainDescription": {
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "AWSAccountId": "111111111111",
    "Domain": "my-domain",
    "S3Bucket": "aws-cognito-prod-pdx-assets",
    "CloudFrontDistribution": "aaaaaaaaaaaaa.cloudfront.net",
    "Version": "20190128175402",
    "Status": "ACTIVE",
    "CustomDomainConfig": {}
  }
}

```



```
}  
}
```

- API-Einheiten finden Sie [DescribeUserPoolDomain](#) in der AWS CLI Befehlsreferenz.

describe-user-pool

Das folgende Codebeispiel zeigt die Verwendung `describe-user-pool`.

AWS CLI

Um einen Benutzerpool zu beschreiben

Dieses Beispiel beschreibt einen Benutzerpool mit der Benutzerpool-ID `us-west-2_aaaaaaaaaa`.

Befehl:

```
aws cognito-idp describe-user-pool --user-pool-id us-west-2_aaaaaaaaaa
```

Ausgabe:

```
{  
  "UserPool": {  
    "SmsVerificationMessage": "Your verification code is {####}. ",  
    "SchemaAttributes": [  
      {  
        "Name": "sub",  
        "StringAttributeConstraints": {  
          "MinLength": "1",  
          "MaxLength": "2048"  
        },  
        "DeveloperOnlyAttribute": false,  
        "Required": true,  
        "AttributeDataType": "String",  
        "Mutable": false  
      },  
      {  
        "Name": "name",  
        "StringAttributeConstraints": {  
          "MinLength": "0",  
          "MaxLength": "2048"  
        },  
        "DeveloperOnlyAttribute": false,  
        "Required": true,  
        "AttributeDataType": "String",  
        "Mutable": false  
      }  
    ]  
  }  
}
```

```
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "given_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "family_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "middle_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "nickname",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
```

```
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "preferred_username",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "profile",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "picture",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "website",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
```

```
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "email",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": true,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
  },
  {
    "Name": "gender",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "birthdate",
    "StringAttributeConstraints": {
      "MinLength": "10",
      "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  },
```

```
{
  "Name": "zoneinfo",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "locale",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "phone_number",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "AttributeDataType": "Boolean",
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "Name": "phone_number_verified",
  "Mutable": true
},
{
  "Name": "address",
  "StringAttributeConstraints": {
    "MinLength": "0",
```

```

        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "updated_at",
    "NumberAttributeConstraints": {
        "MinValue": "0"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "Number",
    "Mutable": true
}
],
"EmailVerificationSubject": "Your verification code",
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"EmailVerificationMessage": "Your verification code is {#####}. ",
"SmsAuthenticationMessage": "Your authentication code is {#####}. ",
"LastModifiedDate": 1547763720.822,
"AdminCreateUserConfig": {
    "InviteMessageTemplate": {
        "EmailMessage": "Your username is {username} and temporary password is
{#####}. ",
        "EmailSubject": "Your temporary password",
        "SMSMessage": "Your username is {username} and temporary password is
{#####}. "
    },
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {
    "ReplyToEmailAddress": "myemail@mydomain.com"
    "SourceArn": "arn:aws:ses:us-east-1:000000000000:identity/
myemail@mydomain.com"
},
"AutoVerifiedAttributes": [
    "email"
],
"Policies": {

```

```
    "PasswordPolicy": {
      "RequireLowercase": true,
      "RequireSymbols": true,
      "RequireNumbers": true,
      "MinimumLength": 8,
      "RequireUppercase": true
    }
  },
  "UserPoolTags": {},
  "UsernameAttributes": [
    "email"
  ],
  "CreationDate": 1547763720.822,
  "EstimatedNumberOfUsers": 1,
  "Id": "us-west-2_aaaaaaaaaa",
  "LambdaConfig": {}
}
}
```

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeUserPool](#) AWS CLI

forget-device

Das folgende Codebeispiel zeigt die Verwendung `forget-device`.

AWS CLI

Um ein Gerät zu vergessen

In diesem Beispiel wird Gerät für Gerät vergessen.

Befehl:

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- API-Einheiten finden Sie [ForgetDevice](#) in der AWS CLI Befehlsreferenz.

forgot-password

Das folgende Codebeispiel zeigt die Verwendung `forgot-password`.

AWS CLI

Um eine Passwortänderung zu erzwingen

Im folgenden forgot-password Beispiel wird eine Nachricht an jane@example.com gesendet, um ihr Passwort zu ändern.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpbsnet7mpld0 --  
username jane@example.com
```

Ausgabe:

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

- API-Einheiten finden Sie [ForgotPassword](#) in der AWS CLI Befehlsreferenz.

get-csv-header

Das folgende Codebeispiel zeigt die Verwendung get-csv-header.

AWS CLI

Um einen CSV-Header zu erstellen

In diesem Beispiel wird ein CSV-Header erstellt.

Weitere Informationen zum Importieren von Benutzern finden Sie unter Benutzer aus einer CSV Datei in Benutzerpools importieren.

Befehl:

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaaa
```

Ausgabe:


```
{
  "UserPoolId": "us-west-2_aaaaaaaaa",
  "CSVHeader": [
    "name",
    "given_name",
    "family_name",
    "middle_name",
    "nickname",
    "preferred_username",
    "profile",
    "picture",
    "website",
    "email",
    "email_verified",
    "gender",
    "birthdate",
    "zoneinfo",
    "locale",
    "phone_number",
    "phone_number_verified",
    "address",
    "updated_at",
    "cognito:mfa_enabled",
    "cognito:username"
  ]
}
```

... Benutzer aus einer CSV Datei in Benutzerpools importieren: <https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-using-import-tool.html>

- API-Einheiten finden Sie [GetCsvHeader](#) in der AWS CLI Befehlsreferenz.

get-group

Das folgende Codebeispiel zeigt die Verwendung `get-group`.

AWS CLI

Um Informationen über eine Gruppe zu erhalten

In diesem Beispiel werden Informationen über eine Gruppe mit dem Namen `MyGroup` abgerufen.

Befehl:

```
aws cognito-idp get-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

Ausgabe:

```
{
  "Group": {
    "GroupName": "MyGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "A sample group.",
    "LastModifiedDate": 1548270073.795,
    "CreationDate": 1548270073.795
  }
}
```

- API-Einheiten finden Sie [GetGroup](#) unter AWS CLI Befehlsreferenz.

get-signing-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-signing-certificate`.

AWS CLI

Um ein Signaturzertifikat zu erhalten

In diesem Beispiel wird ein Signaturzertifikat für einen Benutzerpool abgerufen.

Befehl:

```
aws cognito-idp get-signing-certificate --user-pool-id us-west-2_aaaaaaaaa
```

Ausgabe:

```
{
  "Certificate": "CERTIFICATE_DATA"
}
```

- API-Einheiten finden Sie [GetSigningCertificate](#) in der AWS CLI Befehlsreferenz.

get-ui-customization

Das folgende Codebeispiel zeigt die Verwendung `get-ui-customization`.

AWS CLI

Um Informationen zur Anpassung der Benutzeroberfläche zu erhalten

In diesem Beispiel werden Informationen zur Anpassung der Benutzeroberfläche für einen Benutzerpool abgerufen.

Befehl:

```
aws cognito-idp get-ui-customization --user-pool-id us-west-2_aaaaaaaaa
```

Ausgabe:

```
{
  "UICustomization": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientId": "ALL",
    "ImageUrl": "https://aaaaaaaaaaaaa.cloudfront.net/us-west-2_aaaaaaaaa/
ALL/20190128231240/assets/images/image.jpg",
    "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color:
lightgray;\n}\n.label-customizable {\n\tfont-weight: 300;\n}\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
{\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n
\theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
#D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
#555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n.inputField-
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-
customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-
bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:
#46b8da;\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#31b0d5;\n}\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #faf;\n}\n",
    "CSSVersion": "20190128231240"
  }
}
```

```
}  
}
```

- API-Einheiten finden Sie [GetUiCustomization](#) unter AWS CLI Befehlsreferenz.

list-user-import-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-user-import-jobs`.

AWS CLI

Um Benutzerimportaufträge aufzulisten

In diesem Beispiel werden Benutzerimportaufträge aufgeführt.

Weitere Informationen zum Importieren von Benutzern finden Sie unter Benutzer aus einer CSV Datei in Benutzerpools importieren.

Befehl:

```
aws cognito-idp list-user-import-jobs --user-pool-id us-west-2_aaaaaaaaaa --max-  
results 20
```

Ausgabe:

```
{  
  "UserImportJobs": [  
    {  
      "JobName": "Test2",  
      "JobId": "import-d00nwGA3mV",  
      "UserPoolId": "us-west-2_aaaaaaaaaa",  
      "PreSignedUrl": "PRE_SIGNED_URL",  
      "CreationDate": 1548272793.069,  
      "Status": "Created",  
      "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/  
CognitoCloudWatchLogsRole",  
      "ImportedUsers": 0,  
      "SkippedUsers": 0,  
      "FailedUsers": 0  
    },  
    {
```

```

    "JobName": "Test1",
    "JobId": "import-qQ0DCt2fRh",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548271795.471,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  },
  {
    "JobName": "import-Test1",
    "JobId": "import-TZqNQvDRnW",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548271708.512,
    "StartDate": 1548277247.962,
    "CompletionDate": 1548277248.912,
    "Status": "Failed",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 1,
    "CompletionMessage": "Too many users have failed or been skipped during
the import."
  }
]
}

```

- API-Einheiten finden Sie [ListUserImportJobs](#) unter AWS CLI Befehlsreferenz.

list-user-pools

Das folgende Codebeispiel zeigt die Verwendung `list-user-pools`.

AWS CLI

Benutzerpools auflisten

In diesem Beispiel werden bis zu 20 Benutzerpools aufgelistet.

Befehl:

```
aws cognito-idp list-user-pools --max-results 20
```

Ausgabe:

```
{
  "UserPools": [
    {
      "CreationDate": 1547763720.822,
      "LastModifiedDate": 1547763720.822,
      "LambdaConfig": {},
      "Id": "us-west-2_aaaaaaaaa",
      "Name": "MyUserPool"
    }
  ]
}
```

- API-Einheiten finden Sie [ListUserPools](#) in der AWS CLI Befehlsreferenz.

list-users-in-group

Das folgende Codebeispiel zeigt die Verwendung `list-users-in-group`.

AWS CLI

Um Benutzer in einer Gruppe aufzulisten

In diesem Beispiel werden Benutzer in einer Gruppe aufgeführt `MyGroup`.

Befehl:

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

Ausgabe:

```
{
  "Users": [
    {
      "Username": "acf10624-80bb-401a-ac61-607bee2110ec",

```

```
    "Attributes": [
      {
        "Name": "sub",
        "Value": "acf10624-80bb-401a-ac61-607bee2110ec"
      },
      {
        "Name": "custom:CustomAttr1",
        "Value": "New Value!"
      },
      {
        "Name": "email",
        "Value": "jane@example.com"
      }
    ],
    "UserCreateDate": 1548102770.284,
    "UserLastModifiedDate": 1548103204.893,
    "Enabled": true,
    "UserStatus": "CONFIRMED"
  },
  {
    "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
    "Attributes": [
      {
        "Name": "sub",
        "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
      },
      {
        "Name": "email_verified",
        "Value": "true"
      },
      {
        "Name": "email",
        "Value": "diego@example.com"
      }
    ],
    "UserCreateDate": 1548089817.683,
    "UserLastModifiedDate": 1548089817.683,
    "Enabled": true,
    "UserStatus": "FORCE_CHANGE_PASSWORD"
  }
]
}
```

- API-Einheiten finden Sie [ListUsersInGroup](#) in der AWS CLI Befehlsreferenz.

list-users

Das folgende Codebeispiel zeigt die Verwendung `list-users`.

AWS CLI

Benutzer auflisten

In diesem Beispiel werden bis zu 20 Benutzer aufgelistet.

Befehl:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

Ausgabe:

```
{
  "Users": [
    {
      "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
      "Enabled": true,
      "UserStatus": "FORCE_CHANGE_PASSWORD",
      "UserCreateDate": 1548089817.683,
      "UserLastModifiedDate": 1548089817.683,
      "Attributes": [
        {
          "Name": "sub",
          "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
        },
        {
          "Name": "email_verified",
          "Value": "true"
        },
        {
          "Name": "email",
          "Value": "mary@example.com"
        }
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [ListUsers](#) in der AWS CLI Befehlsreferenz.

resend-confirmation-code

Das folgende Codebeispiel zeigt die Verwendung `resend-confirmation-code`.

AWS CLI

Bestätigungscode erneut senden

Im folgenden `resend-confirmation-code`-Beispiel wird ein Bestätigungscode an den Benutzer `jane` gesendet.

```
aws cognito-idp resend-confirmation-code \  
  --client-id 12a3b456c7de890f11g123hijk \  
  --username jane
```

Ausgabe:

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

Weitere Informationen finden Sie unter [Registrieren und Bestätigen von Benutzerkonten](#) im Amazon-Cognito-Entwicklerhandbuch.

- API-Einheiten finden Sie [ResendConfirmationCode](#) in der AWS CLI Befehlsreferenz.

respond-to-auth-challenge

Das folgende Codebeispiel zeigt die Verwendung `respond-to-auth-challenge`.

AWS CLI

Reaktion auf eine Amazon Cognito SRP-Authentifizierungs-Challenge

Dieses Beispiel veranschaulicht die Reaktion auf eine Authentifizierungs-Challenge, die mit „initiate-auth“ initiiert wurde. Es ist eine Antwort auf die `NEW_PASSWORD_REQUIRED`-Herausforderung. Es wird ein Passwort für den Benutzer `jane@example.com` festgelegt.

Befehl:

```
aws cognito-idp respond-to-auth-challenge --client-id 3n4b5urk1ft4f13mg5e62d9ado
--challenge-name NEW_PASSWORD_REQUIRED --challenge-responses
USERNAME=jane@example.com,NEW_PASSWORD="password" --session "SESSION_TOKEN"
```

Ausgabe:

```
{
  "ChallengeParameters": {},
  "AuthenticationResult": {
    "AccessToken": "ACCESS_TOKEN",
    "ExpiresIn": 3600,
    "TokenType": "Bearer",
    "RefreshToken": "REFRESH_TOKEN",
    "IdToken": "ID_TOKEN",
    "NewDeviceMetadata": {
      "DeviceKey": "us-west-2_fec070d2-fa88-424a-8ec8-b26d7198eb23",
      "DeviceGroupKey": "-wt2ha1Zd"
    }
  }
}
```

- API-Einheiten finden Sie [RespondToAuthChallenge](#) in der AWS CLI Befehlsreferenz.

set-risk-configuration

Das folgende Codebeispiel zeigt die Verwendung `set-risk-configuration`.

AWS CLI

Um die Risikokonfiguration festzulegen

In diesem Beispiel wird die Risikokonfiguration für einen Benutzerpool festgelegt. Es legt die Aktion für das Anmeldeereignis auf `ACTION_NO_` fest.

Befehl:

```
aws cognito-idp set-risk-configuration --user-pool-id us-west-2_aaaaaaaaa --
compromised-credentials-risk-
configuration EventFilter=SIGN_UP,Actions={EventAction=NO_ACTION}
```

Ausgabe:

```
{
  "RiskConfiguration": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "CompromisedCredentialsRiskConfiguration": {
      "EventFilter": [
        "SIGN_UP"
      ],
      "Actions": {
        "EventAction": "NO_ACTION"
      }
    }
  }
}
```

- API-Einheiten finden Sie [SetRiskConfiguration](#) in der AWS CLI Befehlsreferenz.

set-ui-customization

Das folgende Codebeispiel zeigt die Verwendung `set-ui-customization`.

AWS CLI

Um die Anpassung der Benutzeroberfläche festzulegen

In diesem Beispiel wird die CSS-Einstellung für einen Benutzerpool angepasst.

Befehl:

```
aws cognito-idp set-ui-customization --user-pool-id us-west-2_aaaaaaaaa --
css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 300;\n}\n.textDescription-customizable
{\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-
size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n\tpadding-
bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-customizable
{\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
{\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n
\theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
#D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
```

```
#555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n\ninputField-  
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n\nidpButton-  
customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-  
bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:  
#46b8da;\n}\n\nidpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:  
#31b0d5;\n}\n\nsocialButton-customizable {\n\theight: 40px;\n\ttext-align: left;  
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n\nredirect-customizable {\n\ttext-  
align: center;\n}\n\npasswordCheck-notValid-customizable {\n\tcolor: #DF3312;  
\n}\n\npasswordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n\nbackground-  
customizable {\n\tbackground-color: #faf;\n}\n\n"
```

Ausgabe:

```
{
  "UICustomization": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientId": "ALL",
    "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;  
\n}\n\nbanner-customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color:  
lightgray;\n}\n\nlabel-customizable {\n\tfont-weight: 300;\n}\n\ntextDescription-  
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;  
\n\tfont-size: 16px;\n}\n\nidpDescription-customizable {\n\tpadding-top: 10px;\n  
\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n\nlegalText-  
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n\nsubmitButton-customizable  
\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n  
\n\theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;  
\n}\n\nsubmitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:  
#286090;\n}\n\nerrorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;  
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:  
#D64958;\n}\n\ninputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:  
#555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n\ninputField-  
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n\nidpButton-  
customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-  
bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:  
#46b8da;\n}\n\nidpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:  
#31b0d5;\n}\n\nsocialButton-customizable {\n\theight: 40px;\n\ttext-align: left;  
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n\nredirect-customizable {\n\ttext-  
align: center;\n}\n\npasswordCheck-notValid-customizable {\n\tcolor: #DF3312;  
\n}\n\npasswordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n\nbackground-  
customizable {\n\tbackground-color: #faf;\n}\n\n",
    "CSSVersion": "20190129172214"
  }
}
```

- API-Einheiten finden Sie unter [SetUiCustomization AWS CLI](#) Befehlsreferenz.

set-user-mfa-preference

Das folgende Codebeispiel zeigt die Verwendung `set-user-mfa-preference`.

AWS CLI

Um MFA Benutzereinstellungen festzulegen

Im folgenden `set-user-mfa-preference` Beispiel werden die MFA Lieferoptionen geändert. Es ändert das MFA Liefermedium auf SMS.

```
aws cognito-idp set-user-mfa-preference \  
  --access-token "eyJra12345EXAMPLE" \  
  --software-token-mfa-settings Enabled=true,PreferredMfa=true \  
  --sms-mfa-settings Enabled=false,PreferredMfa=false
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen MFA zu einem Benutzerpool](#) im Amazon Cognito Developer Guide.

- API-Einheiten finden Sie unter [SetUserMfaPreference AWS CLI](#) Befehlsreferenz.

set-user-settings

Das folgende Codebeispiel zeigt die Verwendung `set-user-settings`.

AWS CLI

Um Benutzereinstellungen festzulegen

In diesem Beispiel wird die MFA Versandpräferenz auf festgelegt EMAIL.

Befehl:

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-  
options DeliveryMedium=EMAIL
```

- API-Einheiten finden Sie [SetUserSettings](#) in der AWS CLI Befehlsreferenz.

sign-up

Das folgende Codebeispiel zeigt die Verwendung `sign-up`.

AWS CLI

Benutzer registrieren

In diesem Beispiel wird `jane@example.com` registriert.

Befehl:

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4f13mg5e62d9ado --  
username jane@example.com --password PASSWORD --user-attributes  
Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Ausgabe:

```
{  
  "UserConfirmed": false,  
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"  
}
```

- API-Einheiten finden Sie [SignUp](#) in der AWS CLI Befehlsreferenz.

start-user-import-job

Das folgende Codebeispiel zeigt die Verwendung `start-user-import-job`.

AWS CLI

Um einen Benutzerimportjob zu starten

In diesem Beispiel wird ein Benutzereingabebefehl gestartet.

Weitere Informationen zum Importieren von Benutzern finden Sie unter [Benutzer aus einer CSV-Datei in Benutzerpools importieren](#).

Befehl:

```
aws cognito-idp start-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-  
id import-TZqNQvDRnW
```

Ausgabe:

```
{
  "UserImportJob": {
    "JobName": "import-Test10",
    "JobId": "import-lmpxS0uIzH",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548278378.928,
    "StartDate": 1548278397.334,
    "Status": "Pending",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

- API-Einheiten finden Sie [StartUserImportJob](#) unter AWS CLI Befehlsreferenz.

stop-user-import-job

Das folgende Codebeispiel zeigt die Verwendung `stop-user-import-job`.

AWS CLI

Um einen Benutzerimportjob zu beenden

In diesem Beispiel wird ein Benutzereingabeauftrag beendet.

Weitere Informationen zum Importieren von Benutzern finden Sie unter [Benutzer aus einer CSV Datei in Benutzerpools importieren](#).

Befehl:

```
aws cognito-idp stop-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

Ausgabe:

```
{
```

```

  "UserImportJob": {
    "JobName": "import-Test5",
    "JobId": "import-Fx0kARISFL",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548278576.259,
    "StartDate": 1548278623.366,
    "CompletionDate": 1548278626.741,
    "Status": "Stopped",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0,
    "CompletionMessage": "The Import Job was stopped by the developer."
  }
}

```

- API-Einheiten finden Sie [StopUserImportJob](#) unter AWS CLI Befehlsreferenz.

update-auth-event-feedback

Das folgende Codebeispiel zeigt die Verwendung `update-auth-event-feedback`.

AWS CLI

Um das Feedback zu Authentifizierungsereignissen zu aktualisieren

In diesem Beispiel wird das Feedback zu Autorisierungsereignissen aktualisiert. Es markiert das Ereignis als „Gültig“.

Befehl:

```

aws cognito-idp update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --event-id EVENT_ID --feedback-token FEEDBACK_TOKEN --
feedback-value "Valid"

```

- API-Einheiten finden Sie [UpdateAuthEventFeedback](#) in der AWS CLI Befehlsreferenz.

update-device-status

Das folgende Codebeispiel zeigt die Verwendung `update-device-status`.

AWS CLI

Um den Gerätestatus zu aktualisieren

In diesem Beispiel wird der Status eines Geräts auf „not_remembered“ aktualisiert.

Befehl:

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-key DEVICE_KEY --device-remembered-status "not_remembered"
```

- API-Einheiten finden Sie [UpdateDeviceStatus](#) in der AWS CLI Befehlsreferenz.

update-group

Das folgende Codebeispiel zeigt die Verwendung `update-group`.

AWS CLI

Um eine Gruppe zu aktualisieren

In diesem Beispiel werden die Beschreibung und der Vorrang für `MyGroup` aktualisiert.

Befehl:

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup --description "New description" --precedence 2
```

Ausgabe:

```
{
  "Group": {
    "GroupName": "MyGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "New description",
    "RoleArn": "arn:aws:iam::111111111111:role/MyRole",
    "Precedence": 2,
    "LastModifiedDate": 1548800862.812,
    "CreationDate": 1548097827.125
  }
}
```

- API-Einheiten finden Sie [UpdateGroup](#) in der AWS CLI Befehlsreferenz.

update-resource-server

Das folgende Codebeispiel zeigt die Verwendung `update-resource-server`.

AWS CLI

Um einen Ressourcenserver zu aktualisieren

In diesem Beispiel wird der Ressourcenserver Wetter aktualisiert. Es fügt einen neuen Bereich hinzu.

Befehl:

```
aws cognito-idp update-resource-server --user-pool-id us-west-2_aaaaaaaaa
--identifier weather.example.com --name Weather --scopes
ScopeName=NewScope,ScopeDescription="New scope description"
```

Ausgabe:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Happy",
    "Scopes": [
      {
        "ScopeName": "NewScope",
        "ScopeDescription": "New scope description"
      }
    ]
  }
}
```

- API-Einheiten finden Sie [UpdateResourceServer](#) in der AWS CLI Befehlsreferenz.

update-user-attributes

Das folgende Codebeispiel zeigt die Verwendung `update-user-attributes`.

AWS CLI

Um Benutzerattribute zu aktualisieren

In diesem Beispiel wird das Benutzerattribut „Nickname“ aktualisiert.

Befehl:

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
Name="nickname",Value="Dan"
```

- API-Einheiten finden Sie [UpdateUserAttributes](#) in der AWS CLI Befehlsreferenz.

update-user-pool-client

Das folgende Codebeispiel zeigt die Verwendung `update-user-pool-client`.

AWS CLI

Um einen Benutzerpool-Client zu aktualisieren

In diesem Beispiel wird der Name eines Benutzerpool-Clients aktualisiert. Außerdem wird das beschreibbare Attribut „Nickname“ hinzugefügt.

Befehl:

```
aws cognito-idp update-user-pool-client --user-pool-id us-west-2_aaaaaaaa --
client-id 3n4b5urk1ft4fl3mg5e62d9ado --client-name NewClientName --write-
attributes nickname
```

Ausgabe:

```
{
  "UserPoolClient": {
    "UserPoolId": "us-west-2_aaaaaaaa",
    "ClientName": "NewClientName",
    "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
    "LastModifiedDate": 1548802761.334,
    "CreationDate": 1548178931.258,
    "RefreshTokenValidity": 30,
    "WriteAttributes": [
      "nickname"
    ],
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

- API-Einheiten finden Sie [UpdateUserPoolClient](#) in der AWS CLI Befehlsreferenz.

update-user-pool

Das folgende Codebeispiel zeigt die Verwendung `update-user-pool`.

AWS CLI

Um einen Benutzerpool zu aktualisieren

Im folgenden `update-user-pool` Beispiel wird ein Benutzerpool mit Beispielsyntax für jede der verfügbaren Konfigurationsoptionen geändert. Um einen Benutzerpool zu aktualisieren, müssen Sie alle zuvor konfigurierten Optionen angeben. Andernfalls werden sie auf einen Standardwert zurückgesetzt.

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
  --policies PasswordPolicy=
  \{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,RequireSym
  \
  --deletion-protection ACTIVE \
  --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
  --auto-verified-attributes "phone_number" "email" \
  --verification-message-template \{"SmsMessage\":"Your code is
#####"\,"EmailMessage\":"Your code is {#####}"\,"EmailSubject\":"Your
verification code"\,"EmailMessageByLink\":"Click {##here##} to verify
your email address."\,"EmailSubjectByLink\":"Your verification link"\,
\DefaultEmailOption\":"CONFIRM_WITH_LINK\"} \
  --sms-authentication-message "Your code is {#####}" \
  --user-attribute-update-settings
  AttributesRequireVerificationBeforeUpdate="email","phone_number" \
  --mfa-configuration "OPTIONAL" \
  --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
  \
  --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="admin
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",ConfigurationSet
configuration-set" \
  --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-role/
SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
```

```

--admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
{#####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
{#####}"\",EmailSubject=\""Welcome to MyMobileGame""\"} \
--user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
--admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
{#####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
{#####}"\",EmailSubject=\""Welcome to MyMobileGame""\"} \
--user-pool-add-ons AdvancedSecurityMode="AUDIT" \
--account-recovery-setting RecoveryMechanisms=
\[\{Priority=1,Name="verified_email"\},\{Priority=2,Name="verified_phone_number"\}\]

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktualisieren der Benutzerpool-Konfiguration](#) im Amazon Cognito Developer Guide.

- API-Einheiten finden Sie [UpdateUserPool](#) in der AWS CLI Befehlsreferenz.

Amazon Comprehend Comprehend-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Comprehend Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-detect-dominant-language

Das folgende Codebeispiel zeigt die Verwendung `batch-detect-dominant-language`.

AWS CLI

Um die dominante Sprache mehrerer Eingabetexte zu erkennen

Das folgende `batch-detect-dominant-language` Beispiel analysiert mehrere Eingabetexte und gibt jeweils die dominante Sprache zurück. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend batch-detect-dominant-language \
  --text-list "Physics is the natural science that involves the study of matter
  and its motion and behavior through space and time, along with related concepts
  such as energy and force."
```

Ausgabe:

```
{
  "ResultList": [
    {
      "Index": 0,
      "Languages": [
        {
          "LanguageCode": "en",
          "Score": 0.9986501932144165
        }
      ]
    }
  ],
  "ErrorList": []
}
```

Weitere Informationen finden Sie unter [Dominant Language](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [BatchDetectDominantLanguage](#) in der AWS CLI Befehlsreferenz.

`batch-detect-entities`

Das folgende Codebeispiel zeigt die Verwendung `batch-detect-entities`.

AWS CLI

Um Entitäten aus mehreren Eingabetexten zu erkennen

Das folgende `batch-detect-entities` Beispiel analysiert mehrere Eingabetexte und gibt jeweils die benannten Entitäten zurück. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend batch-detect-entities \  
  --language-code en \  
  --text-list "Dear Jane, Your AnyCompany Financial Services LLC credit card  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to  
Alice at AnySpa@example.com."
```

Ausgabe:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Entities": [  
        {  
          "Score": 0.9985517859458923,  
          "Type": "PERSON",  
          "Text": "Jane",  
          "BeginOffset": 5,  
          "EndOffset": 9  
        },  
        {  
          "Score": 0.9767839312553406,  
          "Type": "ORGANIZATION",  
          "Text": "AnyCompany Financial Services, LLC",  
          "BeginOffset": 16,  
          "EndOffset": 50  
        },  
        {  
          "Score": 0.9856694936752319,  
          "Type": "OTHER",  
          "Text": "1111-XXXX-1111-XXXX",  
          "BeginOffset": 71,  
          "EndOffset": 90  
        },  
        {  
          "Score": 0.9652159810066223,  
          "Type": "QUANTITY",  
          "Text": ".53",
```

```
        "BeginOffset": 116,  
        "EndOffset": 119  
    },  
    {  
        "Score": 0.9986667037010193,  
        "Type": "DATE",  
        "Text": "July 31st",  
        "BeginOffset": 135,  
        "EndOffset": 144  
    }  
]  
},  
{  
    "Index": 1,  
    "Entities": [  
        {  
            "Score": 0.720084547996521,  
            "Type": "ORGANIZATION",  
            "Text": "Sunshine Spa",  
            "BeginOffset": 33,  
            "EndOffset": 45  
        },  
        {  
            "Score": 0.9865870475769043,  
            "Type": "LOCATION",  
            "Text": "123 Main St",  
            "BeginOffset": 47,  
            "EndOffset": 58  
        },  
        {  
            "Score": 0.5895616412162781,  
            "Type": "LOCATION",  
            "Text": "Anywhere",  
            "BeginOffset": 60,  
            "EndOffset": 68  
        },  
        {  
            "Score": 0.6809214353561401,  
            "Type": "PERSON",  
            "Text": "Alice",  
            "BeginOffset": 75,  
            "EndOffset": 80  
        },  
        {
```



```

        "Score": 0.9979087114334106,
        "Type": "OTHER",
        "Text": "AnySpa@example.com",
        "BeginOffset": 84,
        "EndOffset": 99
      }
    ]
  },
  "ErrorList": []
}

```

Weitere Informationen finden Sie unter [Entitäten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [BatchDetectEntities](#) in der AWS CLI Befehlsreferenz.

batch-detect-key-phrases

Das folgende Codebeispiel zeigt die Verwendung `batch-detect-key-phrases`.

AWS CLI

Um Schlüsselphrasen mehrerer Texteingaben zu erkennen

Das folgende `batch-detect-key-phrases` Beispiel analysiert mehrere Eingabetexte und gibt die jeweiligen Schlüsselnamenphrasen zurück. Der Konfidenzwert des vortrainierten Modells für jede Vorhersage wird ebenfalls ausgegeben.

```

aws comprehend batch-detect-key-phrases \
  --language-code en \
  --text-list "Hello Zhang Wei, I am John, writing to you about the trip for
next Saturday." "Dear Jane, Your AnyCompany Financial Services LLC credit card
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to
Alice at AnySpa@example.com."

```

Ausgabe:

```

{
  "ResultList": [
    {
      "Index": 0,
      "KeyPhrases": [

```

```
    {
      "Score": 0.99700927734375,
      "Text": "Zhang Wei",
      "BeginOffset": 6,
      "EndOffset": 15
    },
    {
      "Score": 0.9929308891296387,
      "Text": "John",
      "BeginOffset": 22,
      "EndOffset": 26
    },
    {
      "Score": 0.9997230172157288,
      "Text": "the trip",
      "BeginOffset": 49,
      "EndOffset": 57
    },
    {
      "Score": 0.9999470114707947,
      "Text": "next Saturday",
      "BeginOffset": 62,
      "EndOffset": 75
    }
  ]
},
{
  "Index": 1,
  "KeyPhrases": [
    {
      "Score": 0.8358274102210999,
      "Text": "Dear Jane",
      "BeginOffset": 0,
      "EndOffset": 9
    },
    {
      "Score": 0.989359974861145,
      "Text": "Your AnyCompany Financial Services",
      "BeginOffset": 11,
      "EndOffset": 45
    },
    {
      "Score": 0.8812323808670044,
      "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
```

```
        "BeginOffset": 47,
        "EndOffset": 90
    },
    {
        "Score": 0.9999381899833679,
        "Text": "a minimum payment",
        "BeginOffset": 95,
        "EndOffset": 112
    },
    {
        "Score": 0.9997439980506897,
        "Text": ".53",
        "BeginOffset": 116,
        "EndOffset": 119
    },
    {
        "Score": 0.996875524520874,
        "Text": "July 31st",
        "BeginOffset": 135,
        "EndOffset": 144
    }
]
},
{
    "Index": 2,
    "KeyPhrases": [
        {
            "Score": 0.9990295767784119,
            "Text": "customer feedback",
            "BeginOffset": 12,
            "EndOffset": 29
        },
        {
            "Score": 0.9994127750396729,
            "Text": "Sunshine Spa",
            "BeginOffset": 33,
            "EndOffset": 45
        },
        {
            "Score": 0.9892991185188293,
            "Text": "123 Main St",
            "BeginOffset": 47,
            "EndOffset": 58
        }
    ],
}
```

```

    {
      "Score": 0.9969810843467712,
      "Text": "Alice",
      "BeginOffset": 75,
      "EndOffset": 80
    },
    {
      "Score": 0.9703696370124817,
      "Text": "AnySpa@example.com",
      "BeginOffset": 84,
      "EndOffset": 99
    }
  ]
},
"ErrorList": []
}

```

Weitere Informationen finden Sie unter [Schlüsselbegriffe](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [BatchDetectKeyPhrases](#) in der AWS CLI Befehlsreferenz.

batch-detect-sentiment

Das folgende Codebeispiel zeigt die Verwendung `batch-detect-sentiment`.

AWS CLI

Um die vorherrschende Stimmung in mehreren Eingabetexten zu erkennen

Im folgenden `batch-detect-sentiment` Beispiel werden mehrere Eingabetexte analysiert und die vorherrschende Stimmung (POSITIVE, NEUTRAL, oder MIXEDNEGATIVE, für jeden Text) zurückgegeben.

```

aws comprehend batch-detect-sentiment \
  --text-list "That movie was very boring, I can't believe it was over four hours long." "It is a beautiful day for hiking today." "My meal was okay, I'm excited to try other restaurants." \
  --language-code en

```

Ausgabe:

```

{
  "ResultList": [
    {
      "Index": 0,
      "Sentiment": "NEGATIVE",
      "SentimentScore": {
        "Positive": 0.00011316669406369328,
        "Negative": 0.9995445609092712,
        "Neutral": 0.00014722718333359808,
        "Mixed": 0.00019498742767609656
      }
    },
    {
      "Index": 1,
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Positive": 0.9981263279914856,
        "Negative": 0.00015240783977787942,
        "Neutral": 0.0013876151060685515,
        "Mixed": 0.00033366199932061136
      }
    },
    {
      "Index": 2,
      "Sentiment": "MIXED",
      "SentimentScore": {
        "Positive": 0.15930435061454773,
        "Negative": 0.11471917480230331,
        "Neutral": 0.26897063851356506,
        "Mixed": 0.45700588822364807
      }
    }
  ],
  "ErrorList": []
}

```

Weitere Informationen finden Sie unter [Sentiment](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [BatchDetectSentiment](#).AWS CLI

batch-detect-syntax

Das folgende Codebeispiel zeigt die Verwendung `batch-detect-syntax`.

AWS CLI

Um die Syntax und Wortarten von Wörtern in mehreren Eingabetexten zu untersuchen

Im folgenden `batch-detect-syntax` Beispiel wird die Syntax mehrerer Eingabetexte analysiert und die verschiedenen Wortarten zurückgegeben. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend batch-detect-syntax \  
  --text-list "It is a beautiful day." "Can you please pass the salt?" "Please pay  
the bill before the 31st." \  
  --language-code en
```

Ausgabe:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "SyntaxTokens": [  
        {  
          "TokenId": 1,  
          "Text": "It",  
          "BeginOffset": 0,  
          "EndOffset": 2,  
          "PartOfSpeech": {  
            "Tag": "PRON",  
            "Score": 0.9999740719795227  
          }  
        },  
        {  
          "TokenId": 2,  
          "Text": "is",  
          "BeginOffset": 3,  
          "EndOffset": 5,  
          "PartOfSpeech": {  
            "Tag": "VERB",  
            "Score": 0.999937117099762  
          }  
        },  
        {  
          "TokenId": 3,  
          "Text": "a",
```

```
        "BeginOffset": 6,
        "EndOffset": 7,
        "PartOfSpeech": {
            "Tag": "DET",
            "Score": 0.9999926686286926
        }
    },
    {
        "TokenId": 4,
        "Text": "beautiful",
        "BeginOffset": 8,
        "EndOffset": 17,
        "PartOfSpeech": {
            "Tag": "ADJ",
            "Score": 0.9987891912460327
        }
    },
    {
        "TokenId": 5,
        "Text": "day",
        "BeginOffset": 18,
        "EndOffset": 21,
        "PartOfSpeech": {
            "Tag": "NOUN",
            "Score": 0.9999778866767883
        }
    },
    {
        "TokenId": 6,
        "Text": ".",
        "BeginOffset": 21,
        "EndOffset": 22,
        "PartOfSpeech": {
            "Tag": "PUNCT",
            "Score": 0.9999974966049194
        }
    }
]
},
{
    "Index": 1,
    "SyntaxTokens": [
        {
            "TokenId": 1,
```

```
    "Text": "Can",
    "BeginOffset": 0,
    "EndOffset": 3,
    "PartOfSpeech": {
      "Tag": "AUX",
      "Score": 0.9999770522117615
    }
  },
  {
    "TokenId": 2,
    "Text": "you",
    "BeginOffset": 4,
    "EndOffset": 7,
    "PartOfSpeech": {
      "Tag": "PRON",
      "Score": 0.9999986886978149
    }
  },
  {
    "TokenId": 3,
    "Text": "please",
    "BeginOffset": 8,
    "EndOffset": 14,
    "PartOfSpeech": {
      "Tag": "INTJ",
      "Score": 0.9681622385978699
    }
  },
  {
    "TokenId": 4,
    "Text": "pass",
    "BeginOffset": 15,
    "EndOffset": 19,
    "PartOfSpeech": {
      "Tag": "VERB",
      "Score": 0.9999874830245972
    }
  },
  {
    "TokenId": 5,
    "Text": "the",
    "BeginOffset": 20,
    "EndOffset": 23,
    "PartOfSpeech": {
```



```
        "Tag": "DET",
        "Score": 0.9999827146530151
      }
    },
    {
      "TokenId": 6,
      "Text": "salt",
      "BeginOffset": 24,
      "EndOffset": 28,
      "PartOfSpeech": {
        "Tag": "NOUN",
        "Score": 0.9995040893554688
      }
    },
    {
      "TokenId": 7,
      "Text": "?",
      "BeginOffset": 28,
      "EndOffset": 29,
      "PartOfSpeech": {
        "Tag": "PUNCT",
        "Score": 0.999998152256012
      }
    }
  ]
},
{
  "Index": 2,
  "SyntaxTokens": [
    {
      "TokenId": 1,
      "Text": "Please",
      "BeginOffset": 0,
      "EndOffset": 6,
      "PartOfSpeech": {
        "Tag": "INTJ",
        "Score": 0.9997857809066772
      }
    }
  ],
  {
    "TokenId": 2,
    "Text": "pay",
    "BeginOffset": 7,
    "EndOffset": 10,
```

```
        "PartOfSpeech": {
            "Tag": "VERB",
            "Score": 0.9999252557754517
        }
    },
    {
        "TokenId": 3,
        "Text": "the",
        "BeginOffset": 11,
        "EndOffset": 14,
        "PartOfSpeech": {
            "Tag": "DET",
            "Score": 0.9999842643737793
        }
    },
    {
        "TokenId": 4,
        "Text": "bill",
        "BeginOffset": 15,
        "EndOffset": 19,
        "PartOfSpeech": {
            "Tag": "NOUN",
            "Score": 0.9999588131904602
        }
    },
    {
        "TokenId": 5,
        "Text": "before",
        "BeginOffset": 20,
        "EndOffset": 26,
        "PartOfSpeech": {
            "Tag": "ADP",
            "Score": 0.9958304762840271
        }
    },
    {
        "TokenId": 6,
        "Text": "the",
        "BeginOffset": 27,
        "EndOffset": 30,
        "PartOfSpeech": {
            "Tag": "DET",
            "Score": 0.9999947547912598
        }
    }
}
```

```

    },
    {
      "TokenId": 7,
      "Text": "31st",
      "BeginOffset": 31,
      "EndOffset": 35,
      "PartOfSpeech": {
        "Tag": "NOUN",
        "Score": 0.9924124479293823
      }
    },
    {
      "TokenId": 8,
      "Text": ".",
      "BeginOffset": 35,
      "EndOffset": 36,
      "PartOfSpeech": {
        "Tag": "PUNCT",
        "Score": 0.9999955892562866
      }
    }
  ]
}
],
"ErrorList": []
}

```

Weitere Informationen finden Sie unter [Syntaxanalyse](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [BatchDetectSyntax](#) in der AWS CLI Befehlsreferenz.

batch-detect-targeted-sentiment

Das folgende Codebeispiel zeigt die Verwendung `batch-detect-targeted-sentiment`.

AWS CLI

Um das Sentiment und jede benannte Entität für mehrere Eingabetexte zu erkennen

Das folgende `batch-detect-targeted-sentiment` Beispiel analysiert mehrere Eingabetexte und gibt die benannten Entitäten zusammen mit der jeweils vorherrschenden Stimmung zurück. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend batch-detect-targeted-sentiment \
  --language-code en \
  --text-list "That movie was really boring, the original was way more
entertaining" "The trail is extra beautiful today." "My meal was just okay."
```

Ausgabe:

```
{
  "ResultList": [
    {
      "Index": 0,
      "Entities": [
        {
          "DescriptiveMentionIndex": [
            0
          ],
          "Mentions": [
            {
              "Score": 0.9999009966850281,
              "GroupScore": 1.0,
              "Text": "movie",
              "Type": "MOVIE",
              "MentionSentiment": {
                "Sentiment": "NEGATIVE",
                "SentimentScore": {
                  "Positive": 0.13887299597263336,
                  "Negative": 0.8057460188865662,
                  "Neutral": 0.05525200068950653,
                  "Mixed": 0.00012799999967683107
                }
              }
            },
            {
              "BeginOffset": 5,
              "EndOffset": 10
            }
          ]
        }
      ],
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "Score": 0.9921110272407532,
```

```

        "GroupScore": 1.0,
        "Text": "original",
        "Type": "MOVIE",
        "MentionSentiment": {
            "Sentiment": "POSITIVE",
            "SentimentScore": {
                "Positive": 0.9999989867210388,
                "Negative": 9.999999974752427e-07,
                "Neutral": 0.0,
                "Mixed": 0.0
            }
        },
        "BeginOffset": 34,
        "EndOffset": 42
    }
]
}
],
{
    "Index": 1,
    "Entities": [
        {
            "DescriptiveMentionIndex": [
                0
            ],
            "Mentions": [
                {
                    "Score": 0.7545599937438965,
                    "GroupScore": 1.0,
                    "Text": "trail",
                    "Type": "OTHER",
                    "MentionSentiment": {
                        "Sentiment": "POSITIVE",
                        "SentimentScore": {
                            "Positive": 1.0,
                            "Negative": 0.0,
                            "Neutral": 0.0,
                            "Mixed": 0.0
                        }
                    }
                }
            ],
            "BeginOffset": 4,
            "EndOffset": 9
        }
    ]
}

```

```

    ]
  },
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
        "Score": 0.9999960064888,
        "GroupScore": 1.0,
        "Text": "today",
        "Type": "DATE",
        "MentionSentiment": {
          "Sentiment": "NEUTRAL",
          "SentimentScore": {
            "Positive": 9.000000318337698e-06,
            "Negative": 1.999999949504854e-06,
            "Neutral": 0.9999859929084778,
            "Mixed": 3.99999989900971e-06
          }
        }
      },
      {
        "BeginOffset": 29,
        "EndOffset": 34
      }
    ]
  }
]
},
{
  "Index": 2,
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "Score": 0.9999880194664001,
          "GroupScore": 1.0,
          "Text": "My",
          "Type": "PERSON",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {

```

```

        "Positive": 0.0,
        "Negative": 0.0,
        "Neutral": 1.0,
        "Mixed": 0.0
      }
    },
    "BeginOffset": 0,
    "EndOffset": 2
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9995260238647461,
      "GroupScore": 1.0,
      "Text": "meal",
      "Type": "OTHER",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Positive": 0.04695599898695946,
          "Negative": 0.003226999891921878,
          "Neutral": 0.6091709733009338,
          "Mixed": 0.34064599871635437
        }
      }
    },
    "BeginOffset": 3,
    "EndOffset": 7
  ]
}
]
}
],
"ErrorList": []
}

```

Weitere Informationen finden Sie unter [Targeted Sentiment](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [BatchDetectTargetedSentiment](#). AWS CLI

classify-document

Das folgende Codebeispiel zeigt die Verwendung `classify-document`.

AWS CLI

Um ein Dokument mit einem modellspezifischen Endpunkt zu klassifizieren

Im folgenden `classify-document` Beispiel wird ein Dokument mit einem Endpunkt eines benutzerdefinierten Modells klassifiziert. Das Modell in diesem Beispiel wurde anhand eines Datensatzes trainiert, der SMS-Nachrichten enthält, die als Spam oder Nicht-Spam oder „Ham“ gekennzeichnet sind.

```
aws comprehend classify-document \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint \  
  --text "CONGRATULATIONS! TXT 1235550100 to win $5000"
```

Ausgabe:

```
{  
  "Classes": [  
    {  
      "Name": "spam",  
      "Score": 0.9998599290847778  
    },  
    {  
      "Name": "ham",  
      "Score": 0.00014001205272506922  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ClassifyDocument](#) in der AWS CLI Befehlsreferenz.

contains-pii-entities

Das folgende Codebeispiel zeigt die Verwendung `contains-pii-entities`.

AWS CLI

Um den Eingabetext auf das Vorhandensein von PII Informationen zu analysieren

Im folgenden `contains-pii-entities` Beispiel wird der Eingabetext auf das Vorhandensein personenbezogener Daten (PII) analysiert und die Bezeichnungen identifizierter PII Entitätstypen wie Name, Adresse, Bankkontonummer oder Telefonnummer zurückgegeben.

```
aws comprehend contains-pii-entities \
  --language-code en \
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
credit card
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
July 31st. Based on your autopay settings,
we will withdraw your payment on the due date from your bank account number
XXXXXX1111 with the routing number XXXXX0000.
Customer feedback for Sunshine Spa, 100 Main St, Anywhere. Send comments to
Alice at AnySpa@example.com."
```

Ausgabe:

```
{
  "Labels": [
    {
      "Name": "NAME",
      "Score": 1.0
    },
    {
      "Name": "EMAIL",
      "Score": 1.0
    },
    {
      "Name": "BANK_ACCOUNT_NUMBER",
      "Score": 0.9995794296264648
    },
    {
      "Name": "BANK_ROUTING",
      "Score": 0.9173126816749573
    }
  ]
}
```

```

    {
      "Name": "CREDIT_DEBIT_NUMBER",
      "Score": 1.0
    }
  }

```

Weitere Informationen finden Sie unter [Persönlich identifizierbare Informationen \(PII\)](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ContainsPiiEntities](#) in der AWS CLI Befehlsreferenz.

create-dataset

Das folgende Codebeispiel zeigt die Verwendung `create-dataset`.

AWS CLI

Um einen Flywheel-Datensatz zu erstellen

Im folgenden `create-dataset` Beispiel wird ein Datensatz für ein Schwungrad erstellt. Dieser Datensatz wird als zusätzliche Trainingsdaten verwendet, wie im `--dataset-type` Tag angegeben.

```

aws comprehend create-dataset \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity \
  --dataset-name example-dataset \
  --dataset-type "TRAIN" \
  --input-data-config file://inputConfig.json

```

Inhalt von `file://inputConfig.json`:

```

{
  "DataFormat": "COMPREHEND_CSV",
  "DocumentClassifierInputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/training-data.csv"
  }
}

```

Ausgabe:

```

{

```

```
"DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset"  
}
```

Weitere Informationen finden Sie unter [Flywheel Overview](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [CreateDataset](#) in AWS CLI der Befehlsreferenz.

create-document-classifier

Das folgende Codebeispiel zeigt die Verwendung `create-document-classifier`.

AWS CLI

Um einen Dokumentenklassifizierer zur Kategorisierung von Dokumenten zu erstellen

Mit dem folgenden `create-document-classifier` Beispiel wird der Trainingsprozess für ein Dokumentenklassifizierungsmodell gestartet. Die Trainingsdatendatei, `training.csv`, befindet sich am `--input-data-config` Tag. `training.csv` ist ein zweispaltiges Dokument, in dem die Bezeichnungen oder Klassifizierungen in der ersten Spalte und die Dokumente in der zweiten Spalte angegeben sind.

```
aws comprehend create-document-classifier \  
  --document-classifier-name example-classifier \  
  --data-access-arn arn:aws:comprehend:us-west-2:111122223333:pii-entities-  
detection-job/123456abcdeb0e11022f22a11EXAMPLE \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --language-code en
```

Ausgabe:

```
{  
  "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [CreateDocumentClassifier](#) in der AWS CLI Befehlsreferenz.

create-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-endpoint`.

AWS CLI

Um einen Endpunkt für ein benutzerdefiniertes Modell zu erstellen

Im folgenden `create-endpoint` Beispiel wird ein Endpunkt für synchrone Inferenz für ein zuvor trainiertes benutzerdefiniertes Modell erstellt.

```
aws comprehend create-endpoint \  
  --endpoint-name example-classifier-endpoint-1 \  
  --model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier \  
  --desired-inference-units 1
```

Ausgabe:

```
{  
  "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint-1"  
}
```

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [CreateEndpoint](#) AWS CLI

create-entity-recognizer

Das folgende Codebeispiel zeigt die Verwendung `create-entity-recognizer`.

AWS CLI

Um einen benutzerdefinierten Entity Recognizer zu erstellen

Mit dem folgenden `create-entity-recognizer` Beispiel wird der Trainingsprozess für ein benutzerdefiniertes Entitätserkennungsmodell gestartet. In diesem Beispiel wird eine CSV-Datei mit Trainingsdokumenten und eine CSV-Entitätsliste verwendet, `entity_list.csv` um das

Modell zu trainieren. `raw_text.csv` `entity-list.csv` enthält die folgenden Spalten: Text und Typ.

```
aws comprehend create-entity-recognizer \
  --recognizer-name example-entity-recognizer
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role \
  --input-data-config "EntityTypes=[{Type=DEVICE}], Documents={S3Uri=s3://DOC-EXAMPLE-BUCKET/trainingdata/raw_text.csv}, EntityList={S3Uri=s3://DOC-EXAMPLE-BUCKET/trainingdata/entity_list.csv}"
  --language-code en
```

Ausgabe:

```
{
  "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-entity-recognizer/entityrecognizer1"
}
```

Weitere Informationen finden Sie unter [Custom Entity Recognition](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [CreateEntityRecognizer AWS CLI](#) Befehlsreferenz.

create-flywheel

Das folgende Codebeispiel zeigt die Verwendung `create-flywheel`.

AWS CLI

Um ein Schwungrad zu erstellen

Im folgenden `create-flywheel` Beispiel wird ein Schwungrad erstellt, um das fortlaufende Training eines Modells zur Dokumentenklassifizierung oder zur Erkennung von Entitäten zu koordinieren. Das Schwungrad in diesem Beispiel wurde erstellt, um ein vorhandenes, durch das Tag spezifiziertes trainiertes Modell zu verwalten. `--active-model-arn` Wenn das Schwungrad erstellt wird, wird am Tag ein Data Lake erstellt. `--input-data-lake`

```
aws comprehend create-flywheel \
  --flywheel-name example-flywheel \
  --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-model/version/1 \
```

```
--data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role \  
--data-lake-s3-uri "s3://DOC-EXAMPLE-BUCKET"
```

Ausgabe:

```
{  
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel"  
}
```

Weitere Informationen finden Sie unter [Flywheel Overview](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [CreateFlywheel](#) in AWS CLI der Befehlsreferenz.

delete-document-classifier

Das folgende Codebeispiel zeigt die Verwendung `delete-document-classifier`.

AWS CLI

Um einen benutzerdefinierten Dokumentenklassifikator zu löschen

Im folgenden `delete-document-classifier` Beispiel wird ein benutzerdefiniertes Dokumentklassifizierungsmodell gelöscht.

```
aws comprehend delete-document-classifier \  
--document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteDocumentClassifier](#) AWS CLI

delete-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-endpoint`.

AWS CLI

Um einen Endpunkt für ein benutzerdefiniertes Modell zu löschen

Im folgenden `delete-endpoint` Beispiel wird ein modellspezifischer Endpunkt gelöscht. Alle Endpunkte müssen gelöscht werden, damit das Modell gelöscht werden kann.

```
aws comprehend delete-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteEndpoint](#) AWS CLI

`delete-entity-recognizer`

Das folgende Codebeispiel zeigt die Verwendung `delete-entity-recognizer`.

AWS CLI

Um ein benutzerdefiniertes Entity Recognizer-Modell zu löschen

Im folgenden `delete-entity-recognizer` Beispiel wird ein benutzerdefiniertes Entitätserkennungsmodell gelöscht.

```
aws comprehend delete-entity-recognizer \  
  --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-  
recognizer/example-entity-recognizer-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteEntityRecognizer](#) AWS CLI

`delete-flywheel`

Das folgende Codebeispiel zeigt die Verwendung `delete-flywheel`.

AWS CLI

Um ein Schwungrad zu löschen

Im folgenden `delete-flywheel` Beispiel wird ein Schwungrad gelöscht. Der Data Lake oder das Modell, das dem Schwungrad zugeordnet ist, wird nicht gelöscht.

```
aws comprehend delete-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie in der [Übersicht über Flywheel](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DeleteFlywheel](#) in AWS CLI der Befehlsreferenz.

`delete-resource-policy`

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-policy`.

AWS CLI

Um eine ressourcenbasierte Richtlinie zu löschen

Im folgenden `delete-resource-policy` Beispiel wird eine ressourcenbasierte Richtlinie aus einer Amazon Comprehend Comprehend-Ressource gelöscht.

```
aws comprehend delete-resource-policy \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1/version/1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kopieren von benutzerdefinierten Modellen zwischen AWS Konten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [DeleteResourcePolicy AWS CLI](#) Befehlsreferenz.

`describe-dataset`

Das folgende Codebeispiel zeigt die Verwendung `describe-dataset`.

AWS CLI

Um einen Schwungrad Datensatz zu beschreiben

Im folgenden `describe-dataset` Beispiel werden die Eigenschaften eines Schwungrad-Datensatzes abgerufen.

```
aws comprehend describe-dataset \  
  --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset
```

Ausgabe:

```
{  
  "DatasetProperties": {  
    "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset",  
    "DatasetName": "example-dataset",  
    "DatasetType": "TRAIN",  
    "DatasetS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/  
schemaVersion=1/12345678A123456Z/datasets/example-dataset/20230616T203710Z/",  
    "Status": "CREATING",  
    "CreationTime": "2023-06-16T20:37:10.400000+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Flywheel Overview](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DescribeDataset](#) in AWS CLI der Befehlsreferenz.

describe-document-classification-job

Das folgende Codebeispiel zeigt die Verwendung `describe-document-classification-job`.

AWS CLI

Um einen Job zur Dokumentenklassifizierung zu beschreiben

Im folgenden `describe-document-classification-job` Beispiel werden die Eigenschaften eines asynchronen Dokumentenklassifizierungsauftrags abgerufen.

```
aws comprehend describe-document-classification-job \  
  --job-id example-job-id
```

```
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "DocumentClassificationJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "exampleclassificationjob",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
    "EndTime": "2023-06-14T17:15:58.582000+00:00",
    "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/mymodel/version/1",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-
CLN-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
  }
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DescribeDocumentClassificationJob](#) in der AWS CLI Befehlsreferenz.

describe-document-classifier

Das folgende Codebeispiel zeigt die Verwendung `describe-document-classifier`.

AWS CLI

Um einen Dokumentenklassifikator zu beschreiben

Im folgenden `describe-document-classifier` Beispiel werden die Eigenschaften eines benutzerdefinierten Dokumentklassifizierungsmodells abgerufen.

```
aws comprehend describe-document-classifier \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-1
```

Ausgabe:

```
{  
  "DocumentClassifierProperties": {  
    "DocumentClassifierArn": "arn:aws:comprehend:us-  
west-2:111122223333:document-classifier/example-classifier-1",  
    "LanguageCode": "en",  
    "Status": "TRAINED",  
    "SubmitTime": "2023-06-13T19:04:15.735000+00:00",  
    "EndTime": "2023-06-13T19:42:31.752000+00:00",  
    "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",  
    "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",  
    "InputDataConfig": {  
      "DataFormat": "COMPREHEND_CSV",  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"  
    },  
    "OutputDataConfig": {},  
    "ClassifierMetadata": {  
      "NumberOfLabels": 3,  
      "NumberOfTrainedDocuments": 5016,  
      "NumberOfTestDocuments": 557,  
      "EvaluationMetrics": {  
        "Accuracy": 0.9856,  
        "Precision": 0.9919,  
        "Recall": 0.9459,  
        "F1Score": 0.9673,  
        "MicroPrecision": 0.9856,  
        "MicroRecall": 0.9856,  
        "MicroF1Score": 0.9856,  
        "HammingLoss": 0.0144  
      }  
    },  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role",  
    "Mode": "MULTI_CLASS"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [DescribeDocumentClassifier AWS CLI Befehlsreferenz](#).

describe-dominant-language-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-dominant-language-detection-job`.

AWS CLI

Um einen dominanten Job zur Spracherkennung zu beschreiben.

Im folgenden `describe-dominant-language-detection-job` Beispiel werden die Eigenschaften eines asynchronen Auftrags zur Erkennung dominanter Sprache abgerufen.

```
aws comprehend describe-dominant-language-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "DominantLanguageDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "languageanalysis1",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role"
  }
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeDominantLanguageDetectionJob](#) AWS CLI

describe-endpoint

Das folgende Codebeispiel zeigt die Verwendung `describe-endpoint`.

AWS CLI

Um einen bestimmten Endpunkt zu beschreiben

Im folgenden `describe-endpoint` Beispiel werden die Eigenschaften eines modellspezifischen Endpunkts abgerufen.

```
aws comprehend describe-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
  endpoint/example-classifier-endpoint
```

Ausgabe:

```
{  
  "EndpointProperties": {  
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier-endpoint/example-classifier-endpoint",  
    "Status": "IN_SERVICE",  
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
exampleclassifier1",  
    "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/exampleclassifier1",  
    "DesiredInferenceUnits": 1,  
    "CurrentInferenceUnits": 1,  
    "CreationTime": "2023-06-13T20:32:54.526000+00:00",  
    "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeEndpoint](#) AWS CLI

describe-entities-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-entities-detection-job`.

AWS CLI

Um einen Job zur Erkennung von Entitäten zu beschreiben

Im folgenden `describe-entities-detection-job` Beispiel werden die Eigenschaften eines Auftrags zur Erkennung asynchroner Entitäten abgerufen.

```
aws comprehend describe-entities-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "EntitiesDetectionJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "example-entity-detector",  
    "JobStatus": "COMPLETED",  
    "SubmitTime": "2023-06-08T21:30:15.323000+00:00",  
    "EndTime": "2023-06-08T21:40:23.509000+00:00",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/thefolder/111122223333-  
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"  
    },  
    "LanguageCode": "en",  
    "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/  
AmazonComprehendServiceRole-example-role"  
  }  
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeEntitiesDetectionJob](#) AWS CLI


```

    "NumberOfTrainedDocuments": 1814,
    "NumberOfTestDocuments": 486,
    "EvaluationMetrics": {
      "Precision": 100.0,
      "Recall": 100.0,
      "F1Score": 100.0
    },
    "EntityTypes": [
      {
        "Type": "BUSINESS",
        "EvaluationMetrics": {
          "Precision": 100.0,
          "Recall": 100.0,
          "F1Score": 100.0
        },
        "NumberOfTrainMentions": 1520
      }
    ]
  },
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role",
  "VersionName": "1"
}
}

```

Weitere Informationen finden Sie unter [Custom Entity Recognition](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [DescribeEntityRecognizer AWS CLI Befehlsreferenz](#).

describe-events-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-events-detection-job`.

AWS CLI

Um einen Job zur Erkennung von Ereignissen zu beschreiben.

Im folgenden `describe-events-detection-job` Beispiel werden die Eigenschaften eines Auftrags zur asynchronen Erkennung von Ereignissen abgerufen.

```

aws comprehend describe-events-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE

```


Ausgabe:

```
{
  "EventsDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "events_job_1",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-12T18:45:56.054000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/EventsData",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-
EVENTS-123456abcdeb0e11022f22a11EXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "CORPORATE_MERGER",
      "INVESTMENT_GENERAL"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeEventsDetectionJob](#) AWS CLI

describe-flywheel-iteration

Das folgende Codebeispiel zeigt die Verwendung `describe-flywheel-iteration`.

AWS CLI

Um eine Schwungrad-Iteration zu beschreiben

Im folgenden `describe-flywheel-iteration` Beispiel werden die Eigenschaften einer Schwungrad-Iteration abgerufen.

```
aws comprehend describe-flywheel-iteration \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel \  
  --flywheel-iteration-id 20232222AEXAMPLE
```

Ausgabe:

```
{  
  "FlywheelIterationProperties": {  
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity",  
    "FlywheelIterationId": "20232222AEXAMPLE",  
    "CreationTime": "2023-06-16T21:10:26.385000+00:00",  
    "EndTime": "2023-06-16T23:33:16.827000+00:00",  
    "Status": "COMPLETED",  
    "Message": "FULL_ITERATION: Flywheel iteration performed all functions  
successfully.",  
    "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier/version/1",  
    "EvaluatedModelMetrics": {  
      "AverageF1Score": 0.7742663922375772,  
      "AveragePrecision": 0.8287636394041166,  
      "AverageRecall": 0.7427084833645399,  
      "AverageAccuracy": 0.8795394154118689  
    },  
    "TrainedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier/version/Comprehend-Generated-v1-bb52d585",  
    "TrainedModelMetrics": {  
      "AverageF1Score": 0.9767700253081214,  
      "AveragePrecision": 0.9767700253081214,  
      "AverageRecall": 0.9767700253081214,  
      "AverageAccuracy": 0.9858281665190434  
    },  
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/flywheel-  
entity/schemaVersion=1/20230616T200543Z/evaluation/20230616T211026Z/"  
  }  
}
```

Weitere Informationen finden Sie in der [Übersicht über Flywheel](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DescribeFlywheelIteration](#) in AWS CLI der Befehlsreferenz.

describe-flywheel

Das folgende Codebeispiel zeigt die Verwendung `describe-flywheel`.

AWS CLI

Um ein Schwungrad zu beschreiben

Im folgenden `describe-flywheel` Beispiel werden die Eigenschaften eines Schwungrades abgerufen. In diesem Beispiel ist das dem Flywheel zugeordnete Modell ein benutzerdefiniertes Klassifizierungsmodell, das darauf trainiert ist, Dokumente entweder als Spam oder Nonspam oder als „Ham“ zu klassifizieren.

```
aws comprehend describe-flywheel \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel
```

Ausgabe:

```
{
  "FlywheelProperties": {
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel",
    "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-model/version/1",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role",
    "TaskConfig": {
      "LanguageCode": "en",
      "DocumentClassificationConfig": {
        "Mode": "MULTI_CLASS",
        "Labels": [
          "ham",
          "spam"
        ]
      }
    }
  },
}
```

```

    "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/example-flywheel/
schemaVersion=1/20230616T200543Z/",
    "DataSecurityConfig": {},
    "Status": "ACTIVE",
    "ModelType": "DOCUMENT_CLASSIFIER",
    "CreationTime": "2023-06-16T20:05:43.242000+00:00",
    "LastModifiedTime": "2023-06-16T20:21:43.567000+00:00"
  }
}

```

Weitere Informationen finden Sie unter [Flywheel Overview](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DescribeFlywheel](#) in AWS CLI der Befehlsreferenz.

describe-key-phrases-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-key-phrases-detection-job`.

AWS CLI

Um einen Job zur Erkennung von Schlüsselphrasen zu beschreiben

Im folgenden `describe-key-phrases-detection-job` Beispiel werden die Eigenschaften eines Auftrags zur Erkennung asynchroner Schlüssel ausdrücke abgerufen.

```

aws comprehend describe-key-phrases-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE

```

Ausgabe:

```

{
  "KeyPhrasesDetectionJobProperties": {
    "JobId": "69aa080c00fc68934a6a98f10EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/69aa080c00fc68934a6a98f10EXAMPLE",
    "JobName": "example-key-phrases-detection-job",
    "JobStatus": "COMPLETED",
    "SubmitTime": 1686606439.177,
    "EndTime": 1686606806.157,
    "InputDataConfig": {
      "S3Uri": "s3://dereksbucket1001/EventsData/",
      "InputFormat": "ONE_DOC_PER_LINE"
    }
  }
}

```

```
    },
    "OutputDataConfig": {
      "S3Uri": "s3://dereksbucket1002/testfolder/111122223333-
KP-69aa080c00fc68934a6a98f10EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testrole"
  }
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeKeyPhrasesDetectionJob](#) AWS CLI

describe-pii-entities-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-pii-entities-detection-job`.

AWS CLI

Um einen Job zur Erkennung von PII Entitäten zu beschreiben

Im folgenden `describe-pii-entities-detection-job` Beispiel werden die Eigenschaften eines asynchronen Auftrags zur Erkennung von PII-Entitäten abgerufen.

```
aws comprehend describe-pii-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "PiiEntitiesDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "example-pii-entities-job",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
    "EndTime": "2023-06-08T21:40:23.509000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
```

```

        "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribePiiEntitiesDetectionJob](#) AWS CLI

describe-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `describe-resource-policy`.

AWS CLI

Um eine an ein Modell angehängte Ressourcenrichtlinie zu beschreiben

Im folgenden `describe-resource-policy` Beispiel werden die Eigenschaften einer ressourcenbasierten Richtlinie abgerufen, die einem Modell zugeordnet ist.

```

aws comprehend describe-resource-policy \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1

```

Ausgabe:

```

{
  "ResourcePolicy": "{\\"Version\\":\\"2012-10-17\\",\\"Statement\\":[{\\"Effect\\":
\\"Allow\\",\\"Principal\\":{\\"AWS\\":\\"arn:aws:iam::444455556666:root\\"},\\"Action\\":
\\"comprehend:ImportModel\\",\\"Resource\\":\\"*\\"}]}",
  "CreationTime": "2023-06-19T18:44:26.028000+00:00",
  "LastModifiedTime": "2023-06-19T18:53:02.002000+00:00",
  "PolicyRevisionId": "baa675d069d07afaa2aa3106ae280f61"
}

```

Weitere Informationen finden Sie unter [Kopieren von benutzerdefinierten Modellen zwischen AWS Konten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [DescribeResourcePolicy AWS CLI Befehlsreferenz](#).

describe-sentiment-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-sentiment-detection-job`.

AWS CLI

Um einen Job zur Stimmungserkennung zu beschreiben

Im folgenden `describe-sentiment-detection-job` Beispiel werden die Eigenschaften eines asynchronen Stimmungserkennungsauftrags abgerufen.

```
aws comprehend describe-sentiment-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "SentimentDetectionJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "movie_review_analysis",  
    "JobStatus": "IN_PROGRESS",  
    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-  
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"  
    },  
    "LanguageCode": "en",  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-servicerole"  
  }  
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeSentimentDetectionJob](#) AWS CLI

describe-targeted-sentiment-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-targeted-sentiment-detection-job`.

AWS CLI

Um einen Job zur gezielten Stimmungserkennung zu beschreiben

Im folgenden `describe-targeted-sentiment-detection-job` Beispiel werden die Eigenschaften eines asynchronen Auftrags zur gezielten Stimmungserkennung abgerufen.

```
aws comprehend describe-targeted-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "TargetedSentimentDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "movie_review_analysis",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
  }
}
```



```
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz.
[DescribeTargetedSentimentDetectionJob](#) AWS CLI

describe-topics-detection-job

Das folgende Codebeispiel zeigt die Verwendung `describe-topics-detection-job`.

AWS CLI

Um einen Job zur Themenerkennung zu beschreiben

Im folgenden `describe-topics-detection-job` Beispiel werden die Eigenschaften eines asynchronen Themenerkennungsauftrags abgerufen.

```
aws comprehend describe-topics-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "TopicsDetectionJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "example_topics_detection",  
    "JobStatus": "IN_PROGRESS",  
    "SubmitTime": "2023-06-09T18:44:43.414000+00:00",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-  
TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"  
    },  
    "NumberOfTopics": 10,  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-examplerole"  
  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeTopicsDetectionJob](#) AWS CLI

detect-dominant-language

Das folgende Codebeispiel zeigt die Verwendung `detect-dominant-language`.

AWS CLI

Um die dominante Sprache des Eingabetextes zu erkennen

Im Folgenden wird der Eingabetext `detect-dominant-language` analysiert und die dominante Sprache identifiziert. Der Konfidenzwert des vortrainierten Modells wird ebenfalls ausgegeben.

```
aws comprehend detect-dominant-language \  
  --text "It is a beautiful day in Seattle."
```

Ausgabe:

```
{  
  "Languages": [  
    {  
      "LanguageCode": "en",  
      "Score": 0.9877256155014038  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Dominant Language](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DetectDominantLanguage](#) in der AWS CLI Befehlsreferenz.

detect-entities

Das folgende Codebeispiel zeigt die Verwendung `detect-entities`.

AWS CLI

Um benannte Entitäten im Eingabetext zu erkennen

Das folgende `detect-entities` Beispiel analysiert den Eingabetext und gibt die benannten Entitäten zurück. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-entities \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st. Based on your autopay settings, \  
we will withdraw your payment on the due date from your bank account number  
XXXXXX1111 with the routing number XXXXX0000. \  
Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to  
Alice at AnySpa@example.com."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Score": 0.9994556307792664,  
      "Type": "PERSON",  
      "Text": "Zhang Wei",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9981022477149963,  
      "Type": "PERSON",  
      "Text": "John",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.9986887574195862,  
      "Type": "ORGANIZATION",  
      "Text": "AnyCompany Financial Services, LLC",  
      "BeginOffset": 33,  
      "EndOffset": 67  
    }  
  ]  
}
```

```
  },
  {
    "Score": 0.9959119558334351,
    "Type": "OTHER",
    "Text": "1111-XXXX-1111-XXXX",
    "BeginOffset": 88,
    "EndOffset": 107
  },
  {
    "Score": 0.9708039164543152,
    "Type": "QUANTITY",
    "Text": ".53",
    "BeginOffset": 133,
    "EndOffset": 136
  },
  {
    "Score": 0.9987268447875977,
    "Type": "DATE",
    "Text": "July 31st",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9858865737915039,
    "Type": "OTHER",
    "Text": "XXXXXX1111",
    "BeginOffset": 271,
    "EndOffset": 281
  },
  {
    "Score": 0.9700471758842468,
    "Type": "OTHER",
    "Text": "XXXXX0000",
    "BeginOffset": 306,
    "EndOffset": 315
  },
  {
    "Score": 0.9591118693351746,
    "Type": "ORGANIZATION",
    "Text": "Sunshine Spa",
    "BeginOffset": 340,
    "EndOffset": 352
  },
  {
```

```

    "Score": 0.9797496795654297,
    "Type": "LOCATION",
    "Text": "123 Main St",
    "BeginOffset": 354,
    "EndOffset": 365
  },
  {
    "Score": 0.994929313659668,
    "Type": "PERSON",
    "Text": "Alice",
    "BeginOffset": 394,
    "EndOffset": 399
  },
  {
    "Score": 0.9949769377708435,
    "Type": "OTHER",
    "Text": "AnySpa@example.com",
    "BeginOffset": 403,
    "EndOffset": 418
  }
]
}

```

Weitere Informationen finden Sie unter [Entitäten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DetectEntities](#) in der AWS CLI Befehlsreferenz.

detect-key-phrases

Das folgende Codebeispiel zeigt die Verwendung `detect-key-phrases`.

AWS CLI

Um Schlüsselphrasen im Eingabetext zu erkennen

Das folgende `detect-key-phrases` Beispiel analysiert den Eingabetext und identifiziert die wichtigsten Nominalphrasen. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```

aws comprehend detect-key-phrases \
  --language-code en \
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
  credit card \

```

*account 1111-XXXX-1111-XXXX has a minimum payment of \$24.53 that is due by July 31st. Based on your autopay settings, *
*we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000. *
Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at AnySpa@example.com."

Ausgabe:

```
{
  "KeyPhrases": [
    {
      "Score": 0.8996376395225525,
      "Text": "Zhang Wei",
      "BeginOffset": 6,
      "EndOffset": 15
    },
    {
      "Score": 0.9992469549179077,
      "Text": "John",
      "BeginOffset": 22,
      "EndOffset": 26
    },
    {
      "Score": 0.988385021686554,
      "Text": "Your AnyCompany Financial Services",
      "BeginOffset": 28,
      "EndOffset": 62
    },
    {
      "Score": 0.8740853071212769,
      "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
      "BeginOffset": 64,
      "EndOffset": 107
    },
    {
      "Score": 0.9999437928199768,
      "Text": "a minimum payment",
      "BeginOffset": 112,
      "EndOffset": 129
    },
    {
      "Score": 0.9998900890350342,
```

```
    "Text": ".53",
    "BeginOffset": 133,
    "EndOffset": 136
  },
  {
    "Score": 0.9979453086853027,
    "Text": "July 31st",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9983011484146118,
    "Text": "your autopay settings",
    "BeginOffset": 172,
    "EndOffset": 193
  },
  {
    "Score": 0.9996572136878967,
    "Text": "your payment",
    "BeginOffset": 211,
    "EndOffset": 223
  },
  {
    "Score": 0.9995037317276001,
    "Text": "the due date",
    "BeginOffset": 227,
    "EndOffset": 239
  },
  {
    "Score": 0.9702621698379517,
    "Text": "your bank account number XXXXXX1111",
    "BeginOffset": 245,
    "EndOffset": 280
  },
  {
    "Score": 0.9179925918579102,
    "Text": "the routing number XXXXX0000.Customer feedback",
    "BeginOffset": 286,
    "EndOffset": 332
  },
  {
    "Score": 0.9978160858154297,
    "Text": "Sunshine Spa",
    "BeginOffset": 337,
```

```
    "EndOffset": 349
  },
  {
    "Score": 0.9706913232803345,
    "Text": "123 Main St",
    "BeginOffset": 351,
    "EndOffset": 362
  },
  {
    "Score": 0.9941995143890381,
    "Text": "comments",
    "BeginOffset": 379,
    "EndOffset": 387
  },
  {
    "Score": 0.9759287238121033,
    "Text": "Alice",
    "BeginOffset": 391,
    "EndOffset": 396
  },
  {
    "Score": 0.8376792669296265,
    "Text": "AnySpa@example.com",
    "BeginOffset": 400,
    "EndOffset": 415
  }
]
}
```

Weitere Informationen finden Sie unter [Schlüsselbegriffe](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DetectKeyPhrases](#) in der AWS CLI Befehlsreferenz.

detect-pii-entities

Das folgende Codebeispiel zeigt die Verwendung `detect-pii-entities`.

AWS CLI

Um PII-Entitäten im Eingabetext zu erkennen

Das folgende `detect-pii-entities` Beispiel analysiert den Eingabetext und identifiziert Entitäten, die personenbezogene Daten enthalten (PII). Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-pii-entities \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by  
July 31st. Based on your autopay settings, \  
  we will withdraw your payment on the due date from your bank account number  
XXXXXX1111 with the routing number XXXXX0000. \  
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to  
Alice at AnySpa@example.com."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Score": 0.9998322129249573,  
      "Type": "NAME",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9998878240585327,  
      "Type": "NAME",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.9994089603424072,  
      "Type": "CREDIT_DEBIT_NUMBER",  
      "BeginOffset": 88,  
      "EndOffset": 107  
    },  
    {  
      "Score": 0.9999760985374451,  
      "Type": "DATE_TIME",  
      "BeginOffset": 152,  
      "EndOffset": 161  
    }  
  ],  
}
```

```
{
  "Score": 0.9999449253082275,
  "Type": "BANK_ACCOUNT_NUMBER",
  "BeginOffset": 271,
  "EndOffset": 281
},
{
  "Score": 0.9999847412109375,
  "Type": "BANK_ROUTING",
  "BeginOffset": 306,
  "EndOffset": 315
},
{
  "Score": 0.999925434589386,
  "Type": "ADDRESS",
  "BeginOffset": 354,
  "EndOffset": 365
},
{
  "Score": 0.9989161491394043,
  "Type": "NAME",
  "BeginOffset": 394,
  "EndOffset": 399
},
{
  "Score": 0.9994171857833862,
  "Type": "EMAIL",
  "BeginOffset": 403,
  "EndOffset": 418
}
]
```

Weitere Informationen finden Sie unter [Persönlich identifizierbare Informationen \(PII\)](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DetectPiiEntities](#) in der AWS CLI Befehlsreferenz.

detect-sentiment

Das folgende Codebeispiel zeigt die Verwendung `detect-sentiment`.

AWS CLI

Um die Stimmung eines Eingabetextes zu erkennen

Das folgende `detect-sentiment` Beispiel analysiert den Eingabetext und gibt einen Rückschluss auf die vorherrschende Stimmung (POSITIVE, NEUTRALMIXED, oder) zurück. NEGATIVE

```
aws comprehend detect-sentiment \  
  --language-code en \  
  --text "It is a beautiful day in Seattle"
```

Ausgabe:

```
{  
  "Sentiment": "POSITIVE",  
  "SentimentScore": {  
    "Positive": 0.9976957440376282,  
    "Negative": 9.653854067437351e-05,  
    "Neutral": 0.002169104292988777,  
    "Mixed": 3.857641786453314e-05  
  }  
}
```

Weitere Informationen finden Sie unter [Sentiment](#) im Amazon Comprehend Developer Guide

- API-Einheiten finden Sie [DetectSentiment](#) in AWS CLI der Befehlsreferenz.

detect-syntax

Das folgende Codebeispiel zeigt die Verwendung `detect-syntax`.

AWS CLI

Um die Wortarten in einem Eingabetext zu erkennen

Im folgenden `detect-syntax` Beispiel wird die Syntax des Eingabetextes analysiert und die verschiedenen Wortarten zurückgegeben. Der Konfidenzwert des vortrainierten Modells wird ebenfalls für jede Vorhersage ausgegeben.

```
aws comprehend detect-syntax \  
  --text "It is a beautiful day in Seattle"
```

```
--language-code en \  
--text "It is a beautiful day in Seattle."
```

Ausgabe:

```
{  
  "SyntaxTokens": [  
    {  
      "TokenId": 1,  
      "Text": "It",  
      "BeginOffset": 0,  
      "EndOffset": 2,  
      "PartOfSpeech": {  
        "Tag": "PRON",  
        "Score": 0.9999740719795227  
      }  
    },  
    {  
      "TokenId": 2,  
      "Text": "is",  
      "BeginOffset": 3,  
      "EndOffset": 5,  
      "PartOfSpeech": {  
        "Tag": "VERB",  
        "Score": 0.999901294708252  
      }  
    },  
    {  
      "TokenId": 3,  
      "Text": "a",  
      "BeginOffset": 6,  
      "EndOffset": 7,  
      "PartOfSpeech": {  
        "Tag": "DET",  
        "Score": 0.9999938607215881  
      }  
    },  
    {  
      "TokenId": 4,  
      "Text": "beautiful",  
      "BeginOffset": 8,  
      "EndOffset": 17,  
      "PartOfSpeech": {
```

```
        "Tag": "ADJ",
        "Score": 0.9987351894378662
    }
},
{
    "TokenId": 5,
    "Text": "day",
    "BeginOffset": 18,
    "EndOffset": 21,
    "PartOfSpeech": {
        "Tag": "NOUN",
        "Score": 0.9999796748161316
    }
},
{
    "TokenId": 6,
    "Text": "in",
    "BeginOffset": 22,
    "EndOffset": 24,
    "PartOfSpeech": {
        "Tag": "ADP",
        "Score": 0.9998047947883606
    }
},
{
    "TokenId": 7,
    "Text": "Seattle",
    "BeginOffset": 25,
    "EndOffset": 32,
    "PartOfSpeech": {
        "Tag": "PROPN",
        "Score": 0.9940530061721802
    }
}
]
}
```

Weitere Informationen finden Sie unter [Syntaxanalyse](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [DetectSyntax](#) in der AWS CLI Befehlsreferenz.

detect-targeted-sentiment

Das folgende Codebeispiel zeigt die Verwendung `detect-targeted-sentiment`.

AWS CLI

Um die gezielte Stimmung benannter Entitäten in einem Eingabetext zu erkennen

Im folgenden `detect-targeted-sentiment` Beispiel wird der Eingabetext analysiert und die benannten Entitäten sowie die Zielstimmung, die jeder Entität zugeordnet ist, zurückgegeben. Der Konfidenzwert des vortrainierten Modells für jede Vorhersage wird ebenfalls ausgegeben.

```
aws comprehend detect-targeted-sentiment \  
  --language-code en \  
  --text "I do not enjoy January because it is too cold but August is the perfect temperature"
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "DescriptiveMentionIndex": [  
        0  
      ],  
      "Mentions": [  
        {  
          "Score": 0.9999979734420776,  
          "GroupScore": 1.0,  
          "Text": "I",  
          "Type": "PERSON",  
          "MentionSentiment": {  
            "Sentiment": "NEUTRAL",  
            "SentimentScore": {  
              "Positive": 0.0,  
              "Negative": 0.0,  
              "Neutral": 1.0,  
              "Mixed": 0.0  
            }  
          },  
          "BeginOffset": 0,  
          "EndOffset": 1  
        }  
      ]  
    }  
  ]  
}
```

```
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9638869762420654,
      "GroupScore": 1.0,
      "Text": "January",
      "Type": "DATE",
      "MentionSentiment": {
        "Sentiment": "NEGATIVE",
        "SentimentScore": {
          "Positive": 0.0031610000878572464,
          "Negative": 0.9967250227928162,
          "Neutral": 0.00011100000119768083,
          "Mixed": 1.9999999949504854e-06
        }
      }
    },
    {
      "BeginOffset": 15,
      "EndOffset": 22
    }
  ]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9664419889450073,
      "GroupScore": 1.0,
      "Text": "August",
      "Type": "DATE",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Positive": 0.9999549984931946,
          "Negative": 3.99999989900971e-06,
          "Neutral": 4.099999932805076e-05,
          "Mixed": 0.0
        }
      }
    }
  ]
}
```

```

    }
  },
  "BeginOffset": 50,
  "EndOffset": 56
}
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9803199768066406,
      "GroupScore": 1.0,
      "Text": "temperature",
      "Type": "ATTRIBUTE",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Positive": 1.0,
          "Negative": 0.0,
          "Neutral": 0.0,
          "Mixed": 0.0
        }
      }
    },
    {
      "BeginOffset": 77,
      "EndOffset": 88
    }
  ]
}
]
}
}

```

Weitere Informationen finden Sie unter [Targeted Sentiment](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [DetectTargetedSentiment](#).AWS CLI

import-model

Das folgende Codebeispiel zeigt die Verwendung `import-model`.

AWS CLI

Um ein Modell zu importieren

Im folgenden `import-model` Beispiel wird ein Modell aus einem anderen AWS Konto importiert. Das Dokumentenklassifizierungsmodell im Konto 444455556666 verfügt über eine ressourcenbasierte Richtlinie, die es dem Konto ermöglicht, das Modell 111122223333 zu importieren.

```
aws comprehend import-model \  
  --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-  
  classifier/example-classifier
```

Ausgabe:

```
{  
  "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
  example-classifier"  
}
```

Weitere Informationen finden Sie unter [Kopieren von benutzerdefinierten Modellen zwischen AWS Konten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [ImportModel AWS CLI](#) Befehlsreferenz.

list-datasets

Das folgende Codebeispiel zeigt die Verwendung `list-datasets`.

AWS CLI

Um alle Schwungrad Datensätze aufzulisten

Das folgende `list-datasets` Beispiel listet alle Datensätze auf, die einem Schwungrad zugeordnet sind.

```
aws comprehend list-datasets \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
  entity
```

Ausgabe:

```
{
  "DatasetPropertiesList": [
    {
      "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
flywheel-entity/dataset/example-dataset-1",
      "DatasetName": "example-dataset-1",
      "DatasetType": "TRAIN",
      "DatasetS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/
schemaVersion=1/20230616T200543Z/datasets/example-dataset-1/20230616T203710Z/",
      "Status": "CREATING",
      "CreationTime": "2023-06-16T20:37:10.400000+00:00"
    },
    {
      "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
flywheel-entity/dataset/example-dataset-2",
      "DatasetName": "example-dataset-2",
      "DatasetType": "TRAIN",
      "DatasetS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/
schemaVersion=1/20230616T200543Z/datasets/example-dataset-2/20230616T200607Z/",
      "Description": "TRAIN Dataset created by Flywheel creation.",
      "Status": "COMPLETED",
      "NumberOfDocuments": 5572,
      "CreationTime": "2023-06-16T20:06:07.722000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Flywheel Overview](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListDatasets](#) in AWS CLI der Befehlsreferenz.

list-document-classification-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-document-classification-jobs`.

AWS CLI

Um alle Jobs zur Dokumentenklassifizierung aufzulisten

Das folgende `list-document-classification-jobs` Beispiel listet alle Aufträge zur Dokumentenklassifizierung auf.

aws comprehend list-document-classification-jobs

Ausgabe:

```
{
  "DocumentClassificationJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "exampleclassificationjob",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
      "EndTime": "2023-06-14T17:15:58.582000+00:00",
      "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/1234567890101-CLN-e758dd56b824aa717ceab551f11749fb/output/output.tar.gz"
      },
      "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE2",
      "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE2",
      "JobName": "exampleclassificationjob2",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-14T17:22:39.829000+00:00",
      "EndTime": "2023-06-14T17:28:46.107000+00:00",
      "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
```

```
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/1234567890101-CLN-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListDocumentClassificationJobs](#) in der AWS CLI Befehlsreferenz.

list-document-classifier-summaries

Das folgende Codebeispiel zeigt die Verwendung `list-document-classifier-summaries`.

AWS CLI

Um die Zusammenfassungen aller erstellten Dokumentenklassifikatoren aufzulisten

Das folgende `list-document-classifier-summaries` Beispiel listet alle erstellten Zusammenfassungen von Dokumentenklassifikatoren auf.

```
aws comprehend list-document-classifier-summaries
```

Ausgabe:

```
{
  "DocumentClassifierSummariesList": [
    {
      "DocumentClassifierName": "example-classifier-1",
      "NumberOfVersions": 1,
      "LatestVersionCreatedAt": "2023-06-13T22:07:59.825000+00:00",
      "LatestVersionName": "1",
      "LatestVersionStatus": "TRAINED"
    },
    {
      "DocumentClassifierName": "example-classifier-2",
      "NumberOfVersions": 2,
      "LatestVersionCreatedAt": "2023-06-13T21:54:59.589000+00:00",
      "LatestVersionName": "2",

```

```

        "LatestVersionStatus": "TRAINED"
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [ListDocumentClassifierSummaries AWS CLI Befehlsreferenz](#).

list-document-classifiers

Das folgende Codebeispiel zeigt die Verwendung `list-document-classifiers`.

AWS CLI

Zur Liste aller Dokumentenklassifikatoren

Das folgende `list-document-classifiers` Beispiel listet alle trainierten und trainierten Dokumentenklassifizierungsmodelle auf.

```
aws comprehend list-document-classifiers
```

Ausgabe:

```

{
  "DocumentClassifierPropertiesList": [
    {
      "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier1",
      "LanguageCode": "en",
      "Status": "TRAINED",
      "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
      "EndTime": "2023-06-13T19:42:31.752000+00:00",
      "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
      "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
      "InputDataConfig": {
        "DataFormat": "COMPREHEND_CSV",
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
      },
      "OutputDataConfig": {},
      "ClassifierMetadata": {
        "NumberOfLabels": 3,

```

```

        "NumberOfTrainedDocuments": 5016,
        "NumberOfTestDocuments": 557,
        "EvaluationMetrics": {
            "Accuracy": 0.9856,
            "Precision": 0.9919,
            "Recall": 0.9459,
            "F1Score": 0.9673,
            "MicroPrecision": 0.9856,
            "MicroRecall": 0.9856,
            "MicroF1Score": 0.9856,
            "HammingLoss": 0.0144
        }
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
    "Mode": "MULTI_CLASS"
},
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier2",
    "LanguageCode": "en",
    "Status": "TRAINING",
    "SubmitTime": "2023-06-13T21:20:28.690000+00:00",
    "InputDataConfig": {
        "DataFormat": "COMPREHEND_CSV",
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
    },
    "OutputDataConfig": {},
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
    "Mode": "MULTI_CLASS"
}
]
}

```

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [ListDocumentClassifiers AWS CLI Befehlsreferenz](#).

list-dominant-language-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-dominant-language-detection-jobs`.

AWS CLI

Um alle Jobs zur Erkennung dominanter Sprachen aufzulisten

Das folgende `list-dominant-language-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen Aufträge zur Erkennung dominanter Sprache auf.

```
aws comprehend list-dominant-language-detection-jobs
```

Ausgabe:

```
{
  "DominantLanguageDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "languageanalysis1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
      "EndTime": "2023-06-09T18:18:45.498000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
      },
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "languageanalysis2",
      "JobStatus": "STOPPED",
      "SubmitTime": "2023-06-09T18:16:33.690000+00:00",
      "EndTime": "2023-06-09T18:24:40.608000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
```

```

        "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListDominantLanguageDetectionJobsAWS CLI](#)

list-endpoints

Das folgende Codebeispiel zeigt die Verwendung `list-endpoints`.

AWS CLI

Zur Liste aller Endpunkte

Das folgende `list-endpoints` Beispiel listet alle aktiven modellspezifischen Endpunkte auf.

```
aws comprehend list-endpoints
```

Ausgabe:

```

{
  "EndpointPropertiesList": [
    {
      "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/ExampleClassifierEndpoint",
      "Status": "IN_SERVICE",
      "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
      "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier1",
    }
  ]
}

```



```

        "DesiredInferenceUnits": 1,
        "CurrentInferenceUnits": 1,
        "CreationTime": "2023-06-13T20:32:54.526000+00:00",
        "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
    },
    {
        "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/ExampleClassifierEndpoint2",
        "Status": "IN_SERVICE",
        "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2",
        "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2",
        "DesiredInferenceUnits": 1,
        "CurrentInferenceUnits": 1,
        "CreationTime": "2023-06-13T20:32:54.526000+00:00",
        "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [ListEndpoints](#) AWS CLI

list-entities-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-entities-detection-jobs`.

AWS CLI

Um alle Jobs zur Erkennung von Entitäten aufzulisten

Das folgende `list-entities-detection-jobs` Beispiel listet alle Aufträge zur Erkennung asynchroner Entitäten auf.

```
aws comprehend list-entities-detection-jobs
```

Ausgabe:

```
{
  "EntitiesDetectionJobPropertiesList": [
```

```
{
  "JobId": "468af39c28ab45b83eb0c4ab9EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/468af39c28ab45b83eb0c4ab9EXAMPLE",
  "JobName": "example-entities-detection",
  "JobStatus": "COMPLETED",
  "SubmitTime": "2023-06-08T20:57:46.476000+00:00",
  "EndTime": "2023-06-08T21:05:53.718000+00:00",
  "InputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
    "InputFormat": "ONE_DOC_PER_LINE"
  },
  "OutputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-NER-468af39c28ab45b83eb0c4ab9EXAMPLE/output/output.tar.gz"
  },
  "LanguageCode": "en",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
},
{
  "JobId": "809691caeaab0e71406f80a28EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/809691caeaab0e71406f80a28EXAMPLE",
  "JobName": "example-entities-detection-2",
  "JobStatus": "COMPLETED",
  "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
  "EndTime": "2023-06-08T21:40:23.509000+00:00",
  "InputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
    "InputFormat": "ONE_DOC_PER_LINE"
  },
  "OutputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-NER-809691caeaab0e71406f80a28EXAMPLE/output/output.tar.gz"
  },
  "LanguageCode": "en",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
},
{
  "JobId": "e00597c36b448b91d70dea165EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/e00597c36b448b91d70dea165EXAMPLE",
```

```

    "JobName": "example-entities-detection-3",
    "JobStatus": "STOPPED",
    "SubmitTime": "2023-06-08T22:19:28.528000+00:00",
    "EndTime": "2023-06-08T22:27:33.991000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-NER-e00597c36b448b91d70dea165EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

Weitere Informationen finden Sie unter [Entitäten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListEntitiesDetectionJobs](#) in der AWS CLI Befehlsreferenz.

list-entity-recognizer-summaries

Das folgende Codebeispiel zeigt die Verwendung `list-entity-recognizer-summaries`.

AWS CLI

Um eine Liste der Zusammenfassungen für alle erstellten Entitätserkennungen aufzulisten

Das folgende `list-entity-recognizer-summaries` Beispiel listet alle Zusammenfassungen von Entity Recognizern auf.

```
aws comprehend list-entity-recognizer-summaries
```

Ausgabe:

```

{
  "EntityRecognizerSummariesList": [
    {
      "RecognizerName": "entity-recognizer-3",
      "NumberOfVersions": 2,

```

```
    "LatestVersionCreatedAt": "2023-06-15T23:15:07.621000+00:00",
    "LatestVersionName": "2",
    "LatestVersionStatus": "STOP_REQUESTED"
  },
  {
    "RecognizerName": "entity-recognizer-2",
    "NumberOfVersions": 1,
    "LatestVersionCreatedAt": "2023-06-14T22:55:27.805000+00:00",
    "LatestVersionName": "2"
    "LatestVersionStatus": "TRAINED"
  },
  {
    "RecognizerName": "entity-recognizer-1",
    "NumberOfVersions": 1,
    "LatestVersionCreatedAt": "2023-06-14T20:44:59.631000+00:00",
    "LatestVersionName": "1",
    "LatestVersionStatus": "TRAINED"
  }
]
}
```

Weitere Informationen finden Sie unter [Custom Entity Recognition](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [ListEntityRecognizerSummaries AWS CLI Befehlsreferenz](#).

list-entity-recognizers

Das folgende Codebeispiel zeigt die Verwendung `list-entity-recognizers`.

AWS CLI

Zur Liste aller benutzerdefinierten Entitätserkennungen

Das folgende `list-entity-recognizers` Beispiel listet alle erstellten benutzerdefinierten Entitätserkennungen auf.

```
aws comprehend list-entity-recognizers
```

Ausgabe:

```
{
  "EntityRecognizerPropertiesList": [
```

```
{
  "EntityRecognizerArn": "arn:aws:comprehend:us-
west-2:111122223333:entity-recognizer/EntityRecognizer/version/1",
  "LanguageCode": "en",
  "Status": "TRAINED",
  "SubmitTime": "2023-06-14T20:44:59.631000+00:00",
  "EndTime": "2023-06-14T20:59:19.532000+00:00",
  "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",
  "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",
  "InputDataConfig": {
    "DataFormat": "COMPREHEND_CSV",
    "EntityTypes": [
      {
        "Type": "BUSINESS"
      }
    ],
    "Documents": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/dataset/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "EntityList": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/entity.csv"
    }
  },
  "RecognizerMetadata": {
    "NumberOfTrainedDocuments": 1814,
    "NumberOfTestDocuments": 486,
    "EvaluationMetrics": {
      "Precision": 100.0,
      "Recall": 100.0,
      "F1Score": 100.0
    },
    "EntityTypes": [
      {
        "Type": "BUSINESS",
        "EvaluationMetrics": {
          "Precision": 100.0,
          "Recall": 100.0,
          "F1Score": 100.0
        }
      },
      "NumberOfTrainMentions": 1520
    ]
  }
},
```

```
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole",
    "VersionName": "1"
  },
  {
    "EntityRecognizerArn": "arn:aws:comprehend:us-
west-2:111122223333:entity-recognizer/entityrecognizer3",
    "LanguageCode": "en",
    "Status": "TRAINED",
    "SubmitTime": "2023-06-14T22:57:51.056000+00:00",
    "EndTime": "2023-06-14T23:14:13.894000+00:00",
    "TrainingStartTime": "2023-06-14T23:01:33.984000+00:00",
    "TrainingEndTime": "2023-06-14T23:13:02.984000+00:00",
    "InputDataConfig": {
      "DataFormat": "COMPREHEND_CSV",
      "EntityTypes": [
        {
          "Type": "DEVICE"
        }
      ],
      "Documents": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/raw_txt.csv",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "EntityList": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/entity_list.csv"
      }
    },
    "RecognizerMetadata": {
      "NumberOfTrainedDocuments": 4616,
      "NumberOfTestDocuments": 3489,
      "EvaluationMetrics": {
        "Precision": 98.54227405247813,
        "Recall": 100.0,
        "F1Score": 99.26578560939794
      }
    },
    "EntityTypes": [
      {
        "Type": "DEVICE",
        "EvaluationMetrics": {
          "Precision": 98.54227405247813,
          "Recall": 100.0,
          "F1Score": 99.26578560939794
        }
      }
    ],
  },
}
```

```

        "NumberOfTrainMentions": 2764
      }
    ]
  },
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
}
]
}

```

Weitere Informationen finden Sie unter [Custom Entity Recognition](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListEntityRecognizers](#) in der AWS CLI Befehlsreferenz.

list-events-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-events-detection-jobs`.

AWS CLI

Um alle Jobs zur Erkennung von Ereignissen aufzulisten

Im folgenden `list-events-detection-jobs` Beispiel werden alle asynchronen Aufgaben zur Erkennung von Ereignissen aufgeführt.

```
aws comprehend list-events-detection-jobs
```

Ausgabe:

```

{
  "EventsDetectionJobPropertiesList": [
    {
      "JobId": "aa9593f9203e84f3ef032ce18EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/aa9593f9203e84f3ef032ce18EXAMPLE",
      "JobName": "events_job_1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-12T19:14:57.751000+00:00",
      "EndTime": "2023-06-12T19:21:04.962000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/EventsData/",
        "InputFormat": "ONE_DOC_PER_LINE"
      }
    }
  ]
}

```

```

    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/1111222233333-EVENTS-aa9593f9203e84f3ef032ce18EXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "CORPORATE_MERGER",
      "INVESTMENT_GENERAL"
    ]
  },
  {
    "JobId": "4a990a2f7e82adfca6e171135EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/4a990a2f7e82adfca6e171135EXAMPLE",
    "JobName": "events_job_2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-12T19:55:43.702000+00:00",
    "EndTime": "2023-06-12T20:03:49.893000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/EventsData/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/1111222233333-EVENTS-4a990a2f7e82adfca6e171135EXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "CORPORATE_MERGER",
      "INVESTMENT_GENERAL"
    ]
  }
]

```



```
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListEventsDetectionJobs](#) AWS CLI

list-flywheel-iteration-history

Das folgende Codebeispiel zeigt die Verwendung `list-flywheel-iteration-history`.

AWS CLI

Um den gesamten Verlauf der Flywheel-Iterationen aufzulisten

Das folgende `list-flywheel-iteration-history` Beispiel listet alle Iterationen eines Schwungrades auf.

```
aws comprehend list-flywheel-iteration-history
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

Ausgabe:

```
{
  "FlywheelIterationPropertiesList": [
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",
      "FlywheelIterationId": "20230619EXAMPLE",
      "CreationTime": "2023-06-19T04:00:32.594000+00:00",
      "EndTime": "2023-06-19T04:00:49.248000+00:00",
      "Status": "COMPLETED",
      "Message": "FULL_ITERATION: Flywheel iteration performed all functions successfully.",
      "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/1",
      "EvaluatedModelMetrics": {
        "AverageF1Score": 0.7742663922375772,
        "AverageF1Score": 0.9876464664646313,
        "AveragePrecision": 0.9800000253081214,
        "AverageRecall": 0.9445600253081214,
      }
    }
  ]
}
```

```

        "AverageAccuracy": 0.9997281665190434
    },
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-flywheel/
schemaVersion=1/20230619TEXAMPLE/evaluation/20230619TEXAMPLE/"
  },
  {
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
    "FlywheelIterationId": "20230616TEXAMPLE",
    "CreationTime": "2023-06-16T21:10:26.385000+00:00",
    "EndTime": "2023-06-16T23:33:16.827000+00:00",
    "Status": "COMPLETED",
    "Message": "FULL_ITERATION: Flywheel iteration performed all functions
successfully.",
    "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/spamvshamclassify/version/1",
    "EvaluatedModelMetrics": {
      "AverageF1Score": 0.7742663922375772,
      "AverageF1Score": 0.9767700253081214,
      "AveragePrecision": 0.9767700253081214,
      "AverageRecall": 0.9767700253081214,
      "AverageAccuracy": 0.9858281665190434
    },
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-
flywheel-2/schemaVersion=1/20230616TEXAMPLE/evaluation/20230616TEXAMPLE/"
  }
]
}

```

Weitere Informationen finden Sie in der [Übersicht über Flywheel](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListFlywheelIterationHistory](#) in AWS CLI der Befehlsreferenz.

list-flywheels

Das folgende Codebeispiel zeigt die Verwendung `list-flywheels`.

AWS CLI

Um alle Schwungräder aufzulisten

Das folgende `list-flywheels` Beispiel listet alle erstellten Schwungräder auf.

aws comprehend list-flywheels

Ausgabe:

```
{
  "FlywheelSummaryList": [
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-1",
      "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier/version/1",
      "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/example-flywheel-1/
schemaVersion=1/20230616T200543Z/",
      "Status": "ACTIVE",
      "ModelType": "DOCUMENT_CLASSIFIER",
      "CreationTime": "2023-06-16T20:05:43.242000+00:00",
      "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
      "LatestFlywheelIteration": "20230619T040032Z"
    },
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
      "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/exampleclassifier2/version/1",
      "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/example-flywheel-2/
schemaVersion=1/20220616T200543Z/",
      "Status": "ACTIVE",
      "ModelType": "DOCUMENT_CLASSIFIER",
      "CreationTime": "2022-06-16T20:05:43.242000+00:00",
      "LastModifiedTime": "2022-06-19T04:00:43.027000+00:00",
      "LatestFlywheelIteration": "20220619T040032Z"
    }
  ]
}
```

Weitere Informationen finden Sie in der [Übersicht über Flywheel](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListFlywheels](#) in AWS CLI der Befehlsreferenz.

list-key-phrases-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-key-phrases-detection-jobs`.

AWS CLI

Um alle Jobs zur Erkennung von Schlüsselphrasen aufzulisten

Das folgende `list-key-phrases-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen Aufträge zur Erkennung von Schlüsselwörtern auf.

```
aws comprehend list-key-phrases-detection-jobs
```

Ausgabe:

```
{
  "KeyPhrasesDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "keyphrasesanalysis1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-08T22:31:43.767000+00:00",
      "EndTime": "2023-06-08T22:39:52.565000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/AsyncBatchJobs/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-KP-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a33EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a33EXAMPLE",
      "JobName": "keyphrasesanalysis2",
      "JobStatus": "STOPPED",
    }
  ]
}
```

```

    "SubmitTime": "2023-06-08T22:57:52.154000+00:00",
    "EndTime": "2023-06-08T23:05:48.385000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-KP-123456abcdeb0e11022f22a33EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a44EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a44EXAMPLE",
    "JobName": "keyphrasesanalysis3",
    "JobStatus": "FAILED",
    "Message": "NO_READ_ACCESS_TO_INPUT: The provided data access role does
not have proper access to the input data.",
    "SubmitTime": "2023-06-09T16:47:04.029000+00:00",
    "EndTime": "2023-06-09T16:47:18.413000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-KP-123456abcdeb0e11022f22a44EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListKeyPhrasesDetectionJobs](#) AWS CLI

list-pii-entities-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-pii-entities-detection-jobs`.

AWS CLI

Um alle Jobs zur Erkennung von PII-Entitäten aufzulisten

Das folgende `list-pii-entities-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen PII-Erkennungsaufträge auf.

```
aws comprehend list-pii-entities-detection-jobs
```

Ausgabe:

```
{
  "PiiEntitiesDetectionJobPropertiesList": [
    {
      "JobId": "6f9db0c42d0c810e814670ee4EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/6f9db0c42d0c810e814670ee4EXAMPLE",
      "JobName": "example-pii-detection-job",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-09T21:02:46.241000+00:00",
      "EndTime": "2023-06-09T21:12:52.602000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/111122223333-
PII-6f9db0c42d0c810e814670ee4EXAMPLE/output/"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
      "Mode": "ONLY_OFFSETS"
    },
    {
      "JobId": "d927562638cfa739331a99b3cEXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/d927562638cfa739331a99b3cEXAMPLE",
      "JobName": "example-pii-detection-job-2",
```

```

    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T21:20:58.211000+00:00",
    "EndTime": "2023-06-09T21:31:06.027000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-PII-d927562638cfa739331a99b3cEXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "Mode": "ONLY_OFFSETS"
  }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListPiiEntitiesDetectionJobs](#) AWS CLI

list-sentiment-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-sentiment-detection-jobs`.

AWS CLI

Um alle Jobs zur Stimmungserkennung aufzulisten

Das folgende `list-sentiment-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen Stimmungserkennungsaufträge auf.

```
aws comprehend list-sentiment-detection-jobs
```

Ausgabe:

```

{
  "SentimentDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",

```

```

    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "example-sentiment-detection-job",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
    "EndTime": "2023-06-09T22:52:27.416000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE2",
    "JobName": "example-sentiment-detection-job-2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
    "EndTime": "2023-06-09T23:26:00.168000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData2",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE2/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListSentimentDetectionJobs](#) AWS CLI

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für Ressourcen aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für eine Amazon Comprehend Comprehend-Ressource auf.

```
aws comprehend list-tags-for-resource \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
  example-classifier/version/1
```

Ausgabe:

```
{
  "ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
  example-classifier/version/1",
  "Tags": [
    {
      "Key": "Department",
      "Value": "Finance"
    },
    {
      "Key": "location",
      "Value": "Seattle"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging Your Resources](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in AWS CLI der Befehlsreferenz.

list-targeted-sentiment-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-targeted-sentiment-detection-jobs`.

AWS CLI

Um alle Jobs zur gezielten Stimmungserkennung aufzulisten

Das folgende `list-targeted-sentiment-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen Aufträge zur gezielten Stimmungserkennung auf.

```
aws comprehend list-targeted-sentiment-detection-jobs
```

Ausgabe:

```
{
  "TargetedSentimentDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "example-targeted-sentiment-detection-job",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
      "EndTime": "2023-06-09T22:52:27.416000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-I0role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE2",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE2",
      "JobName": "example-targeted-sentiment-detection-job-2",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
      "EndTime": "2023-06-09T23:26:00.168000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData2",
```

```

        "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListTargetedSentimentDetectionJobs](#) AWS CLI

list-topics-detection-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-topics-detection-jobs`.

AWS CLI

Um alle Jobs zur Themenerkennung aufzulisten

Das folgende `list-topics-detection-jobs` Beispiel listet alle laufenden und abgeschlossenen asynchronen Themenerkennungsaufträge auf.

```
aws comprehend list-topics-detection-jobs
```

Ausgabe:

```

{
  "TopicsDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "topic-analysis-1"
      "JobStatus": "IN_PROGRESS",
      "SubmitTime": "2023-06-09T18:40:35.384000+00:00",

```

```
    "EndTime": "2023-06-09T18:46:41.936000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE/output/output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE2",
    "JobName": "topic-analysis-2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
    "EndTime": "2023-06-09T18:50:50.872000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE3",
    "JobName": "topic-analysis-2",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T18:50:56.737000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
  },
```

```

    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE3/output/output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [ListTopicsDetectionJobs](#) AWS CLI

put-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `put-resource-policy`.

AWS CLI

Um eine ressourcenbasierte Richtlinie anzuhängen

Im folgenden `put-resource-policy` Beispiel wird eine ressourcenbasierte Richtlinie an ein Modell angehängt, sodass sie von einem anderen Konto importiert werden kann. AWS Die Richtlinie ist an das Modell im Konto angehängt 111122223333 und ermöglicht den 444455556666 Import des Modells durch das Konto.

```

aws comprehend put-resource-policy \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
  --resource-policy '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Action":"comprehend:ImportModel","Resource":"*","Principal":
{"AWS":["arn:aws:iam::444455556666:root"]}]}]'

```

Ausgabe:

```

{
  "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"
}

```

Weitere Informationen finden Sie unter [Kopieren von benutzerdefinierten Modellen zwischen AWS Konten](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [PutResourcePolicy AWS CLI Befehlsreferenz](#).

start-document-classification-job

Das folgende Codebeispiel zeigt die Verwendung `start-document-classification-job`.

AWS CLI

Um den Job zur Dokumentenklassifizierung zu starten

Im folgenden `start-document-classification-job` Beispiel wird ein Auftrag zur Dokumentenklassifizierung mit einem benutzerdefinierten Modell für alle Dateien an der durch das `--input-data-config` Tag angegebenen Adresse gestartet. In diesem Beispiel enthält der S3-Eingabe-Bucket `SampleSMStext1.txt`, `SampleSMStext2.txt`, und `SampleSMStext3.txt`. Das Modell wurde zuvor anhand von Dokumentklassifizierungen von Spam- und Nicht-Spam-Nachrichten oder „betrügerischen“ Nachrichten trainiert. SMS Wenn der Job abgeschlossen ist, `output.tar.gz` wird er an der durch das `--output-data-config` Tag angegebenen Stelle abgelegt. `output.tar.gz` enthält `predictions.jsonl`, in dem die Klassifikation der einzelnen Dokumente aufgeführt ist. Die Json-Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```
aws comprehend start-document-classification-job \  
  --job-name exampleclassificationjob \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET-INPUT/jobdata/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/mymodel/version/12
```

Inhalt von `SampleSMStext1.txt`:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Inhalt von `SampleSMStext2.txt`:

```
"Hi, when do you want me to pick you up from practice?"
```

Inhalt von `SampleSMStext3.txt`:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Ausgabe:

```
{
  "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
job/e758dd56b824aa717ceab551fEXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von `predictions.jsonl`:

```
{"File": "SampleSMSText1.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
0.9999}, {"Name": "ham", "Score": 0.0001}]}
{"File": "SampleSMStext2.txt", "Line": "0", "Classes": [{"Name": "ham", "Score":
0.9994}, {"Name": "spam", "Score": 0.0006}]}
{"File": "SampleSMSText3.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
0.9999}, {"Name": "ham", "Score": 0.0001}]}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Klassifizierung](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [StartDocumentClassificationJob](#) in der AWS CLI Befehlsreferenz.

start-dominant-language-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-dominant-language-detection-job`.

AWS CLI

Um einen asynchronen Spracherkennungsauftrag zu starten

Im folgenden `start-dominant-language-detection-job` Beispiel wird ein asynchroner Spracherkennungsauftrag für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Der S3-Bucket in diesem Beispiel enthält `Sampletext1.txt`. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Ort platziert. Der Ordner enthält

output.txt die dominante Sprache der einzelnen Textdateien sowie den Konfidenzwert des vortrainierten Modells für jede Vorhersage.

```
aws comprehend start-dominant-language-detection-job \
  --job-name example_language_analysis_job \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
  --language-code en
```

Inhalt von Sampletext1.txt:

```
"Physics is the natural science that involves the study of matter and its motion and
behavior through space and time, along with related concepts such as energy and
force."
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von output.txt:

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score":
0.9913753867149353}], "Line": 0}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StartDominantLanguageDetectionJobAWS CLI](#)

start-entities-detection-job

Das folgende Codebeispiel zeigt die Verwendung start-entities-detection-job.

AWS CLI

Beispiel 1: Um einen Standardauftrag zur Erkennung von Entitäten mit dem vortrainierten Modell zu starten

Im folgenden `start-entities-detection-job` Beispiel wird ein asynchroner Auftrag zur Erkennung von Entitäten für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Der S3-Bucket in diesem Beispiel enthält `Sampletext1.txt`, `Sampletext2.txt`, und `Sampletext3.txt`. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Ort platziert. Der Ordner enthält eine `output.txt` Liste aller benannten Entitäten, die in jeder Textdatei erkannt wurden, sowie den Konfidenzwert des vortrainierten Modells für jede Vorhersage. Die Json-Ausgabe wird in einer Zeile pro Eingabedatei gedruckt, hier jedoch aus Gründen der Lesbarkeit formatiert.

```
aws comprehend start-entities-detection-job \  
  --job-name entitiestest \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --language-code en
```

Inhalt von `Sampletext1.txt`:

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st."
```

Inhalt von `Sampletext2.txt`:

```
"Dear Max, based on your autopay settings for your account example1.org account, we  
will withdraw your payment on the due date from your bank account number XXXXXX1111  
with the routing number XXXXX0000. "
```

Inhalt von `Sampletext3.txt`:

```
"Jane, please submit any customer feedback from this weekend to AnySpa, 123 Main St,  
Anywhere and send comments to Alice at AnySpa@example.com."
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von output.txt mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
  "Entities": [
    {
      "BeginOffset": 6,
      "EndOffset": 15,
      "Score": 0.9994006636420306,
      "Text": "Zhang Wei",
      "Type": "PERSON"
    },
    {
      "BeginOffset": 22,
      "EndOffset": 26,
      "Score": 0.9976647915128143,
      "Text": "John",
      "Type": "PERSON"
    },
    {
      "BeginOffset": 33,
      "EndOffset": 67,
      "Score": 0.9984608700836206,
      "Text": "AnyCompany Financial Services, LLC",
      "Type": "ORGANIZATION"
    },
    {
      "BeginOffset": 88,
      "EndOffset": 107,
      "Score": 0.9868521019555556,
      "Text": "1111-XXXX-1111-XXXX",
      "Type": "OTHER"
    },
    {
      "BeginOffset": 133,
```

```
"EndOffset": 139,
"Score": 0.998242565709204,
"Text": "$24.53",
"Type": "QUANTITY"
},
{
  "BeginOffset": 155,
  "EndOffset": 164,
  "Score": 0.9993039263159287,
  "Text": "July 31st",
  "Type": "DATE"
}
],
"File": "SampleText1.txt",
"Line": 0
}
{
  "Entities": [
    {
      "BeginOffset": 5,
      "EndOffset": 8,
      "Score": 0.9866232147545232,
      "Text": "Max",
      "Type": "PERSON"
    },
    {
      "BeginOffset": 156,
      "EndOffset": 166,
      "Score": 0.9797723450933329,
      "Text": "XXXXXX1111",
      "Type": "OTHER"
    },
    {
      "BeginOffset": 191,
      "EndOffset": 200,
      "Score": 0.9247838572396843,
      "Text": "XXXXX0000",
      "Type": "OTHER"
    }
  ],
  "File": "SampleText2.txt",
  "Line": 0
}
{
```

```
"Entities": [  
  {  
    "Score": 0.9990532994270325,  
    "Type": "PERSON",  
    "Text": "Jane",  
    "BeginOffset": 0,  
    "EndOffset": 4  
  },  
  {  
    "Score": 0.9519651532173157,  
    "Type": "DATE",  
    "Text": "this weekend",  
    "BeginOffset": 47,  
    "EndOffset": 59  
  },  
  {  
    "Score": 0.5566426515579224,  
    "Type": "ORGANIZATION",  
    "Text": "AnySpa",  
    "BeginOffset": 63,  
    "EndOffset": 69  
  },  
  {  
    "Score": 0.8059805631637573,  
    "Type": "LOCATION",  
    "Text": "123 Main St, Anywhere",  
    "BeginOffset": 71,  
    "EndOffset": 92  
  },  
  {  
    "Score": 0.998830258846283,  
    "Type": "PERSON",  
    "Text": "Alice",  
    "BeginOffset": 114,  
    "EndOffset": 119  
  },  
  {  
    "Score": 0.997818112373352,  
    "Type": "OTHER",  
    "Text": "AnySpa@example.com",  
    "BeginOffset": 123,  
    "EndOffset": 138  
  }  
],
```

```

    "File": "SampleText3.txt",
    "Line": 0
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

Beispiel 2: So starten Sie einen benutzerdefinierten Auftrag zur Erkennung von Entitäten

Im folgenden `start-entities-detection-job` Beispiel wird ein asynchroner Auftrag zur Erkennung benutzerdefinierter Entitäten für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. In diesem Beispiel enthält der S3-Bucket in diesem Beispiel `SampleFeedback1.txt`, `SampleFeedback2.txt`, und `SampleFeedback3.txt`. Das Entity Recognizer-Modell wurde anhand der Rückmeldungen des Kundensupports trainiert, um Gerätenamen zu erkennen. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Speicherort abgelegt. Der Ordner enthält eine Liste aller benannten Entitäten `output.txt`, die in jeder Textdatei erkannt wurden, sowie den Konfidenzwert des vortrainierten Modells für jede Vorhersage. Die Json-Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```

aws comprehend start-entities-detection-job \
  --job-name customentiestest \
  --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/entityrecognizer" \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/jobdata/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-I0role"

```

Inhalt von `SampleFeedback1.txt`:

```

"I've been on the AnyPhone app have had issues for 24 hours when trying to pay bill.
Cannot make payment. Sigh. | Oh man! Lets get that app up and running. DM me, and
we can get to work!"

```

Inhalt von `SampleFeedback2.txt`:

```
"Hi, I have a discrepancy with my new bill. Could we get it sorted out? A rep added stuff I didnt sign up for when I did my AnyPhone 10 upgrade. | We can absolutely get this sorted!"
```

Inhalt von SampleFeedback3.txt:

```
"Is the by 1 get 1 free AnySmartPhone promo still going on? | Hi Christian! It ended yesterday, send us a DM if you have any questions and we can take a look at your options!"
```

Ausgabe:

```
{
  "JobId": "019ea9edac758806850fa8a79ff83021",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-job/019ea9edac758806850fa8a79ff83021",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von output.txt mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
  "Entities": [
    {
      "BeginOffset": 17,
      "EndOffset": 25,
      "Score": 0.9999728210205924,
      "Text": "AnyPhone",
      "Type": "DEVICE"
    }
  ],
  "File": "SampleFeedback1.txt",
  "Line": 0
}
{
  "Entities": [
    {
      "BeginOffset": 123,
      "EndOffset": 133,
      "Score": 0.9999892116761524,
      "Text": "AnyPhone 10",

```

```

    "Type": "DEVICE"
  }
],
"File": "SampleFeedback2.txt",
"Line": 0
}
{
  "Entities": [
    {
      "BeginOffset": 23,
      "EndOffset": 35,
      "Score": 0.9999971389852362,
      "Text": "AnySmartPhone",
      "Type": "DEVICE"
    }
  ],
  "File": "SampleFeedback3.txt",
  "Line": 0
}

```

Weitere Informationen finden Sie unter [Custom Entity Recognition](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [StartEntitiesDetectionJob](#) in der AWS CLI Befehlsreferenz.

start-events-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-events-detection-job`.

AWS CLI

Um einen Job zur Erkennung asynchroner Ereignisse zu starten

Im folgenden `start-events-detection-job` Beispiel wird ein Auftrag zur Erkennung asynchroner Ereignisse für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Mögliche Zielereignistypen sind `BANKRUPTCYEMPLOYMENT`, `CORPORATE_ACQUISITION`, `INVESTMENT_GENERAL`, `CORPORATE_MERGER`, `SHELF_OFFERINGTENDER_OFFERING`, und `STOCK_SPLIT`. Der S3-Bucket in diesem Beispiel enthält `SampleText1.txt`, `SampleText2.txt`, und `SampleText3.txt`. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Ort platziert. Der Ordner enthält `SampleText1.txt.out`, `SampleText2.txt.out`, und `SampleText3.txt.out`. Die JSON

Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```
aws comprehend start-events-detection-job \
  --job-name events-detection-1 \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/EventsData" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole \
  --language-code en \
  --target-event-
types "BANKRUPTCY" "EMPLOYMENT" "CORPORATE_ACQUISITION" "CORPORATE_MERGER" "INVESTMENT_GENER
```

Inhalt von SampleText1.txt:

```
"Company AnyCompany grew by increasing sales and through acquisitions. After
purchasing competing firms in 2020, AnyBusiness, a part of the AnyBusinessGroup,
gave Jane Does firm a going rate of one cent a gallon or forty-two cents a barrel."
```

Inhalt von SampleText2.txt:

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars,
surprising and exciting the shareholders."
```

Inhalt von SampleText3.txt:

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for
bankruptcy."
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von SampleText1.txt.out mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
```



```
"Entities": [  
  {  
    "Mentions": [  
      {  
        "BeginOffset": 8,  
        "EndOffset": 18,  
        "Score": 0.99977,  
        "Text": "AnyCompany",  
        "Type": "ORGANIZATION",  
        "GroupScore": 1  
      },  
      {  
        "BeginOffset": 112,  
        "EndOffset": 123,  
        "Score": 0.999747,  
        "Text": "AnyBusiness",  
        "Type": "ORGANIZATION",  
        "GroupScore": 0.979826  
      },  
      {  
        "BeginOffset": 171,  
        "EndOffset": 175,  
        "Score": 0.999615,  
        "Text": "firm",  
        "Type": "ORGANIZATION",  
        "GroupScore": 0.871647  
      }  
    ]  
  },  
  {  
    "Mentions": [  
      {  
        "BeginOffset": 97,  
        "EndOffset": 102,  
        "Score": 0.987687,  
        "Text": "firms",  
        "Type": "ORGANIZATION",  
        "GroupScore": 1  
      }  
    ]  
  },  
  {  
    "Mentions": [  
      {
```

```
        "BeginOffset": 103,
        "EndOffset": 110,
        "Score": 0.999458,
        "Text": "in 2020",
        "Type": "DATE",
        "GroupScore": 1
    }
]
},
{
  "Mentions": [
    {
      "BeginOffset": 160,
      "EndOffset": 168,
      "Score": 0.999649,
      "Text": "John Doe",
      "Type": "PERSON",
      "GroupScore": 1
    }
  ]
}
],
"Events": [
  {
    "Type": "CORPORATE_ACQUISITION",
    "Arguments": [
      {
        "EntityIndex": 0,
        "Role": "INVESTOR",
        "Score": 0.99977
      }
    ]
  },
  {
    "Type": "CORPORATE_ACQUISITION",
    "Arguments": [
      {
        "EntityIndex": 0,
        "Role": "INVESTOR",
        "Score": 0.99977
      }
    ]
  }
],
"Triggers": [
  {
    "BeginOffset": 56,
    "EndOffset": 68,
    "Score": 0.999967,
    "Text": "acquisitions",
    "Type": "CORPORATE_ACQUISITION",
    "GroupScore": 1
  }
]
},
{
```

```
"Type": "CORPORATE_ACQUISITION",
"Arguments": [
  {
    "EntityIndex": 1,
    "Role": "INVESTEES",
    "Score": 0.987687
  },
  {
    "EntityIndex": 2,
    "Role": "DATE",
    "Score": 0.999458
  },
  {
    "EntityIndex": 3,
    "Role": "INVESTOR",
    "Score": 0.999649
  }
],
"Triggers": [
  {
    "BeginOffset": 76,
    "EndOffset": 86,
    "Score": 0.999973,
    "Text": "purchasing",
    "Type": "CORPORATE_ACQUISITION",
    "GroupScore": 1
  }
]
}
"File": "SampleText1.txt",
"Line": 0
}
```

Inhalt von SampleText2.txt.out:

```
{
  "Entities": [
    {
      "Mentions": [
        {
          "BeginOffset": 0,
          "EndOffset": 7,
```

```
        "Score": 0.999473,
        "Text": "In 2021",
        "Type": "DATE",
        "GroupScore": 1
      }
    ],
  },
  {
    "Mentions": [
      {
        "BeginOffset": 9,
        "EndOffset": 19,
        "Score": 0.999636,
        "Text": "AnyCompany",
        "Type": "ORGANIZATION",
        "GroupScore": 1
      }
    ]
  },
  {
    "Mentions": [
      {
        "BeginOffset": 45,
        "EndOffset": 56,
        "Score": 0.999712,
        "Text": "AnyBusiness",
        "Type": "ORGANIZATION",
        "GroupScore": 1
      }
    ]
  },
  {
    "Mentions": [
      {
        "BeginOffset": 61,
        "EndOffset": 80,
        "Score": 0.998886,
        "Text": "100 billion dollars",
        "Type": "MONETARY_VALUE",
        "GroupScore": 1
      }
    ]
  }
],
```

```
"Events": [
  {
    "Type": "CORPORATE_ACQUISITION",
    "Arguments": [
      {
        "EntityIndex": 3,
        "Role": "AMOUNT",
        "Score": 0.998886
      },
      {
        "EntityIndex": 2,
        "Role": "INVESTEES",
        "Score": 0.999712
      },
      {
        "EntityIndex": 0,
        "Role": "DATE",
        "Score": 0.999473
      },
      {
        "EntityIndex": 1,
        "Role": "INVESTOR",
        "Score": 0.999636
      }
    ],
    "Triggers": [
      {
        "BeginOffset": 31,
        "EndOffset": 40,
        "Score": 0.99995,
        "Text": "purchased",
        "Type": "CORPORATE_ACQUISITION",
        "GroupScore": 1
      }
    ]
  }
],
"File": "SampleText2.txt",
"Line": 0
}
```

Inhalt von SampleText3.txt.out:

```
{
  "Entities": [
    {
      "Mentions": [
        {
          "BeginOffset": 9,
          "EndOffset": 19,
          "Score": 0.999774,
          "Text": "AnyCompany",
          "Type": "ORGANIZATION",
          "GroupScore": 1
        },
        {
          "BeginOffset": 66,
          "EndOffset": 70,
          "Score": 0.995717,
          "Text": "they",
          "Type": "ORGANIZATION",
          "GroupScore": 0.997626
        }
      ]
    },
    {
      "Mentions": [
        {
          "BeginOffset": 50,
          "EndOffset": 65,
          "Score": 0.999656,
          "Text": "later that year",
          "Type": "DATE",
          "GroupScore": 1
        }
      ]
    }
  ],
  "Events": [
    {
      "Type": "BANKRUPTCY",
      "Arguments": [
        {
          "EntityIndex": 1,
          "Role": "DATE",
          "Score": 0.999656
        }
      ]
    }
  ]
}
```

```
    },
    {
      "EntityIndex": 0,
      "Role": "FILER",
      "Score": 0.995717
    }
  ],
  "Triggers": [
    {
      "BeginOffset": 81,
      "EndOffset": 91,
      "Score": 0.999936,
      "Text": "bankruptcy",
      "Type": "BANKRUPTCY",
      "GroupScore": 1
    }
  ]
}
],
"File": "SampleText3.txt",
"Line": 0
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StartEventsDetectionJob](#) AWS CLI

start-flywheel-iteration

Das folgende Codebeispiel zeigt die Verwendung `start-flywheel-iteration`.

AWS CLI

Um eine Schwungrad-Iteration zu starten

Im folgenden `start-flywheel-iteration` Beispiel wird eine Schwungrad-Iteration gestartet. Bei dieser Operation werden alle neuen Datensätze im Schwungrad verwendet, um eine neue Modellversion zu trainieren.

```
aws comprehend start-flywheel-iteration \
```

```
--flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

Ausgabe:

```
{
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",
  "FlywheelIterationId": "12345123TEXAMPLE"
}
```

Weitere Informationen finden Sie in der [Übersicht über Flywheel](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [StartFlywheelIteration](#) in AWS CLI der Befehlsreferenz.

start-key-phrases-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-key-phrases-detection-job`.

AWS CLI

Um einen Job zur Erkennung von Schlüsselphrasen zu starten

Im folgenden `start-key-phrases-detection-job` Beispiel wird ein asynchroner Auftrag zur Erkennung von Schlüsselphrasen für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Der S3-Bucket in diesem Beispiel enthält `Sampletext1.txt`, `Sampletext2.txt`, und `Sampletext3.txt`. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Ort platziert. Der Ordner enthält `output.txt` die Datei mit allen Schlüsselbegriffen, die in jeder Textdatei erkannt wurden, sowie den Konfidenzwert des vortrainierten Modells für jede Vorhersage. Die Json-Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```
aws comprehend start-key-phrases-detection-job \
  --job-name keyphrasesanalysistest1 \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role" \
```



```
--language-code en
```

Inhalt von Sampletext1.txt:

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July 31st."
```

Inhalt von Sampletext2.txt:

```
"Dear Max, based on your autopay settings for your account Internet.org account, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000. "
```

Inhalt von Sampletext3.txt:

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123 Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von output.txt mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
  "File": "SampleText1.txt",
  "KeyPhrases": [
    {
      "BeginOffset": 6,
      "EndOffset": 15,
      "Score": 0.9748965572679326,
      "Text": "Zhang Wei"
    },
    {
      "BeginOffset": 22,
```

```
    "EndOffset": 26,
    "Score": 0.9997344722354619,
    "Text": "John"
  },
  {
    "BeginOffset": 28,
    "EndOffset": 62,
    "Score": 0.9843791074032948,
    "Text": "Your AnyCompany Financial Services"
  },
  {
    "BeginOffset": 64,
    "EndOffset": 107,
    "Score": 0.8976122401721824,
    "Text": "LLC credit card account 1111-XXXX-1111-XXXX"
  },
  {
    "BeginOffset": 112,
    "EndOffset": 129,
    "Score": 0.9999612982629748,
    "Text": "a minimum payment"
  },
  {
    "BeginOffset": 133,
    "EndOffset": 139,
    "Score": 0.99975728947036,
    "Text": "$24.53"
  },
  {
    "BeginOffset": 155,
    "EndOffset": 164,
    "Score": 0.9940866241449973,
    "Text": "July 31st"
  }
],
"Line": 0
}
{
  "File": "SampleText2.txt",
  "KeyPhrases": [
    {
      "BeginOffset": 0,
      "EndOffset": 8,
      "Score": 0.9974021100118472,
```

```
    "Text": "Dear Max"
  },
  {
    "BeginOffset": 19,
    "EndOffset": 40,
    "Score": 0.9961120519515884,
    "Text": "your autopay settings"
  },
  {
    "BeginOffset": 45,
    "EndOffset": 78,
    "Score": 0.9980620070116009,
    "Text": "your account Internet.org account"
  },
  {
    "BeginOffset": 97,
    "EndOffset": 109,
    "Score": 0.999919660140754,
    "Text": "your payment"
  },
  {
    "BeginOffset": 113,
    "EndOffset": 125,
    "Score": 0.9998370719754205,
    "Text": "the due date"
  },
  {
    "BeginOffset": 131,
    "EndOffset": 166,
    "Score": 0.9955068678502509,
    "Text": "your bank account number XXXXXX1111"
  },
  {
    "BeginOffset": 172,
    "EndOffset": 200,
    "Score": 0.8653433315829526,
    "Text": "the routing number XXXXX0000"
  }
],
"Line": 0
}
{
  "File": "SampleText3.txt",
  "KeyPhrases": [
```

```
{
  "BeginOffset": 0,
  "EndOffset": 4,
  "Score": 0.9142947833681668,
  "Text": "Jane"
},
{
  "BeginOffset": 20,
  "EndOffset": 41,
  "Score": 0.9984325676596763,
  "Text": "any customer feedback"
},
{
  "BeginOffset": 47,
  "EndOffset": 59,
  "Score": 0.9998782448150636,
  "Text": "this weekend"
},
{
  "BeginOffset": 63,
  "EndOffset": 75,
  "Score": 0.99866741830757,
  "Text": "Sunshine Spa"
},
{
  "BeginOffset": 77,
  "EndOffset": 88,
  "Score": 0.9695803485466054,
  "Text": "123 Main St"
},
{
  "BeginOffset": 108,
  "EndOffset": 116,
  "Score": 0.9997065928550928,
  "Text": "comments"
},
{
  "BeginOffset": 120,
  "EndOffset": 125,
  "Score": 0.9993466833825161,
  "Text": "Alice"
},
{
  "BeginOffset": 129,
```

```

    "EndOffset": 144,
    "Score": 0.9654563612885667,
    "Text": "AnySpa@example.com"
  }
],
"Line": 0
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StartKeyPhrasesDetectionJob](#) AWS CLI

start-pii-entities-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-pii-entities-detection-job`.

AWS CLI

Um einen asynchronen PII Erkennungsauftrag zu starten

Im folgenden `start-pii-entities-detection-job` Beispiel wird eine asynchrone Aufgabe zur Erkennung von Entitäten persönlicher identifizierbarer Informationen (PII) für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Der S3-Bucket in diesem Beispiel enthält `Sampletext1.txt`, `Sampletext2.txt`, und `Sampletext3.txt`. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Ort platziert. Der Ordner enthält `SampleText1.txt.out`, `SampleText2.txt.out`, in `SampleText3.txt.out` denen die benannten Entitäten in jeder Textdatei aufgeführt sind. Die JSON-Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```

aws comprehend start-pii-entities-detection-job \
  --job-name entities_test \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  

AmazonComprehendServiceRole-example-role \
  --language-code en \
  --mode ONLY_OFFSETS

```

Inhalt von Sampletext1.txt:

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July 31st."
```

Inhalt von Sampletext2.txt:

```
"Dear Max, based on your autopay settings for your account Internet.org account, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000. "
```

Inhalt von Sampletext3.txt:

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123 Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von SampleText1.txt.out mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
  "Entities": [
    {
      "BeginOffset": 6,
      "EndOffset": 15,
      "Type": "NAME",
      "Score": 0.9998490510222595
    },
    {
      "BeginOffset": 22,
      "EndOffset": 26,
      "Type": "NAME",
      "Score": 0.9998937958019426
    }
  ]
}
```

```
    },
    {
      "BeginOffset": 88,
      "EndOffset": 107,
      "Type": "CREDIT_DEBIT_NUMBER",
      "Score": 0.9554297245278491
    },
    {
      "BeginOffset": 155,
      "EndOffset": 164,
      "Type": "DATE_TIME",
      "Score": 0.9999720462925257
    }
  ],
  "File": "SampleText1.txt",
  "Line": 0
}
```

Inhalt von `SampleText2.txt.out` mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
  "Entities": [
    {
      "BeginOffset": 5,
      "EndOffset": 8,
      "Type": "NAME",
      "Score": 0.9994390774924007
    },
    {
      "BeginOffset": 58,
      "EndOffset": 70,
      "Type": "URL",
      "Score": 0.9999958276922101
    },
    {
      "BeginOffset": 156,
      "EndOffset": 166,
      "Type": "BANK_ACCOUNT_NUMBER",
      "Score": 0.9999721058045592
    },
    {
      "BeginOffset": 191,
      "EndOffset": 200,
```

```
    "Type": "BANK_ROUTING",
    "Score": 0.9998968945989909
  },
  "File": "SampleText2.txt",
  "Line": 0
}
```

Inhalt von `SampleText3.txt.out` mit Zeileneinzügen zur besseren Lesbarkeit:

```
{
  "Entities": [
    {
      "BeginOffset": 0,
      "EndOffset": 4,
      "Type": "NAME",
      "Score": 0.999949934606805
    },
    {
      "BeginOffset": 77,
      "EndOffset": 88,
      "Type": "ADDRESS",
      "Score": 0.9999035300466904
    },
    {
      "BeginOffset": 120,
      "EndOffset": 125,
      "Type": "NAME",
      "Score": 0.9998203838716296
    },
    {
      "BeginOffset": 129,
      "EndOffset": 144,
      "Type": "EMAIL",
      "Score": 0.9998313473105228
    }
  ],
  "File": "SampleText3.txt",
  "Line": 0
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StartPiiEntitiesDetectionJob](#) AWS CLI

start-sentiment-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-sentiment-detection-job`.

AWS CLI

Um einen asynchronen Stimmungsanalysejob zu starten

Im folgenden `start-sentiment-detection-job` Beispiel wird ein asynchroner Auftrag zur Erkennung der Stimmungsanalyse für alle Dateien gestartet, die sich an der durch das Tag angegebenen Adresse befinden. `--input-data-config` Der S3-Bucket-Ordner in diesem Beispiel enthält `SampleMovieReview1.txt`, `SampleMovieReview2.txt`, und `SampleMovieReview3.txt`. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an der durch das `--output-data-config` Tag angegebenen Position platziert. Der Ordner enthält die Datei `output.txt`, die die vorherrschenden Einstellungen für jede Textdatei und den Konfidenzwert des vortrainierten Modells für jede Vorhersage enthält. Die Json-Ausgabe wird in einer Zeile pro Datei gedruckt, ist hier aber aus Gründen der Lesbarkeit formatiert.

```
aws comprehend start-sentiment-detection-job \
  --job-name example-sentiment-detection-job \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/MovieData" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role
```

Inhalt von `SampleMovieReview1.txt`:

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

Inhalt von `SampleMovieReview2.txt`:

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I highly recommend this movie."
```

Inhalt von `SampleMovieReview3.txt`:

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly
stolen from other modern directors."
```

Ausgabe:

```
{
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/0b5001e25f62ebb40631a9a1a7fde7b3",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von aus Gründen der Lesbarkeit output .txt mit Einrückungen:

```
{
  "File": "SampleMovieReview1.txt",
  "Line": 0,
  "Sentiment": "MIXED",
  "SentimentScore": {
    "Mixed": 0.6591159105300903,
    "Negative": 0.26492202281951904,
    "Neutral": 0.035430654883384705,
    "Positive": 0.04053137078881264
  }
}
{
  "File": "SampleMovieReview2.txt",
  "Line": 0,
  "Sentiment": "POSITIVE",
  "SentimentScore": {
    "Mixed": 0.000008718466233403888,
    "Negative": 0.00006134175055194646,
    "Neutral": 0.0002941041602753103,
    "Positive": 0.9996358156204224
  }
}
{
  "File": "SampleMovieReview3.txt",
  "Line": 0,
  "Sentiment": "NEGATIVE",
  "SentimentScore": {
    "Mixed": 0.004146667663007975,
```

```

        "Negative": 0.9645107984542847,
        "Neutral": 0.016559595242142677,
        "Positive": 0.014782938174903393
    }
}
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StartSentimentDetectionJob](#) AWS CLI

start-targeted-sentiment-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-targeted-sentiment-detection-job`.

AWS CLI

Um einen asynchronen Job zur gezielten Stimmungsanalyse zu starten

Im folgenden `start-targeted-sentiment-detection-job` Beispiel wird ein asynchroner Auftrag zur Erkennung einer gezielten Stimmungsanalyse für alle Dateien gestartet, die sich an der durch das Tag angegebenen Adresse befinden. `--input-data-config` Der S3-Bucket-Ordner in diesem Beispiel enthält `SampleMovieReview1.txt`, `SampleMovieReview2.txt`, und `SampleMovieReview3.txt`. Wenn der Job abgeschlossen ist, `output.tar.gz` wird er an der durch das `--output-data-config` Tag angegebenen Position platziert. `output.tar.gz` enthält die Dateien `SampleMovieReview1.txt.out`, und `SampleMovieReview2.txt.out`, `SampleMovieReview3.txt.out`, die jeweils alle benannten Entitäten und zugehörigen Stimmungen für eine einzelne Eingabetextdatei enthalten.

```

aws comprehend start-targeted-sentiment-detection-job \
  --job-name targeted_movie_review_analysis1 \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/MovieData" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role

```

Inhalt von `SampleMovieReview1.txt`:

```
"The film, AnyMovie, is fairly predictable and just okay."
```

Inhalt von SampleMovieReview2.txt:

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I highly recommend this movie."
```

Inhalt von SampleMovieReview3.txt:

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly stolen from other modern directors."
```

Ausgabe:

```
{
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",
  "JobStatus": "SUBMITTED"
}
```

Inhalt von SampleMovieReview1.txt.out mit Zeileneinbrüchen zur besseren Lesbarkeit:

```
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 4,
          "EndOffset": 8,
          "Score": 0.994972,
          "GroupScore": 1,
          "Text": "film",
          "Type": "MOVIE",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0,
              "Neutral": 1,
              "Positive": 0
            }
          }
        }
      ]
    }
  ]
}
```

```

    }
  }
}
],
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 10,
      "EndOffset": 18,
      "Score": 0.631368,
      "GroupScore": 1,
      "Text": "AnyMovie",
      "Type": "ORGANIZATION",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Mixed": 0.001729,
          "Negative": 0.000001,
          "Neutral": 0.000318,
          "Positive": 0.997952
        }
      }
    }
  ]
}
],
"File": "SampleMovieReview1.txt",
"Line": 0
}

```

Inhalt der SampleMovieReview2.txt.out Zeileneinzüge zur besseren Lesbarkeit:

```

{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [

```

```
{
  "BeginOffset": 0,
  "EndOffset": 8,
  "Score": 0.854024,
  "GroupScore": 1,
  "Text": "AnyMovie",
  "Type": "MOVIE",
  "MentionSentiment": {
    "Sentiment": "POSITIVE",
    "SentimentScore": {
      "Mixed": 0,
      "Negative": 0,
      "Neutral": 0.000007,
      "Positive": 0.999993
    }
  }
},
{
  "BeginOffset": 104,
  "EndOffset": 109,
  "Score": 0.999129,
  "GroupScore": 0.502937,
  "Text": "movie",
  "Type": "MOVIE",
  "MentionSentiment": {
    "Sentiment": "POSITIVE",
    "SentimentScore": {
      "Mixed": 0,
      "Negative": 0,
      "Neutral": 0,
      "Positive": 1
    }
  }
},
{
  "BeginOffset": 33,
  "EndOffset": 37,
  "Score": 0.999823,
  "GroupScore": 0.999252,
  "Text": "film",
  "Type": "MOVIE",
  "MentionSentiment": {
    "Sentiment": "POSITIVE",
    "SentimentScore": {
```

```
        "Mixed": 0,
        "Negative": 0,
        "Neutral": 0.000001,
        "Positive": 0.999999
      }
    }
  ],
},
{
  "DescriptiveMentionIndex": [
    0,
    1,
    2
  ],
  "Mentions": [
    {
      "BeginOffset": 43,
      "EndOffset": 44,
      "Score": 0.999997,
      "GroupScore": 1,
      "Text": "I",
      "Type": "PERSON",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 1,
          "Positive": 0
        }
      }
    }
  ],
  {
    "BeginOffset": 80,
    "EndOffset": 81,
    "Score": 0.999996,
    "GroupScore": 0.52523,
    "Text": "I",
    "Type": "PERSON",
    "MentionSentiment": {
      "Sentiment": "NEUTRAL",
      "SentimentScore": {
        "Mixed": 0,
```

```
        "Negative": 0,
        "Neutral": 1,
        "Positive": 0
      }
    }
  },
  {
    "BeginOffset": 67,
    "EndOffset": 68,
    "Score": 0.999994,
    "GroupScore": 0.999499,
    "Text": "I",
    "Type": "PERSON",
    "MentionSentiment": {
      "Sentiment": "NEUTRAL",
      "SentimentScore": {
        "Mixed": 0,
        "Negative": 0,
        "Neutral": 1,
        "Positive": 0
      }
    }
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 75,
      "EndOffset": 78,
      "Score": 0.999978,
      "GroupScore": 1,
      "Text": "kid",
      "Type": "PERSON",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 1,
          "Positive": 0
        }
      }
    }
  ]
}
```



```

    }
  }
}
],
"File": "SampleMovieReview2.txt",
"Line": 0
}

```

Inhalt von SampleMovieReview3.txt.out mit Zeileneinzügen zur besseren Lesbarkeit:

```

{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        1
      ],
      "Mentions": [
        {
          "BeginOffset": 64,
          "EndOffset": 68,
          "Score": 0.992953,
          "GroupScore": 0.999814,
          "Text": "film",
          "Type": "MOVIE",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0.000004,
              "Negative": 0.010425,
              "Neutral": 0.989543,
              "Positive": 0.000027
            }
          }
        }
      ],
      {
        "BeginOffset": 37,
        "EndOffset": 45,
        "Score": 0.999782,
        "GroupScore": 1,
        "Text": "AnyMovie",
        "Type": "ORGANIZATION",

```

```
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0.000095,
        "Negative": 0.039847,
        "Neutral": 0.000673,
        "Positive": 0.959384
      }
    }
  },
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
        "BeginOffset": 47,
        "EndOffset": 50,
        "Score": 0.999991,
        "GroupScore": 1,
        "Text": "All",
        "Type": "QUANTITY",
        "MentionSentiment": {
          "Sentiment": "NEUTRAL",
          "SentimentScore": {
            "Mixed": 0.000001,
            "Negative": 0.000001,
            "Neutral": 0.999998,
            "Positive": 0
          }
        }
      }
    ]
  },
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
        "BeginOffset": 106,
        "EndOffset": 115,
```

```

        "Score": 0.542083,
        "GroupScore": 1,
        "Text": "directors",
        "Type": "PERSON",
        "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
            }
        }
    ]
}
],
"File": "SampleMovieReview3.txt",
"Line": 0
}

```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StartTargetedSentimentDetectionJob](#) AWS CLI

start-topics-detection-job

Das folgende Codebeispiel zeigt die Verwendung `start-topics-detection-job`.

AWS CLI

Um einen Analyseauftrag zur Themenerkennung zu starten

Im folgenden `start-topics-detection-job` Beispiel wird ein asynchroner Auftrag zur Themenerkennung für alle Dateien gestartet, die sich an der durch das `--input-data-config` Tag angegebenen Adresse befinden. Wenn der Job abgeschlossen ist, wird der Ordner `output`, an dem durch das `--output-data-config` Tag angegebenen Speicherort platziert. `output` enthält `topic-terms.csv` und `doc-topics.csv`. Die erste Ausgabedatei, `topic-terms.csv`, ist eine Liste von Themen in der Sammlung. Für jedes Thema enthält die Liste standardmäßig die wichtigsten Begriffe, sortiert nach Themen, entsprechend ihrer Gewichtung.

In der zweiten Datei werden die Dokumente aufgeführt `doc-topics.csv`, die einem Thema zugeordnet sind, sowie der Anteil des Dokuments, der sich mit dem Thema befasst.

```
aws comprehend start-topics-detection-job \  
  --job-name example_topics_detection_job \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --language-code en
```

Ausgabe:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

Weitere Informationen finden Sie unter [Topic Modeling](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [StartTopicsDetectionJob](#) in der AWS CLI Befehlsreferenz.

stop-dominant-language-detection-job

Das folgende Codebeispiel zeigt die Verwendung `stop-dominant-language-detection-job`.

AWS CLI

Um einen asynchronen Job zur Erkennung dominanter Sprachen zu beenden

Im folgenden `stop-dominant-language-detection-job` Beispiel wird ein in Bearbeitung befindlicher asynchroner Auftrag zur Erkennung dominanter Sprache beendet. Wenn der aktuelle Jobstatus lautet, wird `IN_PROGRESS` der Job zur Kündigung markiert und in den `STOP_REQUESTED` entsprechenden Status versetzt. Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den `COMPLETED` Status versetzt.

```
aws comprehend stop-dominant-language-detection-job \  
  --job-name example_stop_dominant_language_detection_job \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --language-code en
```

```
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopDominantLanguageDetectionJobAWS CLI](#)

stop-entities-detection-job

Das folgende Codebeispiel zeigt die Verwendung `stop-entities-detection-job`.

AWS CLI

Um einen Job zur Erkennung asynchroner Entitäten zu beenden

Im folgenden `stop-entities-detection-job` Beispiel wird ein laufender Auftrag zur Erkennung asynchroner Entitäten beendet. Wenn der aktuelle Jobstatus lautet `IN_PROGRESS`, wird der Job zur Kündigung markiert und in den `STOP_REQUESTED` entsprechenden Status versetzt. Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den `COMPLETED` Status versetzt.

```
aws comprehend stop-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopEntitiesDetectionJob](#) AWS CLI

stop-events-detection-job

Das folgende Codebeispiel zeigt die Verwendung `stop-events-detection-job`.

AWS CLI

Um einen Job zur Erkennung asynchroner Ereignisse zu beenden

Im folgenden `stop-events-detection-job` Beispiel wird ein laufender, asynchroner Job zur Erkennung von Ereignissen beendet. Wenn der aktuelle Jobstatus lautet, wird `IN_PROGRESS` der Job zur Kündigung markiert und in den `STOP_REQUESTED` entsprechenden Status versetzt. Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den `COMPLETED` Status versetzt.

```
aws comprehend stop-events-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopEventsDetectionJob](#) AWS CLI

stop-key-phrases-detection-job

Das folgende Codebeispiel zeigt die Verwendung `stop-key-phrases-detection-job`.

AWS CLI

Um einen Job zur Erkennung asynchroner Schlüsselphrasen zu beenden

Im folgenden `stop-key-phrases-detection-job` Beispiel wird ein laufender, asynchroner Auftrag zur Erkennung von Schlüsselbegriffen beendet. Wenn der aktuelle Jobstatus lautet, wird

IN_PROGRESS der Job zur Kündigung markiert und in den STOP_REQUESTED entsprechenden Status versetzt. Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den COMPLETED Status versetzt.

```
aws comprehend stop-key-phrases-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopKeyPhrasesDetectionJob](#) AWS CLI

stop-pii-entities-detection-job

Das folgende Codebeispiel zeigt die Verwendung stop-pii-entities-detection-job.

AWS CLI

Um einen asynchronen Job zur Erkennung von PII-Entitäten zu beenden

Das folgende stop-pii-entities-detection-job Beispiel beendet einen laufenden, asynchronen Job zur Erkennung von PII-Entitäten. Wenn der aktuelle Jobstatus lautet, wird IN_PROGRESS der Job zur Kündigung markiert und in den Status versetzt. STOP_REQUESTED Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den COMPLETED Status versetzt.

```
aws comprehend stop-pii-entities-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
```

```
"JobStatus": "STOP_REQUESTED"
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopPiiEntitiesDetectionJob](#) AWS CLI

stop-sentiment-detection-job

Das folgende Codebeispiel zeigt die Verwendung `stop-sentiment-detection-job`.

AWS CLI

Um einen asynchronen Stimmungserkennungsjob zu beenden

Im folgenden `stop-sentiment-detection-job` Beispiel wird ein laufender, asynchroner Stimmungserkennungsauftrag beendet. Wenn der aktuelle Jobstatus lautet `IN_PROGRESS`, wird der Job zur Kündigung markiert und in den entsprechenden Status versetzt. `STOP_REQUESTED`. Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den `COMPLETED` Status versetzt.

```
aws comprehend stop-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopSentimentDetectionJob](#) AWS CLI

stop-targeted-sentiment-detection-job

Das folgende Codebeispiel zeigt die Verwendung `stop-targeted-sentiment-detection-job`.

AWS CLI

Um einen asynchronen Job zur gezielten Stimmungserkennung zu beenden

Im folgenden `stop-targeted-sentiment-detection-job` Beispiel wird ein laufender, asynchroner Auftrag zur gezielten Stimmungserkennung gestoppt. Wenn der aktuelle Jobstatus lautet, wird `IN_PROGRESS` der Job zur Kündigung markiert und in den entsprechenden Status versetzt. `STOP_REQUESTED` Wenn der Job abgeschlossen ist, bevor er gestoppt werden kann, wird er in den `COMPLETED` Status versetzt.

```
aws comprehend stop-targeted-sentiment-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Ausgabe:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Weitere Informationen finden Sie unter [Async-Analyse für Amazon Comprehend Insights im Amazon Comprehend Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [StopTargetedSentimentDetectionJob](#) AWS CLI

`stop-training-document-classifier`

Das folgende Codebeispiel zeigt die Verwendung `stop-training-document-classifier`.

AWS CLI

Um das Training eines Dokumentenklassifikatormodells zu beenden

Im folgenden `stop-training-document-classifier` Beispiel wird das Training eines Dokumentenklassifizierer-Modells beendet, während das Training ausgeführt wird.

```
aws comprehend stop-training-document-classifier  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
  classifier/example-classifier
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [StopTrainingDocumentClassifier AWS CLI Befehlsreferenz](#).

stop-training-entity-recognizer

Das folgende Codebeispiel zeigt die Verwendung `stop-training-entity-recognizer`.

AWS CLI

Um das Training eines Entity Recognizer-Modells zu beenden

Im folgenden `stop-training-entity-recognizer` Beispiel wird das Training eines Entitätserkennungsmodells beendet, während es ausgeführt wird.

```
aws comprehend stop-training-entity-recognizer
  --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-
  recognizer/examplerrecognizer1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen und Verwalten von benutzerdefinierten Modellen](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie unter [StopTrainingEntityRecognizer AWS CLI Befehlsreferenz](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Beispiel 1: Um eine Ressource zu taggen

Das folgende `tag-resource` Beispiel fügt einer Amazon Comprehend Comprehend-Ressource ein einzelnes Tag hinzu.

```
aws comprehend tag-resource \
```

```
--resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
--tags Key=Location,Value=Seattle
```

Dieser Befehl hat keine Ausgabe.

Weitere Informationen finden Sie unter [Tagging Your Resources](#) im Amazon Comprehend Developer Guide.

Beispiel 2: So fügen Sie einer Ressource mehrere Tags hinzu

Das folgende `tag-resource` Beispiel fügt einer Amazon Comprehend Comprehend-Ressource mehrere Tags hinzu.

```
aws comprehend tag-resource \
--resource-arn "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1" \
--tags Key=Location,Value=Seattle Key=Department,Value=Finance
```

Dieser Befehl hat keine Ausgabe.

Weitere Informationen finden Sie unter [Tagging Your Resources](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Beispiel 1: Um ein einzelnes Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird ein einzelnes Tag aus einer Amazon Comprehend Comprehend-Ressource entfernt.

```
aws comprehend untag-resource \
--resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

```
--tag-keys Location
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your Resources](#) im Amazon Comprehend Developer Guide.

Beispiel 2: Um mehrere Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden mehrere Tags aus einer Amazon Comprehend Comprehend-Ressource entfernt.

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1  
  --tag-keys Location Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your Resources](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in AWS CLI der Befehlsreferenz.

update-endpoint

Das folgende Codebeispiel zeigt die Verwendung `update-endpoint`.

AWS CLI

Beispiel 1: Um die Inferenzeinheiten eines Endpunkts zu aktualisieren

Im folgenden `update-endpoint` Beispiel werden Informationen über einen Endpunkt aktualisiert. In diesem Beispiel wird die Anzahl der Inferenzeinheiten erhöht.

```
aws comprehend update-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint  
  --desired-inference-units 2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

Beispiel 2: Um das aktive Modell eines Endpunkts zu aktualisieren

Im folgenden `update-endpoint` Beispiel werden Informationen über einen Endpunkt aktualisiert. In diesem Beispiel wird das aktive Modell geändert.

```
aws comprehend update-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint \  
  --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-new
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Amazon Comprehend Endpoints](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [UpdateEndpoint](#) AWS CLI

update-flywheel

Das folgende Codebeispiel zeigt die Verwendung `update-flywheel`.

AWS CLI

Um eine Schwungradkonfiguration zu aktualisieren

Im folgenden `update-flywheel` Beispiel wird eine Schwungradkonfiguration aktualisiert. In diesem Beispiel wird das aktive Modell für das Schwungrad aktualisiert.

```
aws comprehend update-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel-1 \  
  --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier/version/new-example-classifier-model
```

Ausgabe:

```
{
```

```

    "FlywheelProperties": {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity",
      "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier/version/new-example-classifier-model",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
      "TaskConfig": {
        "LanguageCode": "en",
        "DocumentClassificationConfig": {
          "Mode": "MULTI_CLASS"
        }
      },
      "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/
schemaVersion=1/20230616T200543Z/",
      "DataSecurityConfig": {},
      "Status": "ACTIVE",
      "ModelType": "DOCUMENT_CLASSIFIER",
      "CreationTime": "2023-06-16T20:05:43.242000+00:00",
      "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
      "LatestFlywheelIteration": "20230619T040032Z"
    }
  }
}

```

Weitere Informationen finden Sie in der [Übersicht über Flywheel](#) im Amazon Comprehend Developer Guide.

- API-Einheiten finden Sie [UpdateFlywheel](#) in AWS CLI der Befehlsreferenz.

Beispiele von Amazon Comprehend Medical mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Comprehend Medical Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-entities-detection-v2-job

Das folgende Codebeispiel zeigt die Verwendung `describe-entities-detection-v2-job`.

AWS CLI

Um einen Job zur Erkennung von Entitäten zu beschreiben

Im folgenden `describe-entities-detection-v2-job` Beispiel werden die Eigenschaften eines asynchronen Entitätserkennungsauftrags angezeigt.

```
aws comprehendmedical describe-entities-detection-v2-job \
  --job-id "ab9887877365fe70299089371c043b96"
```

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "ab9887877365fe70299089371c043b96",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2020-03-18T21:20:15.614000+00:00",
    "EndTime": "2020-03-18T21:27:07.350000+00:00",
    "ExpirationTime": "2020-07-16T21:20:15+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": ""
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "DetectEntitiesModelV20190930"
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [DescribeEntitiesDetectionV2Job](#) in der Befehlsreferenz.AWS CLI

describe-icd10-cm-inference-job

Das folgende Codebeispiel zeigt, wie man es benutzt. describe-icd10-cm-inference-job

AWS CLI

Um einen ICD -10-CM-Inferenzjob zu beschreiben

Das folgende describe-icd10-cm-inference-job Beispiel beschreibt die Eigenschaften des angeforderten Inferenzjobs mit der angegebenen Job-ID.

```
aws comprehendmedical describe-icd10-cm-inference-job \  
  --job-id "5780034166536cdb52ffa3295a1b00a7"
```

Ausgabe:

```
{  
  "ComprehendMedicalAsyncJobProperties": {  
    "JobId": "5780034166536cdb52ffa3295a1b00a7",  
    "JobStatus": "COMPLETED",  
    "SubmitTime": "2020-05-18T21:20:15.614000+00:00",  
    "EndTime": "2020-05-18T21:27:07.350000+00:00",  
    "ExpirationTime": "2020-09-16T21:20:15+00:00",  
    "InputDataConfig": {  
      "S3Bucket": "comp-med-input",  
      "S3Key": "AKIAIOSFODNN7EXAMPLE"  
    },  
    "OutputDataConfig": {  
      "S3Bucket": "comp-med-output",  
      "S3Key": "AKIAIOSFODNN7EXAMPLE"  
    },  
    "LanguageCode": "en",  
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/  
ComprehendMedicalBatchProcessingRole",
```



```

    "ModelVersion": "0.1.0"
  }
}

```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [DescribeCdmInferenceJob](#) in AWS CLI der Befehlsreferenz.

describe-phi-detection-job

Das folgende Codebeispiel zeigt, wie man `describe-phi-detection-job`.

AWS CLI

Um einen PHI Erkennungsjob zu beschreiben

Im folgenden `describe-phi-detection-job` Beispiel werden die Eigenschaften eines asynchronen Erkennungsauftrags für geschützte Gesundheitsinformationen (PHI) angezeigt.

```

aws comprehendmedical describe-phi-detection-job \
  --job-id "4750034166536cdb52ffa3295a1b00a3"

```

Ausgabe:

```

{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "4750034166536cdb52ffa3295a1b00a3",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
    "EndTime": "2020-03-19T20:45:07.894000+00:00",
    "ExpirationTime": "2020-07-17T20:38:37+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": ""
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "867139942017-PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
    },
  },
}

```

```
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "PHIModelV20190903"
  }
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [DescribePhiDetectionJob](#) in der AWS CLI Befehlsreferenz.

describe-rx-norm-inference-job

Das folgende Codebeispiel zeigt die Verwendung `describe-rx-norm-inference-job`.

AWS CLI

Um einen RxNorm Inferenzjob zu beschreiben

Das folgende `describe-rx-norm-inference-job` Beispiel beschreibt die Eigenschaften des angeforderten Inferenzjobs mit der angegebenen Job-ID.

```
aws comprehendmedical describe-rx-norm-inference-job \
  --job-id "eg8199877365fc70299089371c043b96"
```

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "g8199877365fc70299089371c043b96",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
    "EndTime": "2020-05-18T21:27:07.350000+00:00",
    "ExpirationTime": "2020-09-16T21:20:15+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
```

```

        "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "0.0.0"
}
}

```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [DescribeRxNormInferenceJob](#) in AWS CLI der Befehlsreferenz.

describe-snomedct-inference-job

Das folgende Codebeispiel zeigt die Verwendung `describe-snomedct-inference-job`.

AWS CLI

Um einen SNOMED CT-Inferenzjob zu beschreiben

Das folgende `describe-snomedct-inference-job` Beispiel beschreibt die Eigenschaften des angeforderten Inferenzjobs mit der angegebenen Job-ID.

```

aws comprehendmedical describe-snomedct-inference-job \
  --job-id "2630034166536cdb52ffa3295a1b00a7"

```

Ausgabe:

```

{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "2630034166536cdb52ffa3295a1b00a7",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2021-12-18T21:20:15.614000+00:00",
    "EndTime": "2021-12-18T21:27:07.350000+00:00",
    "ExpirationTime": "2022-05-16T21:20:15+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    }
  },

```

```
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "0.1.0"
  }
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [DescribeSnomedctInferenceJob](#) in AWS CLI der Befehlsreferenz.

detect-entities-v2

Das folgende Codebeispiel zeigt die Verwendung `detect-entities-v2`.

AWS CLI

Beispiel 1: Um Entitäten direkt aus Text zu erkennen

Das folgende `detect-entities-v2` Beispiel zeigt die erkannten Entitäten und beschriftet sie nach Typ direkt aus dem eingegebenen Text.

```
aws comprehendmedical detect-entities-v2 \
  --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and
leg, slightly itchy."
```

Ausgabe:

```
{
  "Id": 0,
  "BeginOffset": 38,
  "EndOffset": 47,
  "Score": 0.9942955374717712,
  "Text": "Clonidine",
  "Category": "MEDICATION",
  "Type": "GENERIC_NAME",
  "Traits": []
}
```

```
}
```

Weitere Informationen finden Sie unter [Detect Entities Version 2](#) im Amazon Comprehend Medical Developer Guide.

Beispiel 2: So erkennen Sie Entitäten anhand eines Dateipfads

Das folgende detect-entities-v2 Beispiel zeigt die erkannten Entitäten und kennzeichnet sie entsprechend ihrem Typ aus einem Dateipfad.

```
aws comprehendmedical detect-entities-v2 \  
  --text file://medical_entities.txt
```

Inhalt von `medical_entities.txt`:

```
{  
  "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,  
  slightly itchy."  
}
```

Ausgabe:

```
{  
  "Id": 0,  
  "BeginOffset": 38,  
  "EndOffset": 47,  
  "Score": 0.9942955374717712,  
  "Text": "Clonidine",  
  "Category": "MEDICATION",  
  "Type": "GENERIC_NAME",  
  "Traits": []  
}
```

Weitere Informationen finden Sie unter [Detect Entities Version 2](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [DetectEntitiesV2](#) in der AWS CLI Befehlsreferenz.

detect-phi

Das folgende Codebeispiel zeigt die Verwendung `detect-phi`.

AWS CLI

Beispiel 1: Um geschützte Gesundheitsinformationen (PHI) direkt aus Text zu erkennen

Im folgenden `detect-phi` Beispiel werden die erkannten Entitäten mit geschützten Gesundheitsinformationen (PHI) direkt aus dem Eingabetext angezeigt.

```
aws comprehendmedical detect-phi \  
  --text "Patient Carlos Salazar presented with rash on his upper extremities and  
  dry cough. He lives at 100 Main Street, Anytown, USA where he works from his home  
  as a carpenter."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Id": 0,  
      "BeginOffset": 8,  
      "EndOffset": 21,  
      "Score": 0.9914507269859314,  
      "Text": "Carlos Salazar",  
      "Category": "PROTECTED_HEALTH_INFORMATION",  
      "Type": "NAME",  
      "Traits": []  
    },  
    {  
      "Id": 1,  
      "BeginOffset": 94,  
      "EndOffset": 109,  
      "Score": 0.871849775314331,  
      "Text": "100 Main Street, Anytown, USA",  
      "Category": "PROTECTED_HEALTH_INFORMATION",  
      "Type": "ADDRESS",  
      "Traits": []  
    },  
    {  
      "Id": 2,  
      "BeginOffset": 145,  
      "EndOffset": 154,  
      "Score": 0.8302185535430908,  
      "Text": "carpenter",  
      "Category": "PROTECTED_HEALTH_INFORMATION",
```

```
        "Type": "PROFESSION",
        "Traits": []
      }
    ],
    "ModelVersion": "0.0.0"
  }
}
```

Weitere Informationen finden Sie unter [Detect PHI](#) im Amazon Comprehend Medical Developer Guide.

Beispiel 2: So erkennen Sie Protect Health Information (PHI) direkt aus einem Dateipfad

Das folgende detect-phi Beispiel zeigt die erkannten Entitäten mit geschützten Gesundheitsinformationen (PHI) aus einem Dateipfad.

```
aws comprehendmedical detect-phi \  
  --text file://phi.txt
```

Inhalt von phi.txt:

```
"Patient Carlos Salazar presented with a rash on his upper extremities and a dry  
cough. He lives at 100 Main Street, Anytown, USA, where he works from his home as a  
carpenter."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Id": 0,  
      "BeginOffset": 8,  
      "EndOffset": 21,  
      "Score": 0.9914507269859314,  
      "Text": "Carlos Salazar",  
      "Category": "PROTECTED_HEALTH_INFORMATION",  
      "Type": "NAME",  
      "Traits": []  
    },  
    {  
      "Id": 1,  
      "BeginOffset": 94,
```

```

        "EndOffset": 109,
        "Score": 0.871849775314331,
        "Text": "100 Main Street, Anytown, USA",
        "Category": "PROTECTED_HEALTH_INFORMATION",
        "Type": "ADDRESS",
        "Traits": []
    },
    {
        "Id": 2,
        "BeginOffset": 145,
        "EndOffset": 154,
        "Score": 0.8302185535430908,
        "Text": "carpenter",
        "Category": "PROTECTED_HEALTH_INFORMATION",
        "Type": "PROFESSION",
        "Traits": []
    }
],
"ModelVersion": "0.0.0"
}

```

Weitere Informationen finden Sie unter [Detect PHI](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [DetectPhi](#) in der AWS CLI Befehlsreferenz.

infer-icd10-cm

Das folgende Codebeispiel zeigt die Verwendung `infer-icd10-cm`.

AWS CLI

Beispiel 1: Um Entitäten für medizinische Erkrankungen zu erkennen und direkt aus dem Text eine Verknüpfung zur ICD-10-CM-Ontologie herzustellen

Das folgende `infer-icd10-cm` Beispiel kennzeichnet die Entitäten für den erkannten Gesundheitszustand und verknüpft diese Entitäten mit Codes in der Ausgabe 2019 der International Classification of Diseases Clinical Modification (ICD-10-CM).

```

aws comprehendmedical infer-icd10-cm \
  --text "The patient complains of abdominal pain, has a long-standing history of diabetes treated with Micronase daily."

```


Ausgabe:

```
{
  "Entities": [
    {
      "Id": 0,
      "Text": "abdominal pain",
      "Category": "MEDICAL_CONDITION",
      "Type": "DX_NAME",
      "Score": 0.9475538730621338,
      "BeginOffset": 28,
      "EndOffset": 42,
      "Attributes": [],
      "Traits": [
        {
          "Name": "SYMPTOM",
          "Score": 0.6724207401275635
        }
      ],
      "ICD10CMConcepts": [
        {
          "Description": "Unspecified abdominal pain",
          "Code": "R10.9",
          "Score": 0.6904221177101135
        },
        {
          "Description": "Epigastric pain",
          "Code": "R10.13",
          "Score": 0.1364113688468933
        },
        {
          "Description": "Generalized abdominal pain",
          "Code": "R10.84",
          "Score": 0.12508003413677216
        },
        {
          "Description": "Left lower quadrant pain",
          "Code": "R10.32",
          "Score": 0.10063883662223816
        },
        {
          "Description": "Lower abdominal pain, unspecified",
          "Code": "R10.30",
          "Score": 0.09933677315711975
        }
      ]
    }
  ]
}
```

```

    }
  ]
},
{
  "Id": 1,
  "Text": "diabetes",
  "Category": "MEDICAL_CONDITION",
  "Type": "DX_NAME",
  "Score": 0.9899052977561951,
  "BeginOffset": 75,
  "EndOffset": 83,
  "Attributes": [],
  "Traits": [
    {
      "Name": "DIAGNOSIS",
      "Score": 0.9258432388305664
    }
  ],
  "ICD10CMConcepts": [
    {
      "Description": "Type 2 diabetes mellitus without complications",
      "Code": "E11.9",
      "Score": 0.7158446311950684
    },
    {
      "Description": "Family history of diabetes mellitus",
      "Code": "Z83.3",
      "Score": 0.5704703330993652
    },
    {
      "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
      "Code": "Z83.49",
      "Score": 0.19856023788452148
    },
    {
      "Description": "Type 1 diabetes mellitus with ketoacidosis
without coma",
      "Code": "E10.10",
      "Score": 0.13285516202449799
    },
    {
      "Description": "Type 2 diabetes mellitus with hyperglycemia",
      "Code": "E11.65",

```

```

        "Score": 0.0993388369679451
      }
    ]
  },
  "ModelVersion": "0.1.0"
}

```

Weitere Informationen finden Sie unter [Infer ICD10-CM](#) im Amazon Comprehend Medical Developer Guide.

Beispiel 2: Um Entitäten für medizinische Erkrankungen zu erkennen und von einem Dateipfad aus eine Verknüpfung zur ICD-10-CM-Ontologie herzustellen

Das folgende `infer-icd-10-cm` Beispiel kennzeichnet die Entitäten für den erkannten Gesundheitszustand und verknüpft diese Entitäten mit Codes in der Ausgabe 2019 der International Classification of Diseases Clinical Modification (ICD-10-CM).

```

aws comprehendmedical infer-icd10-cm \
  --text file://icd10cm.txt

```

Inhalt von `icd10cm.txt`:

```

{
  "The patient complains of abdominal pain, has a long-standing history of
  diabetes treated with Micronase daily."
}

```

Ausgabe:

```

{
  "Entities": [
    {
      "Id": 0,
      "Text": "abdominal pain",
      "Category": "MEDICAL_CONDITION",
      "Type": "DX_NAME",
      "Score": 0.9475538730621338,
      "BeginOffset": 28,
      "EndOffset": 42,
      "Attributes": [],
      "Traits": [

```

```
        {
          "Name": "SYMPTOM",
          "Score": 0.6724207401275635
        }
      ],
      "ICD10CMConcepts": [
        {
          "Description": "Unspecified abdominal pain",
          "Code": "R10.9",
          "Score": 0.6904221177101135
        },
        {
          "Description": "Epigastric pain",
          "Code": "R10.13",
          "Score": 0.1364113688468933
        },
        {
          "Description": "Generalized abdominal pain",
          "Code": "R10.84",
          "Score": 0.12508003413677216
        },
        {
          "Description": "Left lower quadrant pain",
          "Code": "R10.32",
          "Score": 0.10063883662223816
        },
        {
          "Description": "Lower abdominal pain, unspecified",
          "Code": "R10.30",
          "Score": 0.09933677315711975
        }
      ]
    },
    {
      "Id": 1,
      "Text": "diabetes",
      "Category": "MEDICAL_CONDITION",
      "Type": "DX_NAME",
      "Score": 0.9899052977561951,
      "BeginOffset": 75,
      "EndOffset": 83,
      "Attributes": [],
      "Traits": [
        {
```

```

        "Name": "DIAGNOSIS",
        "Score": 0.9258432388305664
    }
],
"ICD10CMConcepts": [
    {
        "Description": "Type 2 diabetes mellitus without complications",
        "Code": "E11.9",
        "Score": 0.7158446311950684
    },
    {
        "Description": "Family history of diabetes mellitus",
        "Code": "Z83.3",
        "Score": 0.5704703330993652
    },
    {
        "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
        "Code": "Z83.49",
        "Score": 0.19856023788452148
    },
    {
        "Description": "Type 1 diabetes mellitus with ketoacidosis
without coma",
        "Code": "E10.10",
        "Score": 0.13285516202449799
    },
    {
        "Description": "Type 2 diabetes mellitus with hyperglycemia",
        "Code": "E11.65",
        "Score": 0.0993388369679451
    }
]
}
],
"ModelVersion": "0.1.0"
}

```

Weitere Informationen finden Sie unter [ICD1Infer-0-CM](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [InferIcd10Cm](#) in der Befehlsreferenz.AWS CLI

infer-rx-norm

Das folgende Codebeispiel zeigt, wie Sie es verwenden `infer-rx-norm`.

AWS CLI

Beispiel 1: Um Entitäten von Medikamenten zu erkennen und RxNorm direkt aus dem Text zu verlinken

Das folgende `infer-rx-norm` Beispiel zeigt und beschriftet die erkannten Arzneimittelentitäten und verknüpft diese Entitäten mit Konzeptkennungen (RxCUI) aus der Datenbank der National Library of Medicine RxNorm .

```
aws comprehendmedical infer-rx-norm \  
  --text "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but  
denies taking Synthroid."
```

Ausgabe:

```
{  
  "Entities": [  
    {  
      "Id": 0,  
      "Text": "Levothyroxine",  
      "Category": "MEDICATION",  
      "Type": "GENERIC_NAME",  
      "Score": 0.9996285438537598,  
      "BeginOffset": 23,  
      "EndOffset": 36,  
      "Attributes": [  
        {  
          "Type": "DOSAGE",  
          "Score": 0.9892290830612183,  
          "RelationshipScore": 0.9997978806495667,  
          "Id": 1,  
          "BeginOffset": 37,  
          "EndOffset": 51,  
          "Text": "125 micrograms",  
          "Traits": []  
        },  
        {  
          "Type": "ROUTE_OR_MODE",  
          "Score": 0.9988924860954285,
```

```

    "RelationshipScore": 0.998291552066803,
    "Id": 2,
    "BeginOffset": 52,
    "EndOffset": 56,
    "Text": "p.o.",
    "Traits": []
  },
  {
    "Type": "FREQUENCY",
    "Score": 0.9953463673591614,
    "RelationshipScore": 0.9999889135360718,
    "Id": 3,
    "BeginOffset": 57,
    "EndOffset": 67,
    "Text": "once daily",
    "Traits": []
  }
],
"Traits": [],
"RxNormConcepts": [
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
    "Code": "966224",
    "Score": 0.9912070631980896
  },
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
    "Code": "966405",
    "Score": 0.8698278665542603
  },
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
    "Code": "966191",
    "Score": 0.7448257803916931
  },
  {
    "Description": "levothyroxine",
    "Code": "10582",
    "Score": 0.7050482630729675
  },
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxy1]",

```

```

        "Code": "966190",
        "Score": 0.6921631693840027
      }
    ]
  },
  {
    "Id": 4,
    "Text": "Synthroid",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Score": 0.9946461319923401,
    "BeginOffset": 86,
    "EndOffset": 95,
    "Attributes": [],
    "Traits": [
      {
        "Name": "NEGATION",
        "Score": 0.5167351961135864
      }
    ],
    "RxNormConcepts": [
      {
        "Description": "Synthroid",
        "Code": "224920",
        "Score": 0.9462039470672607
      },
      {
        "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
        "Code": "966282",
        "Score": 0.8309829235076904
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
        "Code": "966191",
        "Score": 0.4945160448551178
      },
      {
        "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
[Synthroid]",
        "Code": "966247",
        "Score": 0.3674522042274475
      },
    ]
  },

```



```

        {
            "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
[Synthroid]",
            "Code": "966158",
            "Score": 0.2588822841644287
        }
    ]
}
],
"ModelVersion": "0.0.0"
}

```

Weitere Informationen finden Sie unter [Infer RxNorm](#) im Amazon Comprehend Medical Developer Guide.

Beispiel 2: Um Entitäten für Medikamente zu erkennen und von einem Dateipfad RxNorm aus eine Verknüpfung herzustellen.

Das folgende `infer-rx-norm` Beispiel zeigt und beschriftet die erkannten Arzneimittelentitäten und verknüpft diese Entitäten mit Konzeptkennungen (RxCUI) aus der Datenbank der National Library of Medicine RxNorm .

```

aws comprehendmedical infer-rx-norm \
  --text file://rxnorm.txt

```

Inhalt von `rxnorm.txt`:

```

{
  "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies
  taking Synthroid."
}

```

Ausgabe:

```

{
  "Entities": [
    {
      "Id": 0,
      "Text": "Levothyroxine",
      "Category": "MEDICATION",
      "Type": "GENERIC_NAME",
      "Score": 0.9996285438537598,

```

```
"BeginOffset": 23,
"EndOffset": 36,
"Attributes": [
  {
    "Type": "DOSAGE",
    "Score": 0.9892290830612183,
    "RelationshipScore": 0.9997978806495667,
    "Id": 1,
    "BeginOffset": 37,
    "EndOffset": 51,
    "Text": "125 micrograms",
    "Traits": []
  },
  {
    "Type": "ROUTE_OR_MODE",
    "Score": 0.9988924860954285,
    "RelationshipScore": 0.998291552066803,
    "Id": 2,
    "BeginOffset": 52,
    "EndOffset": 56,
    "Text": "p.o.",
    "Traits": []
  },
  {
    "Type": "FREQUENCY",
    "Score": 0.9953463673591614,
    "RelationshipScore": 0.9999889135360718,
    "Id": 3,
    "BeginOffset": 57,
    "EndOffset": 67,
    "Text": "once daily",
    "Traits": []
  }
],
"Traits": [],
"RxNormConcepts": [
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
    "Code": "966224",
    "Score": 0.9912070631980896
  },
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
    "Code": "966405",
```

```

        "Score": 0.8698278665542603
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
        "Code": "966191",
        "Score": 0.7448257803916931
      },
      {
        "Description": "levothyroxine",
        "Code": "10582",
        "Score": 0.7050482630729675
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxy]",
        "Code": "966190",
        "Score": 0.6921631693840027
      }
    ]
  },
  {
    "Id": 4,
    "Text": "Synthroid",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Score": 0.9946461319923401,
    "BeginOffset": 86,
    "EndOffset": 95,
    "Attributes": [],
    "Traits": [
      {
        "Name": "NEGATION",
        "Score": 0.5167351961135864
      }
    ],
    "RxNormConcepts": [
      {
        "Description": "Synthroid",
        "Code": "224920",
        "Score": 0.9462039470672607
      },
      {

```

```

    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
  [Synthroid]",
    "Code": "966282",
    "Score": 0.8309829235076904
  },
  {
    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
  [Synthroid]",
    "Code": "966191",
    "Score": 0.4945160448551178
  },
  {
    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
  [Synthroid]",
    "Code": "966247",
    "Score": 0.3674522042274475
  },
  {
    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
  [Synthroid]",
    "Code": "966158",
    "Score": 0.2588822841644287
  }
]
}
],
  "ModelVersion": "0.0.0"
}

```

Weitere Informationen finden Sie unter [Infer RxNorm](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [InferRxNorm](#) in AWS CLI der Befehlsreferenz.

infer-snomedct

Das folgende Codebeispiel zeigt die Verwendung `infer-snomedct`.

AWS CLI

Beispiel: Um Entitäten zu erkennen und direkt aus dem Text auf die SNOMED CT-Ontologie zu verlinken

Das folgende `infer-snomedct` Beispiel zeigt, wie medizinische Entitäten erkannt und mit Konzepten aus der Version 2021-03 der Systematized Nomenclature of Medicine, Clinical Terms (CT) verknüpft werden. SNOMED

```
aws comprehendmedical infer-snomedct \
  --text "The patient complains of abdominal pain, has a long-standing history of
  diabetes treated with Micronase daily."
```

Ausgabe:

```
{
  "Entities": [
    {
      "Id": 3,
      "BeginOffset": 26,
      "EndOffset": 40,
      "Score": 0.9598260521888733,
      "Text": "abdominal pain",
      "Category": "MEDICAL_CONDITION",
      "Type": "DX_NAME",
      "Traits": [
        {
          "Name": "SYMPTOM",
          "Score": 0.6819021701812744
        }
      ]
    },
    {
      "Id": 4,
      "BeginOffset": 73,
      "EndOffset": 81,
      "Score": 0.9905840158462524,
      "Text": "diabetes",
      "Category": "MEDICAL_CONDITION",
      "Type": "DX_NAME",
      "Traits": [
        {
          "Name": "DIAGNOSIS",
          "Score": 0.9255214333534241
        }
      ]
    }
  ],
  {
```

```

    "Id": 1,
    "BeginOffset": 95,
    "EndOffset": 104,
    "Score": 0.6371926665306091,
    "Text": "Micronase",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Traits": [],
    "Attributes": [
      {
        "Type": "FREQUENCY",
        "Score": 0.9761165380477905,
        "RelationshipScore": 0.9984188079833984,
        "RelationshipType": "FREQUENCY",
        "Id": 2,
        "BeginOffset": 105,
        "EndOffset": 110,
        "Text": "daily",
        "Category": "MEDICATION",
        "Traits": []
      }
    ]
  }
],
"UnmappedAttributes": [],
"ModelVersion": "1.0.0"
}

```

Weitere Informationen finden Sie unter [Infer SNOMEDCT](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [InferSnomedct](#) in AWS CLI der Befehlsreferenz.

list-entities-detection-v2-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-entities-detection-v2-jobs`.

AWS CLI

Um Jobs zur Erkennung von Entitäten aufzulisten

Das folgende `list-entities-detection-v2-jobs` Beispiel listet aktuelle asynchrone Erkennungsaufträge auf.

aws comprehendmedical list-entities-detection-v2-jobs

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "ab9887877365fe70299089371c043b96",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
      "EndTime": "2020-03-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-07-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": ""
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "DetectEntitiesModelV20190930"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [ListEntitiesDetectionV2Jobs](#) in der Befehlsreferenz.AWS CLI

list-icd10-cm-inference-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt. `list-icd10-cm-inference-jobs`

AWS CLI

Um alle aktuellen ICD -10-CM-Inferenzjobs aufzulisten

Das folgende Beispiel zeigt, wie der `list-icd10-cm-inference-jobs` Vorgang eine Liste aktueller asynchroner ICD -10-CM-Batchinferenzjobs zurückgibt.

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "5780034166536cdb52ffa3295a1b00a7",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
      "EndTime": "2020-05-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-09-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "0.1.0"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [ListIcd10 CmlInferenceJobs](#) in AWS CLI der Befehlsreferenz.

list-phi-detection-jobs

Das folgende Codebeispiel zeigt, wie man `list-phi-detection-jobs`.

AWS CLI

Um Jobs zur Erkennung geschützter Gesundheitsinformationen (PHI) aufzulisten

Im folgenden `list-phi-detection-jobs` Beispiel werden aktuelle Aufträge zur Erkennung geschützter Gesundheitsinformationen (PHI) aufgeführt

```
aws comprehendmedical list-phi-detection-jobs
```

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "4750034166536cdb52ffa3295a1b00a3",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
      "EndTime": "2020-03-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-07-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": ""
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "867139942017-
PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "PHIModelV20190903"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [ListPhiDetectionJobs](#) in der AWS CLI Befehlsreferenz.

list-rx-norm-inference-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-rx-norm-inference-jobs`.

AWS CLI

Um alle aktuellen Rx-Norm-Inferenzjobs aufzulisten

Das folgende Beispiel zeigt, wie eine Liste aktueller asynchroner Rx-Norm-Batchinferenzjobs `list-rx-norm-inference-jobs` zurückgegeben wird.

```
aws comprehendmedical list-rx-norm-inference-jobs
```

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "4980034166536cfb52gga3295a1b00a3",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
      "EndTime": "2020-05-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-09-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "0.0.0"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [ListRxNormInferenceJobs](#) in AWS CLI der Befehlsreferenz.

list-snomedct-inference-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-snomedct-inference-jobs`.

AWS CLI

Um alle SNOMED CT-Inferenzjobs aufzulisten

Das folgende Beispiel zeigt, wie der `list-snomedct-inference-jobs` Vorgang eine Liste der aktuellen asynchronen SNOMED CT-Batch-Inferenzjobs zurückgibt.

```
aws comprehendmedical list-snomedct-inference-jobs
```

Ausgabe:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "5780034166536cdb52ffa3295a1b00a7",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
      "EndTime": "2020-05-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-09-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "0.1.0"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [ListSnomedctInferenceJobs](#) in AWS CLI der Befehlsreferenz.

start-entities-detection-v2-job

Das folgende Codebeispiel zeigt die Verwendung `start-entities-detection-v2-job`.

AWS CLI

Um einen Job zur Erkennung von Entitäten zu starten

Im folgenden `start-entities-detection-v2-job` Beispiel wird ein asynchroner Auftrag zur Erkennung von Entitäten gestartet.

```
aws comprehendmedical start-entities-detection-v2-job \  
  --input-data-config "S3Bucket=comp-med-input" \  
  --output-data-config "S3Bucket=comp-med-output" \  
  --data-access-role-arn arn:aws:iam::867139942017:role/  
  ComprehendMedicalBatchProcessingRole \  
  --language-code en
```

Ausgabe:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [StartEntitiesDetectionV2Job](#) in der Befehlsreferenz. AWS CLI

start-icd10-cm-inference-job

Das folgende Codebeispiel zeigt, wie man es benutzt. `start-icd10-cm-inference-job`

AWS CLI

Um einen ICD -10-CM-Inferenzjob zu starten

Im folgenden `start-icd10-cm-inference-job` Beispiel wird ein Batch-Analyseauftrag für Inferenzen mit einer Länge von ICD -10 CM gestartet.

```
aws comprehendmedical start-icd10-cm-inference-job \  

```

```
--input-data-config "S3Bucket=comp-med-input" \  
--output-data-config "S3Bucket=comp-med-output" \  
--data-access-role-arn arn:aws:iam::867139942017:role/  
ComprehendMedicalBatchProcessingRole \  
--language-code en
```

Ausgabe:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96"  
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [StartIcd10 CmlInferenceJob](#) in AWS CLI der Befehlsreferenz.

start-phi-detection-job

Das folgende Codebeispiel zeigt, wie man `start-phi-detection-job`.

AWS CLI

Um einen PHI Erkennungsauftrag zu starten

Im folgenden `start-phi-detection-job` Beispiel wird ein asynchroner PHI Entitätserkennungsauftrag gestartet.

```
aws comprehendmedical start-phi-detection-job \  
--input-data-config "S3Bucket=comp-med-input" \  
--output-data-config "S3Bucket=comp-med-output" \  
--data-access-role-arn arn:aws:iam::867139942017:role/  
ComprehendMedicalBatchProcessingRole \  
--language-code en
```

Ausgabe:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [StartPhiDetectionJob](#) in der AWS CLI Befehlsreferenz.

start-rx-norm-inference-job

Das folgende Codebeispiel zeigt die Verwendung `start-rx-norm-inference-job`.

AWS CLI

Um einen RxNorm Inferenzjob zu starten

Im folgenden `start-rx-norm-inference-job` Beispiel wird ein Auftrag zur Batch-Analyse von RxNorm Inferenzen gestartet.

```
aws comprehendmedical start-rx-norm-inference-job \
  --input-data-config "S3Bucket=comp-med-input" \
  --output-data-config "S3Bucket=comp-med-output" \
  --data-access-role-arn arn:aws:iam::867139942017:role/
  ComprehendMedicalBatchProcessingRole \
  --language-code en
```

Ausgabe:

```
{
  "JobId": "eg8199877365fc70299089371c043b96"
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [StartRxNormInferenceJob](#) in der AWS CLI Befehlsreferenz.

start-snomedct-inference-job

Das folgende Codebeispiel zeigt die Verwendung `start-snomedct-inference-job`.

AWS CLI

Um einen SNOMED CT-Inferenzjob zu starten

Im folgenden `start-snomedct-inference-job` Beispiel wird ein Batch-Analyseauftrag für SNOMED CT-Inferenzen gestartet.

```
aws comprehendmedical start-snomedct-inference-job \  
  --input-data-config "S3Bucket=comp-med-input" \  
  --output-data-config "S3Bucket=comp-med-output" \  
  --data-access-role-arn arn:aws:iam::867139942017:role/  
  ComprehendMedicalBatchProcessingRole \  
  --language-code en
```

Ausgabe:

```
{  
  "JobId": "dg7289877365fc70299089371c043b96"  
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [StartSnomedctInferenceJob](#) in AWS CLI der Befehlsreferenz.

stop-entities-detection-v2-job

Das folgende Codebeispiel zeigt die Verwendung `stop-entities-detection-v2-job`.

AWS CLI

Um einen Job zur Entitätserkennung zu beenden

Im folgenden `stop-entities-detection-v2-job` Beispiel wird ein asynchroner Entitätserkennungsauftrag beendet.

```
aws comprehendmedical stop-entities-detection-v2-job \  
  --job-id "ab9887877365fe70299089371c043b96"
```

Ausgabe:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [StopEntitiesDetectionV2Job](#) in der Befehlsreferenz AWS CLI

stop-icd10-cm-inference-job

Das folgende Codebeispiel zeigt, wie man es benutzt. stop-icd10-cm-inference-job

AWS CLI

Um einen ICD-10-CM-Inferenzjob zu beenden

Im folgenden stop-icd10-cm-inference-job Beispiel wird ein Batch-Analyseauftrag mit einer ICD Inferenz von -10 CM beendet.

```
aws comprehendmedical stop-icd10-cm-inference-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Ausgabe:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96",  
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie unter [StopIcd10 CmlInferenceJob](#) in AWS CLI der Befehlsreferenz.

stop-phi-detection-job

Das folgende Codebeispiel zeigt, wie man stop-phi-detection-job.

AWS CLI

Um einen Job zur Erkennung geschützter Gesundheitsinformationen (PHI) zu beenden

Im folgenden stop-phi-detection-job Beispiel wird ein asynchroner Erkennungsauftrag für geschützte Gesundheitsinformationen (PHI) beendet.

```
aws comprehendmedical stop-phi-detection-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```



```
--job-id "4750034166536cdb52ffa3295a1b00a3"
```

Ausgabe:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Weitere Informationen finden Sie unter [Batch APIs](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [StopPhiDetectionJob](#) in der AWS CLI Befehlsreferenz.

stop-rx-norm-inference-job

Das folgende Codebeispiel zeigt die Verwendung `stop-rx-norm-inference-job`.

AWS CLI

Um einen RxNorm Inferenzjob zu beenden

Im folgenden `stop-rx-norm-inference-job` Beispiel wird ein ICD Batch-Analyseauftrag für Inferenzen mit einem Wert von -10 CM beendet.

```
aws comprehendmedical stop-rx-norm-inference-job \  
  --job-id "eg8199877365fc70299089371c043b96"
```

Ausgabe:

```
{  
  "JobId": "eg8199877365fc70299089371c043b96",  
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [StopRxNormInferenceJob](#) in der AWS CLI Befehlsreferenz.

stop-snomedct-inference-job

Das folgende Codebeispiel zeigt die Verwendung `stop-snomedct-inference-job`.

AWS CLI

Um einen SNOMED CT-Inferenzjob zu beenden

Im folgenden `stop-snomedct-inference-job` Beispiel wird ein Auftrag zur Batch-Analyse von SNOMED CT-Inferenzen beendet.

```
aws comprehendmedical stop-snomedct-inference-job \  
  --job-id "8750034166436cdb52ffa3295a1b00a1"
```

Ausgabe:

```
{  
  "JobId": "8750034166436cdb52ffa3295a1b00a1",  
}
```

Weitere Informationen finden Sie unter [Ontology Linking Batch Analysis](#) im Amazon Comprehend Medical Developer Guide.

- API-Einheiten finden Sie [StopSnomedctInferenceJob](#) in AWS CLI der Befehlsreferenz.

AWS Config Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Config.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-config-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-config-rule`.

AWS CLI

Um eine AWS Config-Regel zu löschen

Der folgende Befehl löscht eine AWS Config-Regel mit dem Namen `MyConfigRule`:

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- API-Einheiten finden Sie [DeleteConfigRule](#) in der AWS CLI Befehlsreferenz.

delete-delivery-channel

Das folgende Codebeispiel zeigt die Verwendung `delete-delivery-channel`.

AWS CLI

Um einen Lieferkanal zu löschen

Der folgende Befehl löscht den Standardzustellungschannel:

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- API-Einheiten finden Sie [DeleteDeliveryChannel](#) in der AWS CLI Befehlsreferenz.

delete-evaluation-results

Das folgende Codebeispiel zeigt die Verwendung `delete-evaluation-results`.

AWS CLI

Um Bewertungsergebnisse manuell zu löschen

Der folgende Befehl löscht die aktuellen Auswertungsergebnisse für die AWS verwaltete Regel `s3-bucket-versioning-enabled`

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- API-Einheiten finden Sie [DeleteEvaluationResults](#) in der AWS CLI Befehlsreferenz.

deliver-config-snapshot

Das folgende Codebeispiel zeigt die Verwendung `deliver-config-snapshot`.

AWS CLI

Um einen Konfigurations-Snapshot bereitzustellen

Der folgende Befehl übermittelt einen Konfigurations-Snapshot an den Amazon S3 S3-Bucket, der zum Standard-Lieferkanal gehört:

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

Ausgabe:

```
{
  "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- API-Einheiten finden Sie [DeliverConfigSnapshot](#) unter AWS CLI Befehlsreferenz.

describe-compliance-by-config-rule

Das folgende Codebeispiel zeigt die Verwendung `describe-compliance-by-config-rule`.

AWS CLI

Um Compliance-Informationen für Ihre AWS Config-Regeln zu erhalten

Der folgende Befehl gibt Konformitätsinformationen für jede AWS Config-Regel zurück, gegen die eine oder mehrere AWS Ressourcen verstoßen:

```
aws configservice describe-compliance-by-config-rule --compliance-  
types NON_COMPLIANT
```

In der Ausgabe gibt der Wert für jedes `CappedCount` Attribut an, wie viele Ressourcen der zugehörigen Regel nicht entsprechen. Die folgende Ausgabe gibt beispielsweise an, dass 3 Ressourcen der genannten Regel nicht entsprechen `InstanceTypesAreT2micro`.

Ausgabe:

```

{
  "ComplianceByConfigRules": [
    {
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 3,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      },
      "ConfigRuleName": "InstanceTypesAreT2micro"
    },
    {
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 10,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      },
      "ConfigRuleName": "RequiredTagsForVolumes"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeComplianceByConfigRule](#) unter AWS CLI Befehlsreferenz.

describe-compliance-by-resource

Das folgende Codebeispiel zeigt die Verwendung `describe-compliance-by-resource`.

AWS CLI

Um Compliance-Informationen für Ihre AWS Ressourcen zu erhalten

Der folgende Befehl gibt Konformitätsinformationen für jede EC2 Instanz zurück, die von AWS Config aufgezeichnet wurde und gegen eine oder mehrere Regeln verstößt:

```

aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
--compliance-types NON_COMPLIANT

```

In der Ausgabe gibt der Wert für jedes CappedCount Attribut an, gegen wie viele Regeln die Ressource verstößt. Die folgende Ausgabe gibt beispielsweise an, dass die Instanz `i-1a2b3c4d` gegen zwei Regeln verstößt.

Ausgabe:

```
{
  "ComplianceByResources": [
    {
      "ResourceType": "AWS::EC2::Instance",
      "ResourceId": "i-1a2b3c4d",
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 2,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      }
    },
    {
      "ResourceType": "AWS::EC2::Instance",
      "ResourceId": "i-2a2b3c4d ",
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 3,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      }
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeComplianceByResource](#) in der AWS CLI Befehlsreferenz.

describe-config-rule-evaluation-status

Das folgende Codebeispiel zeigt die Verwendung `describe-config-rule-evaluation-status`.

AWS CLI

Um Statusinformationen für eine AWS Config-Regel abzurufen

Der folgende Befehl gibt die Statusinformationen für eine AWS Config-Regel mit dem Namen zurückMyConfigRule:

```
aws configservice describe-config-rule-evaluation-status --config-rule-names MyConfigRule
```

Ausgabe:

```
{
  "ConfigRulesEvaluationStatus": [
    {
      "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/config-rule-abcdef",
      "FirstActivatedTime": 1450311703.844,
      "ConfigRuleId": "config-rule-abcdef",
      "LastSuccessfulInvocationTime": 1450314643.156,
      "ConfigRuleName": "MyConfigRule"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeConfigRuleEvaluationStatus](#) in der AWS CLI Befehlsreferenz.

describe-config-rules

Das folgende Codebeispiel zeigt die Verwendung `describe-config-rules`.

AWS CLI

Um Details für eine AWS Config-Regel abzurufen

Der folgende Befehl gibt Details für eine AWS Config-Regel mit dem Namen zurückInstanceTypesAreT2micro:

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

Ausgabe:

```
{
```

```

    "ConfigRules": [
      {
        "ConfigRuleState": "ACTIVE",
        "Description": "Evaluates whether EC2 instances are the t2.micro type.",
        "ConfigRuleName": "InstanceTypesAreT2micro",
        "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
        "Source": {
          "Owner": "CUSTOM_LAMBDA",
          "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
          "SourceDetails": [
            {
              "EventSource": "aws.config",
              "MessageType": "ConfigurationItemChangeNotification"
            }
          ]
        },
        "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}",
        "Scope": {
          "ComplianceResourceTypes": [
            "AWS::EC2::Instance"
          ]
        },
        "ConfigRuleId": "config-rule-abcdef"
      }
    ]
  }
}

```

- API-Einheiten finden Sie [DescribeConfigRules](#) in der AWS CLI Befehlsreferenz.

describe-configuration-recorder-status

Das folgende Codebeispiel zeigt die Verwendung `describe-configuration-recorder-status`.

AWS CLI

Um Statusinformationen für den Konfigurationsrekorder abzurufen

Der folgende Befehl gibt den Status des Standardkonfigurationsrekorders zurück:

```
aws configservice describe-configuration-recorder-status
```


Ausgabe:

```
{
  "ConfigurationRecordersStatus": [
    {
      "name": "default",
      "lastStatus": "SUCCESS",
      "recording": true,
      "lastStatusChangeTime": 1452193834.344,
      "lastStartTime": 1441039997.819,
      "lastStopTime": 1441039992.835
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeConfigurationRecorderStatus](#) in der AWS CLI Befehlsreferenz.

describe-configuration-recorders

Das folgende Codebeispiel zeigt die Verwendung `describe-configuration-recorders`.

AWS CLI

Um Details zum Konfigurationsrekorder zu erhalten

Der folgende Befehl gibt Details zum Standardkonfigurationsrekorder zurück:

```
aws configservice describe-configuration-recorders
```

Ausgabe:

```
{
  "ConfigurationRecorders": [
    {
      "recordingGroup": {
        "allSupported": true,
        "resourceTypes": [],
        "includeGlobalResourceTypes": true
      },
      "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-A1B2C3D4E5F6",
    }
  ]
}
```

```
        "name": "default"
      }
    ]
  }
```

- API-Einheiten finden Sie [DescribeConfigurationRecorders](#) in der AWS CLI Befehlsreferenz.

describe-delivery-channel-status

Das folgende Codebeispiel zeigt die Verwendung `describe-delivery-channel-status`.

AWS CLI

Um Statusinformationen für den Lieferkanal abzurufen

Der folgende Befehl gibt den Status des Lieferkanals zurück:

```
aws configservice describe-delivery-channel-status
```

Ausgabe:

```
{
  "DeliveryChannelsStatus": [
    {
      "configStreamDeliveryInfo": {
        "lastStatusChangeTime": 1452193834.381,
        "lastStatus": "SUCCESS"
      },
      "configHistoryDeliveryInfo": {
        "lastSuccessfulTime": 1450317838.412,
        "lastStatus": "SUCCESS",
        "lastAttemptTime": 1450317838.412
      },
      "configSnapshotDeliveryInfo": {
        "lastSuccessfulTime": 1452185597.094,
        "lastStatus": "SUCCESS",
        "lastAttemptTime": 1452185597.094
      },
      "name": "default"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeDeliveryChannelStatus](#) in der AWS CLI Befehlsreferenz.

describe-delivery-channels

Das folgende Codebeispiel zeigt die Verwendung `describe-delivery-channels`.

AWS CLI

Um Details zum Lieferkanal zu erhalten

Der folgende Befehl gibt Details zum Lieferkanal zurück:

```
aws configservice describe-delivery-channels
```

Ausgabe:

```
{
  "DeliveryChannels": [
    {
      "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
      "name": "default",
      "s3BucketName": "config-bucket-123456789012"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeDeliveryChannels](#) in der AWS CLI Befehlsreferenz.

get-compliance-details-by-config-rule

Das folgende Codebeispiel zeigt die Verwendung `get-compliance-details-by-config-rule`.

AWS CLI

Um die Auswertungsergebnisse für eine AWS Config-Regel abzurufen

Der folgende Befehl gibt die Auswertungsergebnisse für alle Ressourcen zurück, die nicht einer AWS Config-Regel mit dem Namen entsprechen `InstanceTypesAreT2micro`:

```
aws configservice get-compliance-details-by-config-rule --config-rule-
name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```

Ausgabe:

```
{
  "EvaluationResults": [
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-1a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314645.261,
      "ConfigRuleInvokedTime": 1450314642.948,
      "ComplianceType": "NON_COMPLIANT"
    },
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-2a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314645.18,
      "ConfigRuleInvokedTime": 1450314642.902,
      "ComplianceType": "NON_COMPLIANT"
    },
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-3a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314643.346,
      "ConfigRuleInvokedTime": 1450314643.124,
      "ComplianceType": "NON_COMPLIANT"
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie [GetComplianceDetailsByConfigRule](#) in der AWS CLI Befehlsreferenz.

get-compliance-details-by-resource

Das folgende Codebeispiel zeigt die Verwendung `get-compliance-details-by-resource`.

AWS CLI

Um die Evaluierungsergebnisse für eine AWS Ressource abzurufen

Der folgende Befehl gibt die Evaluierungsergebnisse für jede Regel zurück, der die EC2 Instanz `i-1a2b3c4d` nicht entspricht:

```
aws configservice get-compliance-details-by-resource --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

Ausgabe:

```
{
  "EvaluationResults": [
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-1a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314643.288,
      "ConfigRuleInvokedTime": 1450314643.034,
      "ComplianceType": "NON_COMPLIANT"
    },
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
```

```
        "ResourceId": "i-1a2b3c4d",
        "ConfigRuleName": "RequiredTagForEC2Instances"
    }
},
"ResultRecordedTime": 1450314645.261,
"ConfigRuleInvokedTime": 1450314642.948,
"ComplianceType": "NON_COMPLIANT"
}
]
}
```

- API-Einheiten finden Sie [GetComplianceDetailsByResource](#) unter AWS CLI Befehlsreferenz.

get-compliance-summary-by-config-rule

Das folgende Codebeispiel zeigt die Verwendung `get-compliance-summary-by-config-rule`.

AWS CLI

Um die Konformitätsübersicht für Ihre AWS Config-Regeln abzurufen

Der folgende Befehl gibt die Anzahl der Regeln zurück, die konform sind, und die Anzahl der nicht konformen Regeln:

```
aws configservice get-compliance-summary-by-config-rule
```

In der Ausgabe gibt der Wert für jedes `CappedCount` Attribut an, wie viele Regeln konform oder nicht konform sind.

Ausgabe:

```
{
  "ComplianceSummary": {
    "NonCompliantResourceCount": {
      "CappedCount": 3,
      "CapExceeded": false
    },
    "ComplianceSummaryTimestamp": 1452204131.493,
    "CompliantResourceCount": {
      "CappedCount": 2,
      "CapExceeded": false
    }
  }
}
```

```
}  
  }  
}
```

- API-Einheiten finden Sie unter [GetComplianceSummaryByConfigRule AWS CLI-Befehlsreferenz](#).

get-compliance-summary-by-resource-type

Das folgende Codebeispiel zeigt die Verwendung `get-compliance-summary-by-resource-type`.

AWS CLI

Um die Konformitätsübersicht für alle Ressourcentypen abzurufen

Der folgende Befehl gibt die Anzahl der AWS Ressourcen zurück, die nicht konform sind, und die Anzahl, die konform sind:

```
aws configservice get-compliance-summary-by-resource-type
```

In der Ausgabe gibt der Wert für jedes `CappedCount` Attribut an, wie viele Ressourcen konform oder nicht konform sind.

Ausgabe:

```
{  
  "ComplianceSummariesByResourceType": [  
    {  
      "ComplianceSummary": {  
        "NonCompliantResourceCount": {  
          "CappedCount": 16,  
          "CapExceeded": false  
        },  
        "ComplianceSummaryTimestamp": 1453237464.543,  
        "CompliantResourceCount": {  
          "CappedCount": 10,  
          "CapExceeded": false  
        }  
      }  
    }  
  ]  
}
```

```
]
}
```

Um die Konformitätsübersicht für einen bestimmten Ressourcentyp abzurufen

Der folgende Befehl gibt die Anzahl der EC2 Instanzen zurück, die nicht konform sind, und die Anzahl, die konform sind:

```
aws configservice get-compliance-summary-by-resource-type --resource-
types AWS::EC2::Instance
```

In der Ausgabe gibt der Wert für jedes CappedCount Attribut an, wie viele Ressourcen konform oder nicht konform sind.

Ausgabe:

```
{
  "ComplianceSummariesByResourceType": [
    {
      "ResourceType": "AWS::EC2::Instance",
      "ComplianceSummary": {
        "NonCompliantResourceCount": {
          "CappedCount": 3,
          "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1452204923.518,
        "CompliantResourceCount": {
          "CappedCount": 7,
          "CapExceeded": false
        }
      }
    }
  ]
}
```

- APIEinzelheiten finden Sie unter [GetComplianceSummaryByResourceType AWS CLIBefehlsreferenz](#).

get-resource-config-history

Das folgende Codebeispiel zeigt die Verwendung `get-resource-config-history`.

AWS CLI

Um den Konfigurationsverlauf einer AWS Ressource abzurufen

Der folgende Befehl gibt eine Liste von Konfigurationselementen für eine EC2 Instanz mit der ID zurück `i-1a2b3c4d`:

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d
```

- API-Einheiten finden Sie [GetResourceConfigHistory](#) in der AWS CLI Befehlsreferenz.

get-status

Das folgende Codebeispiel zeigt die Verwendung `get-status`.

AWS CLI

Um den Status für AWS Config abzurufen

Der folgende Befehl gibt den Status des Lieferkanals und des Konfigurationsrekorders zurück:

```
aws configservice get-status
```

Ausgabe:

```
Configuration Recorders:  
  
name: default  
recorder: ON  
last status: SUCCESS  
  
Delivery Channels:  
  
name: default  
last stream delivery status: SUCCESS  
last history delivery status: SUCCESS  
last snapshot delivery status: SUCCESS
```

- API-Einheiten finden Sie [GetStatus](#) in der AWS CLI Befehlsreferenz.

list-discovered-resources

Das folgende Codebeispiel zeigt die Verwendung `list-discovered-resources`.

AWS CLI

Um Ressourcen aufzulisten, die AWS Config entdeckt hat

Der folgende Befehl listet die EC2 Instanzen auf, die AWS Config entdeckt hat:

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

Ausgabe:

```
{
  "resourceIdentifiers": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-1a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-2a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-3a2b3c4d"
    }
  ]
}
```

- API-Einheiten finden Sie [ListDiscoveredResources](#) in der AWS CLI Befehlsreferenz.

put-config-rule

Das folgende Codebeispiel zeigt die Verwendung `put-config-rule`.

AWS CLI

So fügen Sie eine AWS verwaltete Konfigurationsregel hinzu

Der folgende Befehl enthält JSON Code zum Hinzufügen einer AWS verwalteten Konfigurationsregel:

```
aws configservice put-config-rule --config-rule file://  
RequiredTagsForEC2Instances.json
```

`RequiredTagsForEC2Instances.json` ist eine JSON Datei, die die Regelkonfiguration enthält:

```
{  
  "ConfigRuleName": "RequiredTagsForEC2Instances",  
  "Description": "Checks whether the CostCenter and Owner tags are applied to EC2  
instances.",  
  "Scope": {  
    "ComplianceResourceTypes": [  
      "AWS::EC2::Instance"  
    ]  
  },  
  "Source": {  
    "Owner": "AWS",  
    "SourceIdentifier": "REQUIRED_TAGS"  
  },  
  "InputParameters": "{\"tag1Key\":\"CostCenter\",\"tag2Key\":\"Owner\"}"  
}
```

Für das `ComplianceResourceTypes` Attribut beschränkt dieser JSON Code den Bereich auf Ressourcen des `AWS::EC2::Instance` Typs, sodass AWS Config nur EC2 Instanzen anhand der Regel auswertet. Da es sich bei der Regel um eine verwaltete Regel handelt, ist das `Owner` Attribut auf `festgelegtAWS`, und das `SourceIdentifier` Attribut ist auf den Regelbezeichner, `festgelegtREQUIRED_TAGS`. Für das `InputParameters` Attribut werden die Tag-Schlüssel, die die Regel benötigt `Owner`, `CostCenter` und, angegeben.

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Um die Regelkonfiguration zu überprüfen, führen Sie den `describe-config-rules` Befehl aus und geben Sie den Regelnamen an.

So fügen Sie eine vom Kunden verwaltete Konfigurationsregel hinzu

Der folgende Befehl enthält JSON Code zum Hinzufügen einer vom Kunden verwalteten Konfigurationsregel:

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

`InstanceTypesAreT2micro.json` ist eine JSON Datei, die die Regelkonfiguration enthält:

```
{
  "ConfigRuleName": "InstanceTypesAreT2micro",
  "Description": "Evaluates whether EC2 instances are the t2.micro type.",
  "Scope": {
    "ComplianceResourceTypes": [
      "AWS::EC2::Instance"
    ]
  },
  "Source": {
    "Owner": "CUSTOM_LAMBDA",
    "SourceIdentifier": "arn:aws:lambda:us-east-1:123456789012:function:InstanceTypeCheck",
    "SourceDetails": [
      {
        "EventSource": "aws.config",
        "MessageType": "ConfigurationItemChangeNotification"
      }
    ]
  },
  "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}"
}
```

Für das `ComplianceResourceTypes` Attribut beschränkt dieser JSON Code den Bereich auf Ressourcen des `AWS::EC2::Instance` Typs, sodass AWS Config nur EC2 Instanzen anhand der Regel auswertet. Da es sich bei dieser Regel um eine vom Kunden verwaltete Regel handelt, ist das `CUSTOM_LAMBDA` Owner Attribut auf und das `SourceIdentifier` Attribut auf das ARN der AWS Lambda-Funktion gesetzt. Das `SourceDetails` Objekt ist erforderlich. Die für das `InputParameters` Attribut angegebenen Parameter werden an die AWS Lambda-Funktion übergeben, wenn AWS Config sie aufruft, um Ressourcen anhand der Regel auszuwerten.

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Um die Regelkonfiguration zu überprüfen, führen Sie den `describe-config-rules` Befehl aus und geben Sie den Regelnamen an.

- API-Einheiten finden Sie [PutConfigRule](#) unter AWS CLI Befehlsreferenz.

put-configuration-recorder

Das folgende Codebeispiel zeigt die Verwendung `put-configuration-recorder`.

AWS CLI

Beispiel 1: Um alle unterstützten Ressourcen aufzuzeichnen

Der folgende Befehl erstellt einen Konfigurationsrekorder, der Änderungen an allen unterstützten Ressourcentypen, einschließlich globaler Ressourcentypen, verfolgt:

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group allSupported=true,includeGlobalResourceTypes=true
```

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Führen Sie den describe-configuration-recorders Befehl aus, um die Einstellungen Ihres Konfigurationsrekorders zu überprüfen.

Beispiel 2: Um bestimmte Arten von Ressourcen aufzuzeichnen

Der folgende Befehl erstellt einen Konfigurationsrekorder, der Änderungen nur an den Ressourcentypen verfolgt, die in der JSON Datei für die Option --recording-group angegeben sind:

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group file://recordingGroup.json
```

recordingGroup.json ist eine JSON Datei, die die Arten von Ressourcen angibt, die AWS Config aufzeichnet:

```
{  
  "allSupported": false,  
  "includeGlobalResourceTypes": false,  
  "resourceTypes": [  
    "AWS::EC2::EIP",  
    "AWS::EC2::Instance",  
    "AWS::EC2::NetworkAcl",  
    "AWS::EC2::SecurityGroup",  
    "AWS::CloudTrail::Trail",  
    "AWS::EC2::Volume",  
    "AWS::EC2::VPC",  
    "AWS::IAM::User",
```

```

    "AWS::IAM::Policy"
  ]
}

```

Bevor Sie Ressourcentypen für den `resourceTypes` Schlüssel angeben können, müssen Sie die Optionen `allSupported` und `includeGlobalResourceTypes` auf `False` setzen oder sie weglassen.

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Führen Sie den `describe-configuration-records` Befehl aus, um die Einstellungen Ihres Konfigurationsrekorders zu überprüfen.

Beispiel 3: Um alle unterstützten Ressourcen mit Ausnahme bestimmter Ressourcentypen auszuwählen

Der folgende Befehl erstellt einen Konfigurationsrekorder, der Änderungen an allen aktuellen und future unterstützten Ressourcentypen verfolgt, mit Ausnahme der Ressourcentypen, die in der JSON Datei für die Option `--recording-group` angegeben sind:

```

aws configservice put-configuration-recorder \
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/
config-role \
  --recording-group file://recordingGroup.json

```

`recordingGroup.json` ist eine JSON Datei, die die Arten von Ressourcen angibt, die AWS Config aufzeichnet:

```

{
  "allSupported": false,
  "exclusionByResourceTypes": {
    "resourceTypes": [
      "AWS::Redshift::ClusterSnapshot",
      "AWS::RDS::DBClusterSnapshot",
      "AWS::CloudFront::StreamingDistribution"
    ]
  },
  "includeGlobalResourceTypes": false,
  "recordingStrategy": {
    "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"
  },
}

```

Bevor Sie Ressourcentypen angeben können, die von der Aufzeichnung ausgeschlossen werden sollen: 1) Sie müssen die Optionen `allSupported` und `includeGlobalResourceTypes` auf Falsch setzen oder sie weglassen, und 2) Sie müssen das `useOnly` Feld von `RecordingStrategy` auf `EXCLUSION_RESOURCE_BY_` setzen. `TYPES`

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Führen Sie den `describe-configuration-records` Befehl aus, um die Einstellungen Ihres Konfigurationsrekorders zu überprüfen.

- API-Einheiten finden Sie [PutConfigurationRecorder](#) in der AWS CLI Befehlsreferenz.

put-delivery-channel

Das folgende Codebeispiel zeigt die Verwendung `put-delivery-channel`.

AWS CLI

Um einen Lieferkanal zu erstellen

Der folgende Befehl stellt die Einstellungen für den Lieferkanal als JSON Code bereit:

```
aws configservice put-delivery-channel --delivery-channel file://  
deliveryChannel.json
```

Die `deliveryChannel.json` Datei spezifiziert die Attribute des Lieferkanals:

```
{  
  "name": "default",  
  "s3BucketName": "config-bucket-123456789012",  
  "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",  
  "configSnapshotDeliveryProperties": {  
    "deliveryFrequency": "Twelve_Hours"  
  }  
}
```

In diesem Beispiel werden die folgenden Attribute festgelegt:

`name`- Der Name des Lieferkanals. Standardmäßig weist AWS Config den Namen einem neuen Lieferkanal `default` zu. Sie können den Namen des Lieferkanals nicht mit dem Befehl `put-delivery-channel` aktualisieren. Die Schritte zum Ändern des Namens finden Sie unter

Umbenennen des Lieferkanals. `s3BucketName` - Der Name des Amazon S3 S3-Buckets, für den AWS Config Konfigurations-Snapshots und Konfigurationsverlaufsdateien bereitstellt. Wenn Sie einen Bucket angeben, der zu einem anderen AWS Konto gehört, muss dieser Bucket über Richtlinien verfügen, die Config Zugriffsberechtigungen gewähren. AWS Weitere Informationen finden Sie unter Berechtigungen für den Amazon-S3-Bucket.

`snsTopicARN`— Der Amazon-Ressourcenname (ARN) des SNS Amazon-Themas, an das AWS Config Benachrichtigungen über Konfigurationsänderungen sendet. Wenn Sie ein Thema aus einem anderen Konto auswählen, muss das Thema Richtlinien enthalten, die Config Zugriffsberechtigungen gewähren. AWS Weitere Informationen finden Sie unter Berechtigungen für das SNS Amazon-Thema.

`configSnapshotDeliveryProperties`- Enthält das `deliveryFrequency` Attribut, das festlegt, wie oft AWS Config Konfigurations-Snapshots liefert und wie oft es Evaluierungen für periodische Config-Regeln aufruft.

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Führen Sie den `describe-delivery-channels` Befehl aus, um die Einstellungen Ihres Lieferkanals zu überprüfen.

- APIEinzelheiten finden Sie [PutDeliveryChannel](#) in der AWS CLI Befehlsreferenz.

start-config-rules-evaluation

Das folgende Codebeispiel zeigt die Verwendung `start-config-rules-evaluation`.

AWS CLI

So führen Sie eine On-Demand-Evaluierung für AWS Config-Regeln durch

Mit dem folgenden Befehl wird eine Evaluierung für zwei AWS verwaltete Regeln gestartet:

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-versioning-enabled cloudtrail-enabled
```

- APIEinzelheiten finden Sie [StartConfigRulesEvaluation](#) in der AWS CLI Befehlsreferenz.

start-configuration-recorder

Das folgende Codebeispiel zeigt die Verwendung `start-configuration-recorder`.

AWS CLI

Um den Konfigurationsrekorder zu starten

Der folgende Befehl startet den Standard-Konfigurationsrekorder:

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Führen Sie den Befehl `get-status` aus, um zu überprüfen, ob AWS Config Ihre Ressourcen aufzeichnet.

- API-Einheiten finden Sie [StartConfigurationRecorder](#) in der AWS CLI Befehlsreferenz.

stop-configuration-recorder

Das folgende Codebeispiel zeigt die Verwendung `stop-configuration-recorder`.

AWS CLI

Um den Konfigurationsrekorder zu stoppen

Der folgende Befehl stoppt den Standardkonfigurationsrekorder:

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

Wenn der Befehl erfolgreich ist, gibt AWS Config keine Ausgabe zurück. Führen Sie den Befehl `get-status` aus, um sicherzustellen, dass AWS Config Ihre Ressourcen nicht aufzeichnet.

- API-Einheiten finden Sie [StopConfigurationRecorder](#) in der AWS CLI Befehlsreferenz.

subscribe

Das folgende Codebeispiel zeigt die Verwendung `subscribe`.

AWS CLI

Um AWS Config zu abonnieren

Mit dem folgenden Befehl werden der Standardlieferkanal und der Konfigurationsrekorder erstellt. Der Befehl gibt auch den Amazon S3 S3-Bucket und SNS das Amazon-Thema an, an das AWS Config Konfigurationsinformationen liefert:

```
aws configservice subscribe --s3-bucket config-bucket-123456789012
--sns-topic arn:aws:sns:us-east-1:123456789012:config-topic --iam-
role arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6
```

Ausgabe:

```
Using existing S3 bucket: config-bucket-123456789012
Using existing SNS topic: arn:aws:sns:us-east-1:123456789012:config-topic
Subscribe succeeded:

Configuration Recorders: [
  {
    "recordingGroup": {
      "allSupported": true,
      "resourceTypes": [],
      "includeGlobalResourceTypes": false
    },
    "roleARN": "arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6",
    "name": "default"
  }
]

Delivery Channels: [
  {
    "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
    "name": "default",
    "s3BucketName": "config-bucket-123456789012"
  }
]
```

- API-Einheiten finden Sie unter [Abonnieren](#) in der AWS CLI Befehlsreferenz.

Amazon Connect Connect-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Connect Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-user

Das folgende Codebeispiel zeigt die Verwendung `create-user`.

AWS CLI

Um einen Benutzer zu erstellen

Das folgende `create-user` Beispiel fügt der angegebenen Amazon Connect Connect-Instance einen Benutzer mit den angegebenen Attributen hinzu.

```
aws connect create-user \  
  --username Mary \  
  --password Pass@Word1 \  
  --identity-info FirstName=Mary,LastName=Major \  
  --phone-  
config PhoneType=DESK_PHONE,AutoAccept=true,AfterContactWorkTimeLimit=60,DeskPhoneNumber=  
+15555551212 \  
  --security-profile-id 12345678-1111-2222-aaaa-a1b2c3d4f5g7 \  
  --routing-profile-id 87654321-9999-3434-abcd-x1y2z3a1b2c3 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "UserId": "87654321-2222-1234-1234-111234567891",  
  "UserArn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111/agent/87654321-2222-1234-1234-111234567891"  
}
```

Weitere Informationen finden [Sie unter Benutzer hinzufügen](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [CreateUser](#) in der AWS CLI Befehlsreferenz.

delete-user

Das folgende Codebeispiel zeigt die Verwendung `delete-user`.

AWS CLI

Benutzer löschen

Das folgende `delete-user` Beispiel löscht den angegebenen Benutzer aus der angegebenen Amazon Connect Connect-Instance.

```
aws connect delete-user \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --user-id 87654321-2222-1234-1234-111234567891
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzer verwalten](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteUser](#) in der AWS CLI Befehlsreferenz.

describe-user-hierarchy-group

Das folgende Codebeispiel zeigt die Verwendung `describe-user-hierarchy-group`.

AWS CLI

Um die Details für eine Hierarchiegruppe anzuzeigen

Im folgenden `describe-user-hierarchy-group` Beispiel werden die Details für die angegebene Amazon Connect Connect-Hierarchiegruppe angezeigt.

```
aws connect describe-user-hierarchy-group \  
  --hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "HierarchyGroup": {
```

```

    "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111/agent-group/12345678-1111-2222-800e-a2b3c4d5f6g7",
    "Name": "Example Corporation",
    "LevelId": "1",
    "HierarchyPath": {
      "LevelOne": {
        "Id": "abcdefgh-3333-4444-8af3-201123456789",
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group/
abcdefgh-3333-4444-8af3-201123456789",
        "Name": "Example Corporation"
      }
    }
  }
}

```

Weitere Informationen finden Sie unter [Agentenhierarchien einrichten](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie unter [DescribeUserHierarchyGroup AWS CLI Befehlsreferenz](#).

describe-user-hierarchy-structure

Das folgende Codebeispiel zeigt die Verwendung `describe-user-hierarchy-structure`.

AWS CLI

Um die Details für eine Hierarchiestruktur anzuzeigen

Im folgenden `describe-user-hierarchy-structure` Beispiel werden die Details für die Hierarchiestruktur für die angegebene Amazon Connect Connect-Instance angezeigt.

```

aws connect describe-user-hierarchy-group \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "HierarchyStructure": {
    "LevelOne": {
      "Id": "12345678-1111-2222-800e-aaabbb555gg",

```

```

    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/1",
    "Name": "Corporation"
  },
  "LevelTwo": {
    "Id": "87654321-2222-3333-ac99-123456789102",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/2",
    "Name": "Services Division"
  },
  "LevelThree": {
    "Id": "abcdefgh-3333-4444-8af3-201123456789",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/3",
    "Name": "EU Site"
  }
}
}
}

```

Weitere Informationen finden Sie unter [Agentenhierarchien einrichten](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie unter [DescribeUserHierarchyStructure AWS CLI Befehlsreferenz](#).

describe-user

Das folgende Codebeispiel zeigt die Verwendung `describe-user`.

AWS CLI

Um die Details für einen Benutzer anzuzeigen

Im folgenden `describe-user` Beispiel werden die Details für den angegebenen Amazon Connect Connect-Benutzer angezeigt.

```

aws connect describe-user \
  --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Ausgabe:

```
{
```

```
"User": {
  "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
  "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
  "Username": "Jane",
  "IdentityInfo": {
    "FirstName": "Jane",
    "LastName": "Doe",
    "Email": "example.com"
  },
  "PhoneConfig": {
    "PhoneType": "SOFT_PHONE",
    "AutoAccept": false,
    "AfterContactWorkTimeLimit": 0,
    "DeskPhoneNumber": ""
  },
  "DirectoryUserId": "8b444cf6-b368-4f29-ba18-07af27405658",
  "SecurityProfileIds": [
    "b6f85a42-1dc5-443b-b621-de0abf70c9cf"
  ],
  "RoutingProfileId": "0be36ee9-2b5f-4ef4-bcf7-87738e5be0e5",
  "Tags": {}
}
```

Weitere Informationen finden Sie unter [Benutzer verwalten](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeUser](#) in der AWS CLI Befehlsreferenz.

get-contact-attributes

Das folgende Codebeispiel zeigt die Verwendung von `get-contact-attributes`.

AWS CLI

Um die Attribute für einen Kontakt abzurufen

Im folgenden `get-contact-attributes` Beispiel werden die Attribute abgerufen, die für den angegebenen Amazon Connect Connect-Kontakt festgelegt wurden.

```
aws connect get-contact-attributes \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
```

```
--initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7
```

Ausgabe:

```
{
  "Attributes": {
    "greetingPlayed": "true"
  }
}
```

Weitere Informationen finden Sie unter [Verwenden von Amazon Connect Connect-Kontaktattributen](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [GetContactAttributes](#) in der AWS CLI Befehlsreferenz.

list-contact-flows

Das folgende Codebeispiel zeigt die Verwendung `list-contact-flows`.

AWS CLI

Um die Kontaktflüsse in einer Instanz aufzulisten

Das folgende `list-contact-flows` Beispiel listet die Kontaktabläufe in der angegebenen Amazon Connect Connect-Instanz auf.

```
aws connect list-contact-flows \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "ContactFlowSummaryList": [
    {
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/12345678-1111-2222-800e-a2b3c4d5f6g7",
      "Name": "Default queue transfer",
      "ContactFlowType": "QUEUE_TRANSFER"
    },
    {
```



```

        "Id": "87654321-2222-3333-ac99-123456789102",
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/87654321-2222-3333-
ac99-123456789102",
        "Name": "Default agent hold",
        "ContactFlowType": "AGENT_HOLD"
    },
    {
        "Id": "abcdefgh-3333-4444-8af3-201123456789",
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/
abcdefgh-3333-4444-8af3-201123456789",
        "Name": "Default customer hold",
        "ContactFlowType": "CUSTOMER_HOLD"
    },
]
}

```

Weitere Informationen finden Sie unter [Amazon Connect Connect-Kontaktbläufe erstellen](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListContactFlows](#) in der AWS CLI Befehlsreferenz.

list-hours-of-operations

Das folgende Codebeispiel zeigt die Verwendung `list-hours-of-operations`.

AWS CLI

Um die Öffnungszeiten einer Instanz aufzulisten

Das folgende `list-hours-of-operations` Beispiel listet die Betriebszeiten für die angegebene Amazon Connect Connect-Instanz auf.

```

aws connect list-hours-of-operations \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Ausgabe:

```

{
  "HoursOfOperationSummaryList": [
    {

```

```

        "Id": "d69f1f84-7457-4924-8fbe-e64875546259",
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/operating-hours/d69f1f84-7457-4924-8fbe-e64875546259",
        "Name": "Basic Hours"
    }
]
}

```

Weitere Informationen finden Sie unter [Festlegen der Öffnungszeiten für eine Warteschlange](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListHoursOfOperations](#) unter AWS CLI Befehlsreferenz.

list-phone-numbers

Das folgende Codebeispiel zeigt die Verwendung `list-phone-numbers`.

AWS CLI

Um die Telefonnummern in einer Instanz aufzulisten

Das folgende `list-phone-numbers` Beispiel listet die Telefonnummern in der angegebenen Amazon Connect Connect-Instanz auf.

```

aws connect list-phone-numbers \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "PhoneNumberSummaryList": [
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/xyz80zxy-xyz1-80zx-
zx80-11111EXAMPLE",
      "PhoneNumber": "+17065551212",
      "PhoneNumberType": "DID",
      "PhoneNumberCountryCode": "US"
    },
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",

```

```

        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/ccc0ccc-xyz1-80zx-
zx80-22222EXAMPLE",
        "PhoneNumber": "+18555551212",
        "PhoneNumberType": "TOLL_FREE",
        "PhoneNumberCountryCode": "US"
    }
]
}

```

Weitere Informationen finden Sie unter [Telefonnummern für Ihr Contact Center einrichten](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListPhoneNumbers](#) unter AWS CLI Befehlsreferenz.

list-queues

Das folgende Codebeispiel zeigt die Verwendung `list-queues`.

AWS CLI

Um die Warteschlangen in einer Instanz aufzulisten

Das folgende `list-queues` Beispiel listet die Warteschlangen in der angegebenen Amazon Connect Connect-Instance auf.

```

aws connect list-queues \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "QueueSummaryList": [
    {
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/12345678-1111-2222-800e-
a2b3c4d5f6g7",
      "QueueType": "AGENT"
    },
    {
      "Id": "87654321-2222-3333-ac99-123456789102",

```

```

    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/87654321-2222-3333-
ac99-123456789102",
    "QueueType": "AGENT"
  },
  {
    "Id": "abcdefgh-3333-4444-8af3-201123456789",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/
abcdefgh-3333-4444-8af3-201123456789",
    "QueueType": "AGENT"
  },
  {
    "Id": "hgfedcba-4444-5555-a31f-123456789102",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/hgfedcba-4444-5555-a31f-123456789102",
    "Name": "BasicQueue",
    "QueueType": "STANDARD"
  },
]
}

```

Weitere Informationen finden Sie unter [Eine Warteschlange erstellen](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListQueues](#) in der AWS CLI Befehlsreferenz.

list-routing-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-routing-profiles`.

AWS CLI

Um die Routing-Profile in einer Instanz aufzulisten

Das folgende `list-routing-profiles` Beispiel listet die Routing-Profile in der angegebenen Amazon Connect Connect-Instanz auf.

```

aws connect list-routing-profiles \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```
{
  "RoutingProfileSummaryList": [
    {
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/routing-profile/12345678-1111-2222-800e-
a2b3c4d5f6g7",
      "Name": "Basic Routing Profile"
    },
  ]
}
```

Weitere Informationen finden Sie unter [Erstellen eines Routing-Profils](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListRoutingProfiles](#) unter AWS CLI Befehlsreferenz.

list-security-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-security-profiles`.

AWS CLI

Um die Sicherheitsprofile in einer Instanz aufzulisten

Das folgende `list-security-profiles` Beispiel listet die Sicherheitsprofile in der angegebenen Amazon Connect Connect-Instanz auf.

```
aws connect list-security-profiles \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "SecurityProfileSummaryList": [
    {
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
a2b3c4d5f6g7",
      "Name": "CallCenterManager"
    },
  ],
}
```

```

    {
      "Id": "87654321-2222-3333-ac99-123456789102",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/87654321-2222-3333-
ac99-123456789102",
      "Name": "QualityAnalyst"
    },
    {
      "Id": "abcdefgh-3333-4444-8af3-201123456789",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/
abcdefgh-3333-4444-8af3-201123456789",
      "Name": "Agent"
    },
    {
      "Id": "12345678-1111-2222-800e-x2y3c4d5fzzzz",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
x2y3c4d5fzzzz",
      "Name": "Admin"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Berechtigungen zuweisen: Sicherheitsprofile](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListSecurityProfiles](#) in der AWS CLI Befehlsreferenz.

list-user-hierarchy-groups

Das folgende Codebeispiel zeigt die Verwendung `list-user-hierarchy-groups`.

AWS CLI

Um die Benutzerhierarchiegruppen in einer Instanz aufzulisten

Das folgende `list-user-hierarchy-groups` Beispiel listet die Benutzerhierarchiegruppen in der angegebenen Amazon Connect Connect-Instanz auf.

```

aws connect list-user-hierarchy-groups \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Ausgabe:

```
{
  "UserHierarchyGroupSummaryList": [
    {
      "Id": "0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent-group/0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
      "Name": "Example Corporation"
    },
  ]
}
```

Weitere Informationen finden Sie unter [Agentenhierarchien einrichten](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie unter [ListUserHierarchyGroups AWS CLI](#) Befehlsreferenz.

list-users

Das folgende Codebeispiel zeigt die Verwendung `list-users`.

AWS CLI

Um die Benutzerhierarchiegruppen in einer Instanz aufzulisten

Das folgende `list-users` Beispiel listet die Benutzer in der angegebenen Amazon Connect Connect-Instanz auf.

```
aws connect list-users \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Ausgabe:

```
{
  "UserSummaryList": [
    {
      "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
      "Username": "Jane"
    },
  ],
}
```

```

    {
      "Id": "46f0c67c-3fc7-4806-ac99-403798788c14",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/46f0c67c-3fc7-4806-ac99-403798788c14",
      "Username": "Paulo"
    },
    {
      "Id": "55a83578-95e1-4710-8af3-2b7afe310e48",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/55a83578-95e1-4710-8af3-2b7afe310e48",
      "Username": "JohnD"
    },
    {
      "Id": "703e27b5-c9f0-4f1f-a239-64ccbb160125",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/703e27b5-c9f0-4f1f-a239-64ccbb160125",
      "Username": "JohnS"
    }
  ]
}

```

Weitere Informationen finden [Sie unter Benutzer hinzufügen](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [ListUsers](#) in der AWS CLI Befehlsreferenz.

update-contact-attributes

Das folgende Codebeispiel zeigt die Verwendung `update-contact-attributes`.

AWS CLI

Um das Attribut eines Kontakts zu aktualisieren

Das folgende `update-contact-attributes` Beispiel aktualisiert das `greetingPlayed` Attribut für den angegebenen Amazon Connect Connect-Benutzer.

```

aws connect update-contact-attributes \
  --initial-contact-id 11111111-2222-3333-4444-12345678910 \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --attributes greetingPlayed=false

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von Amazon Connect Connect-Kontaktattributen](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateContactAttributes](#) in der AWS CLI Befehlsreferenz.

update-user-hierarchy

Das folgende Codebeispiel zeigt die Verwendung `update-user-hierarchy`.

AWS CLI

Um die Hierarchie eines Benutzers zu aktualisieren

Das folgende `update-user-hierarchy` Beispiel aktualisiert die Agentenhierarchie für den angegebenen Amazon Connect Connect-Benutzer.

```
aws connect update-user-hierarchy \  
  --hierarchy-group-id 12345678-a1b2-c3d4-e5f6-123456789abc \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Agenteneinstellungen konfigurieren](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateUserHierarchy](#) in der AWS CLI Befehlsreferenz.

update-user-identity-info

Das folgende Codebeispiel zeigt die Verwendung `update-user-identity-info`.

AWS CLI

Um die Identitätsinformationen eines Benutzers zu aktualisieren

Im folgenden `update-user-identity-info` Beispiel werden die Identitätsinformationen für den angegebenen Amazon Connect Connect-Benutzer aktualisiert.

```
aws connect update-user-identity-info \  
  --identity-info FirstName=Mary,LastName=Major,Email=marym@example.com \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

```
--user-id 87654321-2222-1234-1234-111234567891 \  
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Agenteneinstellungen konfigurieren](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateUserIdentityInfo](#) in der AWS CLI Befehlsreferenz.

update-user-phone-config

Das folgende Codebeispiel zeigt die Verwendung `update-user-phone-config`.

AWS CLI

Um die Telefonkonfiguration eines Benutzers zu aktualisieren

Im folgenden `update-user-phone-config` Beispiel wird die Telefonkonfiguration für den angegebenen Benutzer aktualisiert.

```
aws connect update-user-phone-config \  
  --phone-  
config PhoneType=SOFT_PHONE,AutoAccept=false,AfterContactWorkTimeLimit=60,DeskPhoneNumber=  
+18005551212 \  
  --user-id 12345678-4444-3333-2222-111122223333 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Agenteneinstellungen konfigurieren](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateUserPhoneConfig](#) in der AWS CLI Befehlsreferenz.

update-user-routing-profile

Das folgende Codebeispiel zeigt die Verwendung `update-user-routing-profile`.

AWS CLI

Um das Routing-Profil eines Benutzers zu aktualisieren

Das folgende `update-user-routing-profile` Beispiel aktualisiert das Routing-Profil für den angegebenen Amazon Connect Connect-Benutzer.

```
aws connect update-user-routing-profile \  
  --routing-profile-id 12345678-1111-3333-2222-4444EXAMPLE \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Agenteneinstellungen konfigurieren](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateUserRoutingProfile](#) in der AWS CLI Befehlsreferenz.

update-user-security-profiles

Das folgende Codebeispiel zeigt die Verwendung `update-user-security-profiles`.

AWS CLI

Um die Sicherheitsprofile eines Benutzers zu aktualisieren

Das folgende `update-user-security-profiles` Beispiel aktualisiert das Sicherheitsprofil für den angegebenen Amazon Connect Connect-Benutzer.

```
aws connect update-user-security-profiles \  
  --security-profile-ids 12345678-1234-1234-1234-1234567892111 \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Berechtigungen zuweisen: Sicherheitsprofile](#) im Amazon Connect Connect-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateUserSecurityProfiles](#) in der AWS CLI Befehlsreferenz.

AWS Cost and Usage Report Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Cost and Usage Report.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-report-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-report-definition`.

AWS CLI

Um einen AWS Kosten- und Nutzungsbericht zu löschen

In diesem Beispiel wird ein AWS Kosten- und Nutzungsbericht gelöscht.

Befehl:

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- API-Einheiten finden Sie [DeleteReportDefinition](#) in der AWS CLI Befehlsreferenz.

describe-report-definitions

Das folgende Codebeispiel zeigt die Verwendung `describe-report-definitions`.

AWS CLI

Um eine Liste mit AWS Kosten- und Nutzungsberichten abzurufen

In diesem Beispiel wird eine Liste von AWS Kosten- und Nutzungsberichten beschrieben, die einem Konto gehören.

Befehl:

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

Ausgabe:

```
{
  "ReportDefinitions": [
    {
      "ReportName": "ExampleReport",
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
        "REDSHIFT",
        "QUICKSIGHT"
      ],
      "AdditionalSchemaElements": [
        "RESOURCES"
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeReportDefinitions](#) in der AWS CLI Befehlsreferenz.

put-report-definition

Das folgende Codebeispiel zeigt die Verwendung `put-report-definition`.

AWS CLI

Um AWS Kosten- und Nutzungsberichte zu erstellen

Im folgenden `put-report-definition` Beispiel wird ein täglicher AWS Kosten- und Nutzungsbericht erstellt, den Sie in Amazon Redshift oder Amazon QuickSight hochladen können.

```
aws cur put-report-definition --report-definition file://report-definition.json
```

Inhalt von `report-definition.json`:

```
{
  "ReportName": "ExampleReport",
  "TimeUnit": "DAILY",
  "Format": "textORcsv",
  "Compression": "ZIP",
  "AdditionalSchemaElements": [
    "RESOURCES"
  ],
  "S3Bucket": "example-s3-bucket",
  "S3Prefix": "exampleprefix",
  "S3Region": "us-east-1",
  "AdditionalArtifacts": [
    "REDSHIFT",
    "QUICKSIGHT"
  ]
}
```

- API-Einheiten finden Sie [PutReportDefinition](#) in der AWS CLI Befehlsreferenz.

Beispiele für den Cost Explorer Explorer-Service mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe des AWS Command Line Interface with Cost Explorer Service Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-cost-and-usage

Das folgende Codebeispiel zeigt die Verwendung `get-cost-and-usage`.

AWS CLI

Um die S3-Nutzung eines Kontos für den Monat September 2017 abzurufen

Im folgenden `get-cost-and-usage` Beispiel wird die S3-Nutzung eines Kontos für den Monat September 2017 abgerufen.

```
aws ce get-cost-and-usage \  
  --time-period Start=2017-09-01,End=2017-10-01 \  
  --granularity MONTHLY \  
  --metrics "BlendedCost" "UnblendedCost" "UsageQuantity" \  
  --group-by Type=DIMENSION,Key=SERVICE Type=TAG,Key=Environment \  
  --filter file://filters.json
```

Inhalt von `filters.json`:

```
{  
  "Dimensions": {  
    "Key": "SERVICE",  
    "Values": [  
      "Amazon Simple Storage Service"  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "GroupDefinitions": [  
    {  
      "Type": "DIMENSION",  
      "Key": "SERVICE"  
    },  
    {  
      "Type": "TAG",  
      "Key": "Environment"  
    }  
  ],  
  "ResultsByTime": [  
    {  
      "Estimated": false,  
      "TimePeriod": {  
        "Start": "2017-09-01",
```

```
    "End": "2017-10-01"
  },
  "Total": {},
  "Groups": [
    {
      "Keys": [
        "Amazon Simple Storage Service",
        "Environment$"
      ],
      "Metrics": {
        "BlendedCost": {
          "Amount": "40.3527508453",
          "Unit": "USD"
        },
        "UnblendedCost": {
          "Amount": "40.3543773134",
          "Unit": "USD"
        },
        "UsageQuantity": {
          "Amount": "9312771.098461578",
          "Unit": "N/A"
        }
      }
    }
  ],
  {
    "Keys": [
      "Amazon Simple Storage Service",
      "Environment$Dev"
    ],
    "Metrics": {
      "BlendedCost": {
        "Amount": "0.2682364644",
        "Unit": "USD"
      },
      "UnblendedCost": {
        "Amount": "0.2682364644",
        "Unit": "USD"
      },
      "UsageQuantity": {
        "Amount": "22403.4395271182",
        "Unit": "N/A"
      }
    }
  }
}
```



```

    ]
  }
]
}

```

- API-Einheiten finden Sie [GetCostAndUsage](#) in der AWS CLI Befehlsreferenz.

get-dimension-values

Das folgende Codebeispiel zeigt die Verwendung `get-dimension-values`.

AWS CLI

Um die Tags für die Dimension `SERVICE` mit dem Wert „Elastic“ abzurufen

In diesem Beispiel werden die Tags für die Dimension `SERVICE` mit dem Wert „Elastic“ für den Zeitraum 1. Januar 2017 bis 18. Mai 2017 abgerufen.

Befehl:

```
aws ce get-dimension-values --search-string Elastic --time-period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

Ausgabe:

```
{
  "TotalSize": 6,
  "DimensionValues": [
    {
      "Attributes": {},
      "Value": "Amazon ElastiCache"
    },
    {
      "Attributes": {},
      "Value": "EC2 - Other"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Compute Cloud - Compute"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Load Balancing"
    }
  ]
}
```

```

    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic MapReduce"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elasticsearch Service"
    }
  ],
  "ReturnSize": 6
}

```

- API-Einheiten finden Sie [GetDimensionValues](#) in der AWS CLI Befehlsreferenz.

get-reservation-coverage

Das folgende Codebeispiel zeigt die Verwendung `get-reservation-coverage`.

AWS CLI

Um die Reservierungsabdeckung für EC2 t2.nano-Instances in der Region us-east-1 abzurufen

In diesem Beispiel wird die Reservierungsabdeckung für EC2 t2.nano-Instances in der Region us-east-1 für Juli-September 2017 abgerufen.

Befehl:

```
aws ce get-reservation-coverage --time-period Start=2017-07-01,End=2017-10-01 --
group-by Type=Dimension,Key=REGION --filter file://filters.json
```

filters.json:

```

{
  "And": [
    {
      "Dimensions": {
        "Key": "INSTANCE_TYPE",
        "Values": [
          "t2.nano"
        ]
      },
    },
    "Dimensions": {

```

```
    "Key": "REGION",
    "Values": [
      "us-east-1"
    ]
  }
]
}
```

Ausgabe:

```
{
  "TotalSize": 6,
  "DimensionValues": [
    {
      "Attributes": {},
      "Value": "Amazon ElastiCache"
    },
    {
      "Attributes": {},
      "Value": "EC2 - Other"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Compute Cloud - Compute"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Load Balancing"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic MapReduce"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elasticsearch Service"
    }
  ],
  "ReturnSize": 6
}
```

- API-Einheiten finden Sie [GetReservationCoverage](#) in der AWS CLI Befehlsreferenz.

get-reservation-purchase-recommendation

Das folgende Codebeispiel zeigt die Verwendung `get-reservation-purchase-recommendation`.

AWS CLI

Um die Reservierungsempfehlungen für Partial Upfront EC2 RIs mit einer Laufzeit von drei Jahren abzurufen

Im folgenden `get-reservation-purchase-recommendation` Beispiel werden Empfehlungen für Partial EC2 Upfront-Instances mit einer Laufzeit von drei Jahren abgerufen, basierend auf den letzten 60 Nutzungstagen. EC2

```
aws ce get-reservation-purchase-recommendation \
  --service "Amazon Redshift" \
  --lookback-period-in-days SIXTY_DAYS \
  --term-in-years THREE_YEARS \
  --payment-option PARTIAL_UPFRONT
```

Ausgabe:

```
{
  "Recommendations": [],
  "Metadata": {
    "GenerationTimestamp": "2018-08-08T15:20:57Z",
    "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"
  }
}
```

- API-Einheiten finden Sie [GetReservationPurchaseRecommendation](#) in AWS CLI der Befehlsreferenz.

get-reservation-utilization

Das folgende Codebeispiel zeigt die Verwendung `get-reservation-utilization`.

AWS CLI

Um die Reservierungsnutzung für Ihr Konto abzurufen

Im folgenden `get-reservation-utilization` Beispiel wird die RI-Auslastung für alle `t2.nano`-Instance-Typen vom 01.03.2018 bis 01.08.2018 für das Konto abgerufen.

```
aws ce get-reservation-utilization \  
  --time-period Start=2018-03-01,End=2018-08-01 \  
  --filter file://filters.json
```

Inhalt von `filters.json`:

```
{  
  "Dimensions": {  
    "Key": "INSTANCE_TYPE",  
    "Values": [  
      "t2.nano"  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "Total": {  
    "TotalAmortizedFee": "0",  
    "UtilizationPercentage": "0",  
    "PurchasedHours": "0",  
    "NetRISavings": "0",  
    "TotalActualHours": "0",  
    "AmortizedRecurringFee": "0",  
    "UnusedHours": "0",  
    "TotalPotentialRISavings": "0",  
    "OnDemandCostOfRIHoursUsed": "0",  
    "AmortizedUpfrontFee": "0"  
  },  
  "UtilizationsByTime": []  
}
```

- Einzelheiten API finden Sie unter Befehlsreferenz. [GetReservationUtilization](#) AWS CLI

get-tags

Das folgende Codebeispiel zeigt die Verwendung `get-tags`.

AWS CLI

Um Schlüssel und Werte für ein Kostenzuweisungs-Tag abzurufen

In diesem Beispiel werden alle Kostenzuordnungs-Tags mit dem Schlüssel „Project“ und einem Wert abgerufen, der "secretProject" enthält.

Befehl:

```
aws ce get-tags --search-string secretProject --time-  
period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

Ausgabe:

```
{  
  "ReturnSize": 2,  
  "Tags": [  
    "secretProject1",  
    "secretProject2"  
  ],  
  "TotalSize": 2  
}
```

- API-Einheiten finden Sie unter [GetTags AWS CLI](#) Befehlsreferenz.

Firehose-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Firehose Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

list-delivery-streams

Das folgende Codebeispiel zeigt die Verwendung `list-delivery-streams`.

AWS CLI

Um die verfügbaren Lieferdatenströme aufzulisten

Das folgende `list-delivery-streams` Beispiel listet die verfügbaren Lieferstreams in Ihrem AWS Konto auf.

```
aws firehose list-delivery-streams
```

Ausgabe:

```
{
  "DeliveryStreamNames": [
    "my-stream"
  ],
  "HasMoreDeliveryStreams": false
}
```

Weitere Informationen finden Sie unter [Erstellen eines Amazon Kinesis Data Firehose-Bereitstellungs-Streams](#) im Amazon Kinesis Data Firehose-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListDeliveryStreams](#) in der AWS CLI Befehlsreferenz.

put-record-batch

Das folgende Codebeispiel zeigt die Verwendung `put-record-batch`.

AWS CLI

Um mehrere Datensätze in einen Stream zu schreiben

Das folgende `put-record-batch` Beispiel schreibt drei Datensätze in einen Stream. Die Daten sind im Base64-Format codiert.

```
aws firehose put-record-batch \
```

```
--delivery-stream-name my-stream \  
--records file://records.json
```

Inhalt von `myfile.json`:

```
[  
  {"Data": "Rmlyc3QgdGhpbmc="},  
  {"Data": "U2Vjb25kIHRoaW5n"},  
  {"Data": "VGhpcmQgdGhpbmc="}  
]
```

Ausgabe:

```
{  
  "FailedPutCount": 0,  
  "Encrypted": false,  
  "RequestResponses": [  
    {  
      "RecordId": "9D20J6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhY0aWKPSmZy/  
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn  
+3boDnURDEaD42V7Giixp0yLJkYZcae1i7HzlCEoy9LJhMr8EjDSi40m/9Vc2uhwwuAtGE0XKpxJ2WD7ZRwtAnY1KAnv  
    },  
    {  
      "RecordId": "jFirejqxCLlK5xjH/UNm1MVcjkTEn76I7916X9PaZ  
+PVa0SXDFu1WG0qEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GkmR5z9016XKJ  
+/+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP90iHLbzGwy8/  
DfH8sqWEUYASNJKS8GXP5s"  
    },  
    {  
      "RecordId":  
      "oy0amQ40o5Y2YV4vxzufdcM00w6n3EPr3tpPJGoYVnKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3iie  
DTBt3qBlmTj7Xq8SKVb01S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/s0kQLe+qh0sYKXWl"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Senden von Daten an einen Amazon Kinesis Data Firehose Delivery Stream](#) im Amazon Kinesis Data Firehose Developer Guide.

- API-Einheiten finden Sie unter [PutRecordBatch](#) Befehlsreferenz. AWS CLI

put-record

Das folgende Codebeispiel zeigt die Verwendung `put-record`.

AWS CLI

Um einen Datensatz in einen Stream zu schreiben

Das folgende `put-record` Beispiel schreibt Daten in einen Stream. Die Daten sind im Base64-Format codiert.

```
aws firehose put-record \  
  --delivery-stream-name my-stream \  
  --record '{"Data": "SGVsbG8gd29ybGQ="}'
```

Ausgabe:

```
{  
  "RecordId": "RjB5K/nnoGFHqwTsZ1Nd/  
TTqvjE8V5dsyXZTQn2JXrdpMT0wssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/  
j1Xujv31iDhnPdaLw4B0yM9Amv7PcCuB2079RuM0NhoakbyUymLwY8yt20G8X2420wu1j1Fafhci4erAt7QhDEvpwuK8  
  "Encrypted": false  
}
```

Weitere Informationen finden Sie unter [Senden von Daten an einen Amazon Kinesis Data Firehose Delivery Stream](#) im Amazon Kinesis Data Firehose Developer Guide.

- API-Einheiten finden Sie unter [PutRecord](#) Befehlsreferenz.AWS CLI

Amazon Data Lifecycle Manager Manager-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Data Lifecycle Manager Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-default-role

Das folgende Codebeispiel zeigt die Verwendung `create-default-role`.

AWS CLI

Um die erforderliche IAM Rolle für Amazon zu erstellen DLM

Im folgenden `d1m create-default-role` Beispiel wird die AWS DataLifecycleManagerDefaultRole Standardrolle für die Verwaltung von Snapshots erstellt.

```
aws d1m create-default-role \  
  --resource-type snapshot
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Standard-Service rollen für Amazon Data Lifecycle Manager](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDefaultRole](#) in der AWS CLI Befehlsreferenz.

create-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `create-lifecycle-policy`.

AWS CLI

Um eine Lebenszyklusrichtlinie zu erstellen

Im folgenden `create-lifecycle-policy` Beispiel wird eine Lebenszyklusrichtlinie erstellt, die einen täglichen Snapshot der Volumes zum angegebenen Zeitpunkt erstellt. Die angegebenen Tags werden den Snapshots hinzugefügt, und Tags werden ebenfalls aus dem Volume kopiert und den Snapshots hinzugefügt. Wenn die Erstellung eines neuen Snapshots die angegebene maximale Anzahl überschreitet, wird der älteste Snapshot gelöscht.

```
aws d1m create-lifecycle-policy \  
  --resource-type snapshot
```

```
--description "My first policy" \  
--state ENABLED \  
--execution-role-arn arn:aws:iam::12345678910:role/  
AWSDataLifecycleManagerDefaultRole \  
--policy-details file://policyDetails.json
```

Inhalt von policyDetails.json:

```
{  
  "ResourceTypes": [  
    "VOLUME"  
  ],  
  "TargetTags": [  
    {  
      "Key": "costCenter",  
      "Value": "115"  
    }  
  ],  
  "Schedules": [  
    {  
      "Name": "DailySnapshots",  
      "CopyTags": true,  
      "TagsToAdd": [  
        {  
          "Key": "type",  
          "Value": "myDailySnapshot"  
        }  
      ],  
      "CreateRule": {  
        "Interval": 24,  
        "IntervalUnit": "HOURS",  
        "Times": [  
          "03:00"  
        ]  
      },  
      "RetainRule": {  
        "Count": 5  
      }  
    }  
  ]  
}
```

Ausgabe:

```
{
  "PolicyId": "policy-0123456789abcdef0"
}
```

- API-Einheiten finden Sie [CreateLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

delete-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-lifecycle-policy`.

AWS CLI

Um eine Lebenszyklusrichtlinie zu löschen

Im folgenden Beispiel wird die angegebene Lebenszyklusrichtlinie gelöscht. :

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- API-Einheiten finden Sie [DeleteLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

get-lifecycle-policies

Das folgende Codebeispiel zeigt die Verwendung `get-lifecycle-policies`.

AWS CLI

Um eine Zusammenfassung Ihrer Lebenszyklusrichtlinien zu erhalten

Das folgende `get-lifecycle-policies` Beispiel listet alle Ihre Lebenszyklusrichtlinien auf.

```
aws dlm get-lifecycle-policies
```

Ausgabe:

```
{
  "Policies": [
    {
      "PolicyId": "policy-0123456789abcdef0",
      "Description": "My first policy",
      "State": "ENABLED"
    }
  ]
}
```

```
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetLifecyclePolicies](#) in der AWS CLI Befehlsreferenz.

get-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `get-lifecycle-policy`.

AWS CLI

Um eine Lebenszyklusrichtlinie zu beschreiben

Im folgenden `get-lifecycle-policy` Beispiel werden Details für die angegebene Lebenszyklusrichtlinie angezeigt.

```
aws dlm get-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0
```

Ausgabe:

```
{  
  "Policy": {  
    "PolicyId": "policy-0123456789abcdef0",  
    "Description": "My policy",  
    "State": "ENABLED",  
    "ExecutionRoleArn": "arn:aws:iam::123456789012:role/  
AWSDataLifecycleManagerDefaultRole",  
    "DateCreated": "2019-08-08T17:45:42Z",  
    "DateModified": "2019-08-08T17:45:42Z",  
    "PolicyDetails": {  
      "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",  
      "ResourceTypes": [  
        "VOLUME"  
      ],  
      "TargetTags": [  
        {  
          "Key": "costCenter",  
          "Value": "115"  
        }  
      ],  
    }  
  }  
}
```

```

    "Schedules": [
      {
        "Name": "DailySnapshots",
        "CopyTags": true,
        "TagsToAdd": [
          {
            "Key": "type",
            "Value": "myDailySnapshot"
          }
        ],
        "CreateRule": {
          "Interval": 24,
          "IntervalUnit": "HOURS",
          "Times": [
            "03:00"
          ]
        },
        "RetainRule": {
          "Count": 5
        }
      }
    ]
  }
}

```

- API-Einheiten finden Sie [GetLifecyclePolicy](#) unter AWS CLI Befehlsreferenz.

update-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `update-lifecycle-policy`.

AWS CLI

Beispiel 1: Um eine Lebenszyklusrichtlinie zu aktivieren

Das folgende `update-lifecycle-policy` Beispiel aktiviert die angegebene Lebenszyklusrichtlinie.

```

aws dlm update-lifecycle-policy \
  --policy-id policy-0123456789abcdef0 \
  --state ENABLED

```

Beispiel 2: Um eine Lebenszyklusrichtlinie zu deaktivieren

Im folgenden `update-lifecycle-policy` Beispiel wird die angegebene Lebenszyklusrichtlinie deaktiviert.

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --state DISABLED
```

Beispiel 3: Um die Details für die Lebenszyklusrichtlinie zu aktualisieren

Im folgenden `update-lifecycle-policy` Beispiel werden die Ziel-Tags für die angegebene Lebenszyklusrichtlinie aktualisiert.

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --policy-details file://policyDetails.json
```

Inhalt von `policyDetails.json`. Andere Details, auf die in dieser Datei nicht verwiesen wird, werden durch den Befehl nicht geändert.

```
{  
  "TargetTags": [  
    {  
      "Key": "costCenter",  
      "Value": "120"  
    },  
    {  
      "Key": "project",  
      "Value": "lima"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [UpdateLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

AWS Data Pipeline Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Data Pipeline.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

activate-pipeline

Das folgende Codebeispiel zeigt die Verwendung `activate-pipeline`.

AWS CLI

Um eine Pipeline zu aktivieren

In diesem Beispiel wird die angegebene Pipeline aktiviert:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Verwenden Sie den folgenden Befehl, um die Pipeline an einem bestimmten Datum und zu einer bestimmten Uhrzeit zu aktivieren:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --start-timestamp 2015-04-07T00:00:00Z
```

- API-Einheiten finden Sie [ActivatePipeline](#) in der AWS CLI Befehlsreferenz.

add-tags

Das folgende Codebeispiel zeigt die Verwendung `add-tags`.

AWS CLI

Um einer Pipeline ein Tag hinzuzufügen

In diesem Beispiel wird der angegebenen Pipeline das angegebene Tag hinzugefügt:


```
aws datapipeline add-tags --pipeline-id df-00627471S0VYZEXAMPLE --  
tags key=environment,value=production key=owner,value=sales
```

Verwenden Sie den Befehl `describe-pipelines`, um die Tags anzuzeigen. Die im Beispielbefehl hinzugefügten Tags werden beispielsweise in der Ausgabe für `describe-pipelines` wie folgt angezeigt:

```
{  
  ...  
  "tags": [  
    {  
      "value": "production",  
      "key": "environment"  
    },  
    {  
      "value": "sales",  
      "key": "owner"  
    }  
  ]  
  ...  
}
```

- API-Einheiten finden Sie [AddTags](#) in AWS CLI der Befehlsreferenz.

create-pipeline

Das folgende Codebeispiel zeigt die Verwendung `create-pipeline`.

AWS CLI

Um eine Pipeline zu erstellen

In diesem Beispiel wird eine Pipeline erstellt:

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

Das Folgende ist Ausgabebeispiel:

```
{  
  "pipelineId": "df-00627471S0VYZEXAMPLE"
```

```
}
```

- API-Einheiten finden Sie [CreatePipeline](#) in der AWS CLI Befehlsreferenz.

deactivate-pipeline

Das folgende Codebeispiel zeigt die Verwendung `deactivate-pipeline`.

AWS CLI

Um eine Pipeline zu deaktivieren

In diesem Beispiel wird die angegebene Pipeline deaktiviert:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Verwenden Sie den folgenden Befehl, um die Pipeline erst zu deaktivieren, wenn alle laufenden Aktivitäten abgeschlossen sind:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --no-cancel-active
```

- API-Einheiten finden Sie [DeactivatePipeline](#) in der AWS CLI Befehlsreferenz.

delete-pipeline

Das folgende Codebeispiel zeigt die Verwendung `delete-pipeline`.

AWS CLI

Um eine Pipeline zu löschen

In diesem Beispiel wird die angegebene Pipeline gelöscht:

```
aws datapipeline delete-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

- API-Einheiten finden Sie [DeletePipeline](#) in der AWS CLI Befehlsreferenz.

describe-pipelines

Das folgende Codebeispiel zeigt die Verwendung `describe-pipelines`.

AWS CLI

Um Ihre Pipelines zu beschreiben

Dieses Beispiel beschreibt die angegebene Pipeline:

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471S0VYZEXAMPLE
```

Das Folgende ist Ausgabebeispiel:

```
{
  "pipelineDescriptionList": [
    {
      "fields": [
        {
          "stringValue": "PENDING",
          "key": "@pipelineState"
        },
        {
          "stringValue": "my-pipeline",
          "key": "name"
        },
        {
          "stringValue": "2015-04-07T16:05:58",
          "key": "@creationTime"
        },
        {
          "stringValue": "df-00627471S0VYZEXAMPLE",
          "key": "@id"
        },
        {
          "stringValue": "123456789012",
          "key": "pipelineCreator"
        },
        {
          "stringValue": "PIPELINE",
          "key": "@sphere"
        },
        {
          "stringValue": "123456789012",
          "key": "@userId"
        },
        {
```

```

        "stringValue": "123456789012",
        "key": "@accountId"
    },
    {
        "stringValue": "my-pipeline-token",
        "key": "uniqueId"
    }
],
"pipelineId": "df-00627471S0VYZEXAMPLE",
"name": "my-pipeline",
"tags": []
}
]
}

```

- API-Einheiten finden Sie [DescribePipelines](#) in der AWS CLI Befehlsreferenz.

get-pipeline-definition

Das folgende Codebeispiel zeigt die Verwendung von `get-pipeline-definition`.

AWS CLI

Um eine Pipeline-Definition zu erhalten

In diesem Beispiel wird die Pipeline-Definition für die angegebene Pipeline abgerufen:

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE
```

Das Folgende ist ein Ausgabebeispiel:

```

{
  "parameters": [
    {
      "type": "AWS::S3::ObjectKey",
      "id": "myS3OutputLoc",
      "description": "S3 output folder"
    },
    {
      "default": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/data",
      "type": "AWS::S3::ObjectKey",
      "id": "myS3InputLoc",
      "description": "S3 input folder"
    }
  ]
}

```

```

    },
    {
      "default": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
${OUTPUT1_STAGING_DIR}/output.txt",
      "type": "String",
      "id": "myShellCmd",
      "description": "Shell command to run"
    }
  ],
  "objects": [
    {
      "type": "Ec2Resource",
      "terminateAfter": "20 Minutes",
      "instanceType": "t1.micro",
      "id": "EC2ResourceObj",
      "name": "EC2ResourceObj"
    },
    {
      "name": "Default",
      "failureAndRerunMode": "CASCADE",
      "resourceRole": "DataPipelineDefaultResourceRole",
      "schedule": {
        "ref": "DefaultSchedule"
      },
      "role": "DataPipelineDefaultRole",
      "scheduleType": "cron",
      "id": "Default"
    },
    {
      "directoryPath": "#{myS3OutputLoc}/#{format(@scheduledStartTime, 'YYYY-MM-
dd-HH-mm-ss')}}",
      "type": "S3DataNode",
      "id": "S3OutputLocation",
      "name": "S3OutputLocation"
    },
    {
      "directoryPath": "#{myS3InputLoc}",
      "type": "S3DataNode",
      "id": "S3InputLocation",
      "name": "S3InputLocation"
    },
    {
      "startAt": "FIRST_ACTIVATION_DATE_TIME",
      "name": "Every 15 minutes",

```

```

        "period": "15 minutes",
        "occurrences": "4",
        "type": "Schedule",
        "id": "DefaultSchedule"
    },
    {
        "name": "ShellCommandActivityObj",
        "command": "#{myShellCmd}",
        "output": {
            "ref": "S3OutputLocation"
        },
        "input": {
            "ref": "S3InputLocation"
        },
        "stage": "true",
        "type": "ShellCommandActivity",
        "id": "ShellCommandActivityObj",
        "runsOn": {
            "ref": "EC2ResourceObj"
        }
    }
],
"values": {
    "myS3OutputLoc": "s3://my-s3-bucket/",
    "myS3InputLoc": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/
data",
    "myShellCmd": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
${OUTPUT1_STAGING_DIR}/output.txt"
}
}

```

- API-Einheiten finden Sie [GetPipelineDefinition](#) unter AWS CLI Befehlsreferenz.

list-pipelines

Das folgende Codebeispiel zeigt die Verwendung `list-pipelines`.

AWS CLI

Um Ihre Pipelines aufzulisten

In diesem Beispiel werden Ihre Pipelines aufgeführt:

aws datapipeline list-pipelines

Das Folgende ist Ausgabebeispiel:

```
{
  "pipelineIdList": [
    {
      "id": "df-00627471S0VYZEXAMPLE",
      "name": "my-pipeline"
    },
    {
      "id": "df-09028963KNVMREXAMPLE",
      "name": "ImportDDB"
    },
    {
      "id": "df-0870198233ZYVEXAMPLE",
      "name": "CrossRegionDDB"
    },
    {
      "id": "df-00189603TB4MZEXAMPLE",
      "name": "CopyRedshift"
    }
  ]
}
```

- API-Einheiten finden Sie [ListPipelines](#) in der AWS CLI Befehlsreferenz.

list-runs

Das folgende Codebeispiel zeigt die Verwendung `list-runs`.

AWS CLI

Beispiel 1: Um Ihre Pipeline-Läufe aufzulisten

Das folgende `list-runs` Beispiel listet die Läufe für die angegebene Pipeline auf.

```
aws datapipeline list-runs --pipeline-id df-00627471S0VYZEXAMPLE
```

Ausgabe:

Name	Scheduled Start	Status	ID
	Started	Ended	
1. EC2ResourceObj	2015-04-12T17:33:02	CREATING	
@EC2ResourceObj_2015-04-12T17:33:02		2015-04-12T17:33:10	
2. S3InputLocation	2015-04-12T17:33:02	FINISHED	
@S3InputLocation_2015-04-12T17:33:02		2015-04-12T17:33:09	
3. S3OutputLocation	2015-04-12T17:33:02	WAITING_ON_DEPENDENCIES	
@S3OutputLocation_2015-04-12T17:33:02		2015-04-12T17:33:09	
4. ShellCommandActivityObj	2015-04-12T17:33:02	WAITING_FOR_RUNNER	
@ShellCommandActivityObj_2015-04-12T17:33:02		2015-04-12T17:33:09	

Beispiel 2: Um die Pipeline-Läufe zwischen den angegebenen Daten aufzulisten

Im folgenden `list-runs` Beispiel werden die Daten verwendet `--start-interval`, um die Daten anzugeben, die in die Ausgabe aufgenommen werden sollen.

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUK05 --start-interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- API-Einheiten finden Sie [ListRuns](#) unter AWS CLI Befehlsreferenz.

put-pipeline-definition

Das folgende Codebeispiel zeigt die Verwendung `put-pipeline-definition`.

AWS CLI

Um eine Pipeline-Definition hochzuladen

In diesem Beispiel wird die angegebene Pipeline-Definition in die angegebene Pipeline hochgeladen:

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --pipeline-definition file://my-pipeline-definition.json
```

Das Folgende ist Ausgabebeispiel:

```
{
```



```
"validationErrors": [],  
"errored": false,  
"validationWarnings": []  
}
```

- API-Einheiten finden Sie unter [PutPipelineDefinition AWS CLI](#) Befehlsreferenz.

remove-tags

Das folgende Codebeispiel zeigt die Verwendung `remove-tags`.

AWS CLI

Um ein Tag aus einer Pipeline zu entfernen

In diesem Beispiel wird das angegebene Tag aus der angegebenen Pipeline entfernt:

```
aws datapipeline remove-tags --pipeline-id df-00627471S0VYZEXAMPLE --tag-  
keys environment
```

- API-Einheiten finden Sie [RemoveTags](#) unter AWS CLI Befehlsreferenz.

DataSync Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren DataSync.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)


```
--location-arn arn:aws:datsync:us-west-2:123456789012:location/Loc-  
abcdef01234567890 \  
--agent-arns arn:aws:datsync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
--kerberos-keytab file://hdfs.keytab  
--kerberos-krb5-conf file://krb5.conf
```

Inhalt von `hdfs.keytab`:

N/A. The content of this file is encrypted and not human readable.

Inhalt von `krb5.conf`:

```
[libdefaults]  
    default_realm = EXAMPLE.COM  
    dns_lookup_realm = false  
    dns_lookup_kdc = false  
    rdns = true  
    ticket_lifetime = 24h  
    forwardable = true  
    udp_preference_limit = 1000000  
    default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-  
sha1  
    default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-  
sha1  
    permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-  
sha1  
  
[realms]  
    EXAMPLE.COM = {  
        kdc = kdc1.example.com  
        admin_server = krbadmin.example.com  
        default_domain = example.com  
    }  
  
[domain_realm]  
    .example.com = EXAMPLE.COM  
    example.com = EXAMPLE.COM  
  
[logging]  
    kdc = FILE:/var/log/krb5kdc.log  
    admin_server = FILE:/var/log/kerberos/kadmin.log
```

```
default = FILE:/var/log/krb5libs.log
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS DataSync Benutzerhandbuch unter [Austauschen Ihres Agenten](#).

- API-Einheiten finden Sie [UpdateLocationHdfs](#) in der AWS CLI Befehlsreferenz.

update-location-nfs

Das folgende Codebeispiel zeigt die Verwendung `update-location-nfs`.

AWS CLI

Um Ihren Transferstandort mit einem neuen Agenten zu aktualisieren

Im folgenden `update-location-nfs` Beispiel wird Ihr DataSync NFS Standort mit einem neuen Agenten aktualisiert.

```
aws datasync update-location-nfs \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
  abcdef01234567890 \  
  --on-prem-config AgentArns=arn:aws:datasync:us-west-2:123456789012:agent/  
  agent-1234567890abcdef0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS DataSync Benutzerhandbuch unter [Austauschen Ihres Agenten](#).

- API-Einheiten finden Sie [UpdateLocationNfs](#) in der AWS CLI Befehlsreferenz.

update-location-object-storage

Das folgende Codebeispiel zeigt die Verwendung `update-location-object-storage`.

AWS CLI

Um Ihren Transferstandort mit einem neuen Agenten zu aktualisieren

Im folgenden `update-location-object-storage` Beispiel wird Ihr DataSync Objektspeicherort mit einem neuen Agenten aktualisiert.

```
aws datasync update-location-object-storage \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
  --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
  --secret-key secret-key-for-object-storage
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS DataSync Benutzerhandbuch unter [Austauschen Ihres Agenten](#).

- API-Einheiten finden Sie [UpdateLocationObjectStorage](#) in der AWS CLI Befehlsreferenz.

update-location-smb

Das folgende Codebeispiel zeigt die Verwendung `update-location-smb`.

AWS CLI

Um Ihren Transferstandort mit einem neuen Agenten zu aktualisieren

Im folgenden `update-location-smb` Beispiel wird Ihr DataSync SMB Standort mit einem neuen Agenten aktualisiert.

```
aws datasync update-location-smb \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
  --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
  --password smb-file-server-password
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS DataSync Benutzerhandbuch unter [Austauschen Ihres Agenten](#).

- API-Einheiten finden Sie [UpdateLocationSmb](#) in der AWS CLI Befehlsreferenz.

DAXBeispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementierenDAX.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-cluster

Das folgende Codebeispiel zeigt die Verwendungcreate-cluster.

AWS CLI

Um einen DAX Cluster zu erstellen

Im folgenden create-cluster Beispiel wird ein DAX Cluster mit den angegebenen Einstellungen erstellt.

```
aws dax create-cluster \  
  --cluster-name daxcluster \  
  --node-type dax.r4.large \  
  --replication-factor 3 \  
  --iam-role-arn roleARN \  
  --sse-specification Enabled=true
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterName": "daxcluster",  
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
```

```

    "TotalNodes": 3,
    "ActiveNodes": 0,
    "NodeType": "dax.r4.large",
    "Status": "creating",
    "ClusterDiscoveryEndpoint": {
      "Port": 8111
    },
    "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
    "SubnetGroup": "default",
    "SecurityGroups": [
      {
        "SecurityGroupIdentifier": "sg-1af6e36e",
        "Status": "active"
      }
    ],
    "IamRoleArn": "arn:aws:iam::123456789012:role/DAXServiceRoleForDynamoDBAccess",
    "ParameterGroup": {
      "ParameterGroupName": "default.dax1.0",
      "ParameterApplyStatus": "in-sync",
      "NodeIdsToReboot": []
    },
    "SSEDescription": {
      "Status": "ENABLED"
    }
  }
}

```

Weitere Informationen finden Sie unter [Schritt 3: Einen DAX Cluster erstellen](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [CreateCluster AWS CLI Befehlsreferenz](#).

create-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `create-parameter-group`.

AWS CLI

Um eine Parametergruppe zu erstellen

Das folgende `create-parameter-group` -Beispiel erstellt eine Parametergruppe mit den angegebenen Einstellungen.

```
aws dax create-parameter-group \  
  --parameter-group-name daxparametergroup \  
  --description "A new parameter group"
```

Ausgabe:

```
{  
  "ParameterGroup": {  
    "ParameterGroupName": "daxparametergroup",  
    "Description": "A new parameter group"  
  }  
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [CreateParameterGroup](#) in der AWS CLI Befehlsreferenz.

create-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `create-subnet-group`.

AWS CLI

Um eine DAX Subnetzgruppe zu erstellen

Im folgenden `create-subnet-group` Beispiel wird eine Subnetzgruppe mit den angegebenen Einstellungen erstellt.

```
aws dax create-subnet-group \  
  --subnet-group-name daxSubnetGroup \  
  --subnet-ids subnet-11111111 subnet-22222222
```

Ausgabe:

```
{  
  "SubnetGroup": {  
    "SubnetGroupName": "daxSubnetGroup",  
    "VpcId": "vpc-05a1fa8e00c325226",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-11111111",
```



```

        "SubnetAvailabilityZone": "us-west-2b"
      },
      {
        "SubnetIdentifier": "subnet-22222222",
        "SubnetAvailabilityZone": "us-west-2c"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Schritt 2: Eine Subnetzgruppe erstellen](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateSubnetGroup](#).AWS CLI

decrease-replication-factor

Das folgende Codebeispiel zeigt die Verwendung `decrease-replication-factor`.

AWS CLI

Um einen oder mehrere Knoten aus dem Cluster zu entfernen

Im folgenden `decrease-replication-factor` Beispiel wird die Anzahl der Knoten im angegebenen DAX Cluster auf einen reduziert.

```

aws dax decrease-replication-factor \
  --cluster-name daxcluster \
  --new-replication-factor 1

```

Ausgabe:

```

{
  "Cluster": {
    "ClusterName": "daxcluster",
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
    "TotalNodes": 3,
    "ActiveNodes": 3,
    "NodeType": "dax.r4.large",
    "Status": "modifying",
    "ClusterDiscoveryEndpoint": {
      "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
      "Port": 8111
    }
  }
}

```

```
    },
    "Nodes": [
      {
        "NodeId": "daxcluster-a",
        "Endpoint": {
          "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
          "Port": 8111
        },
        "NodeCreateTime": 1576625059.509,
        "AvailabilityZone": "us-west-2c",
        "NodeStatus": "available",
        "ParameterGroupStatus": "in-sync"
      },
      {
        "NodeId": "daxcluster-b",
        "Endpoint": {
          "Address": "daxcluster-
b.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
          "Port": 8111
        },
        "NodeCreateTime": 1576625059.509,
        "AvailabilityZone": "us-west-2a",
        "NodeStatus": "available",
        "ParameterGroupStatus": "in-sync"
      },
      {
        "NodeId": "daxcluster-c",
        "Endpoint": {
          "Address": "daxcluster-
c.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
          "Port": 8111
        },
        "NodeCreateTime": 1576625059.509,
        "AvailabilityZone": "us-west-2b",
        "NodeStatus": "available",
        "ParameterGroupStatus": "in-sync"
      }
    ],
    "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
    "SubnetGroup": "default",
    "SecurityGroups": [
      {
        "SecurityGroupIdentifier": "sg-1af6e36e",
```

```

        "Status": "active"
      }
    ],
    "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
    "ParameterGroup": {
      "ParameterGroupName": "default.dax1.0",
      "ParameterApplyStatus": "in-sync",
      "NodeIdsToReboot": []
    },
    "SSEDescription": {
      "Status": "ENABLED"
    }
  }
}

```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DecreaseReplicationFactor](#) in der AWS CLI Befehlsreferenz.

delete-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster`.

AWS CLI

Um einen DAX Cluster zu löschen

Im folgenden `delete-cluster` Beispiel wird der angegebene DAX Cluster gelöscht.

```

aws dax delete-cluster \
  --cluster-name daxcluster

```

Ausgabe:

```

{
  "Cluster": {
    "ClusterName": "daxcluster",
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
    "TotalNodes": 3,
    "ActiveNodes": 0,
    "NodeType": "dax.r4.large",

```

```
"Status": "deleting",
"ClusterDiscoveryEndpoint": {
  "Address": "dd.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
  "Port": 8111
},
"PreferredMaintenanceWindow": "fri:06:00-fri:07:00",
"SubnetGroup": "default",
"SecurityGroups": [
  {
    "SecurityGroupIdentifier": "sg-1af6e36e",
    "Status": "active"
  }
],
"IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
"ParameterGroup": {
  "ParameterGroupName": "default.dax1.0",
  "ParameterApplyStatus": "in-sync",
  "NodeIdsToReboot": []
},
"SSEDescription": {
  "Status": "ENABLED"
}
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DeleteCluster](#) in der AWS CLI Befehlsreferenz.

delete-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `delete-parameter-group`.

AWS CLI

Um eine Parametergruppe zu löschen

Im folgenden `delete-parameter-group` Beispiel wird die angegebene DAX Parametergruppe gelöscht.

```
aws dax delete-parameter-group \
```

```
--parameter-group-name daxparametergroup
```

Ausgabe:

```
{
  "DeletionMessage": "Parameter group daxparametergroup has been deleted."
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DeleteParameterGroup](#) in der AWS CLI Befehlsreferenz.

delete-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `delete-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu löschen

Im folgenden `delete-subnet-group` Beispiel wird die angegebene DAX Subnetzgruppe gelöscht.

```
aws dax delete-subnet-group \  
  --subnet-group-name daxSubnetGroup
```

Ausgabe:

```
{
  "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DeleteSubnetGroup](#) in der AWS CLI Befehlsreferenz.

describe-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-clusters`.

AWS CLI

Um Informationen zu allen bereitgestellten DAX Clustern zurückzugeben

Im folgenden `describe-clusters` Beispiel werden Details zu allen bereitgestellten Clustern DAX angezeigt.

```
aws dax describe-clusters
```

Ausgabe:

```
{
  "Clusters": [
    {
      "ClusterName": "daxcluster",
      "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
      "TotalNodes": 1,
      "ActiveNodes": 1,
      "NodeType": "dax.r4.large",
      "Status": "available",
      "ClusterDiscoveryEndpoint": {
        "Address":
"daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
        "Port": 8111
      },
      "Nodes": [
        {
          "NodeId": "daxcluster-a",
          "Endpoint": {
            "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
            "Port": 8111
          },
          "NodeCreateTime": 1576625059.509,
          "AvailabilityZone": "us-west-2c",
          "NodeStatus": "available",
          "ParameterGroupStatus": "in-sync"
        }
      ],
      "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
      "SubnetGroup": "default",
      "SecurityGroups": [
        {
```

```

        "SecurityGroupIdentifier": "sg-1af6e36e",
        "Status": "active"
      }
    ],
    "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
    "ParameterGroup": {
      "ParameterGroupName": "default.dax1.0",
      "ParameterApplyStatus": "in-sync",
      "NodeIdsToReboot": []
    },
    "SSEDescription": {
      "Status": "ENABLED"
    }
  }
]
}

```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DescribeClusters](#) in der AWS CLI Befehlsreferenz.

describe-default-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-default-parameters`.

AWS CLI

Um die Standardsystemparameterinformationen für zurückzugeben DAX

Im folgenden `describe-default-parameters` Beispiel werden die Standardsystemparameterinformationen für angezeigt DAX.

```
aws dax describe-default-parameters
```

Ausgabe:

```

{
  "Parameters": [
    {
      "ParameterName": "query-ttl-millis",
      "ParameterType": "DEFAULT",

```

```

        "ParameterValue": "300000",
        "NodeTypeSpecificValues": [],
        "Description": "Duration in milliseconds for queries to remain cached",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": "TRUE",
        "ChangeType": "IMMEDIATE"
    },
    {
        "ParameterName": "record-ttl-millis",
        "ParameterType": "DEFAULT",
        "ParameterValue": "300000",
        "NodeTypeSpecificValues": [],
        "Description": "Duration in milliseconds for records to remain valid in
cache (Default: 0 = infinite)",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": "TRUE",
        "ChangeType": "IMMEDIATE"
    }
]
}

```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DescribeDefaultParameters](#) in der AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um alle Ereignisse zurückzugeben, die sich auf DAX Cluster und Parametergruppen beziehen

Im folgenden `describe-events` Beispiel werden Details zu Ereignissen angezeigt, die sich auf DAX Cluster und Parametergruppen beziehen.

```
aws dax describe-events
```


Ausgabe:

```
{
  "Events": [
    {
      "SourceName": "daxcluster",
      "SourceType": "CLUSTER",
      "Message": "Cluster deleted.",
      "Date": 1576702736.706
    },
    {
      "SourceName": "daxcluster",
      "SourceType": "CLUSTER",
      "Message": "Removed node daxcluster-b.",
      "Date": 1576702691.738
    },
    {
      "SourceName": "daxcluster",
      "SourceType": "CLUSTER",
      "Message": "Removed node daxcluster-a.",
      "Date": 1576702633.498
    },
    {
      "SourceName": "daxcluster",
      "SourceType": "CLUSTER",
      "Message": "Removed node daxcluster-c.",
      "Date": 1576702631.329
    },
    {
      "SourceName": "daxcluster",
      "SourceType": "CLUSTER",
      "Message": "Cluster created.",
      "Date": 1576626560.057
    }
  ]
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

describe-parameter-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-parameter-groups`.

AWS CLI

Zur Beschreibung der Parametergruppen, die in definiert sind DAX

Im folgenden `describe-parameter-groups` Beispiel werden Details zu den Parametergruppen abgerufen, die in DAX definiert sind.

```
aws dax describe-parameter-groups
```

Ausgabe:

```
{
  "ParameterGroups": [
    {
      "ParameterGroupName": "default.dax1.0",
      "Description": "Default parameter group for dax1.0"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DescribeParameterGroups](#) in der AWS CLI Befehlsreferenz.

describe-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-parameters`.

AWS CLI

Um die in einer DAX Parametergruppe definierten Parameter zu beschreiben

Im folgenden `describe-parameters` Beispiel werden Details zu den Parametern abgerufen, die in der angegebenen DAX Parametergruppe definiert sind.

```
aws dax describe-parameters \
```

```
--parameter-group-name default.dax1.0
```

Ausgabe:

```
{
  "Parameters": [
    {
      "ParameterName": "query-ttl-millis",
      "ParameterType": "DEFAULT",
      "ParameterValue": "300000",
      "NodeTypeSpecificValues": [],
      "Description": "Duration in milliseconds for queries to remain cached",
      "Source": "user",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": "TRUE",
      "ChangeType": "IMMEDIATE"
    },
    {
      "ParameterName": "record-ttl-millis",
      "ParameterType": "DEFAULT",
      "ParameterValue": "300000",
      "NodeTypeSpecificValues": [],
      "Description": "Duration in milliseconds for records to remain valid in
cache (Default: 0 = infinite)",
      "Source": "user",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": "TRUE",
      "ChangeType": "IMMEDIATE"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DescribeParameters](#) in der AWS CLI Befehlsreferenz.

describe-subnet-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-subnet-groups`.

AWS CLI

Zur Beschreibung von Subnetzgruppen, die in definiert sind DAX

Im folgenden `describe-subnet-groups` Beispiel werden Details für die in definierten Subnetzgruppen abgerufen. DAX

```
aws dax describe-subnet-groups
```

Ausgabe:

```
{
  "SubnetGroups": [
    {
      "SubnetGroupName": "default",
      "Description": "Default CacheSubnetGroup",
      "VpcId": "vpc-ee70a196",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-874953af",
          "SubnetAvailabilityZone": "us-west-2d"
        },
        {
          "SubnetIdentifier": "subnet-bd3d1fc4",
          "SubnetAvailabilityZone": "us-west-2a"
        },
        {
          "SubnetIdentifier": "subnet-72c2ff28",
          "SubnetAvailabilityZone": "us-west-2c"
        },
        {
          "SubnetIdentifier": "subnet-09e6aa42",
          "SubnetAvailabilityZone": "us-west-2b"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [DescribeSubnetGroups](#) in der AWS CLI Befehlsreferenz.

increase-replication-factor

Das folgende Codebeispiel zeigt die Verwendung `increase-replication-factor`.

AWS CLI

Um den Replikationsfaktor für einen DAX Cluster zu erhöhen

Im folgenden `increase-replication-factor` Beispiel wird der Replikationsfaktor des angegebenen DAX Clusters auf 3 erhöht.

```
aws dax increase-replication-factor \  
  --cluster-name daxcluster \  
  --new-replication-factor 3
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterName": "daxcluster",  
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",  
    "TotalNodes": 3,  
    "ActiveNodes": 1,  
    "NodeType": "dax.r4.large",  
    "Status": "modifying",  
    "ClusterDiscoveryEndpoint": {  
      "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",  
      "Port": 8111  
    },  
    "Nodes": [  
      {  
        "NodeId": "daxcluster-a",  
        "Endpoint": {  
          "Address": "daxcluster-  
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",  
          "Port": 8111  
        },  
        "NodeCreateTime": 1576625059.509,  
        "AvailabilityZone": "us-west-2c",  
        "NodeStatus": "available",  
        "ParameterGroupStatus": "in-sync"  
      },  
      {
```

```
        "NodeId": "daxcluster-b",
        "NodeStatus": "creating"
    },
    {
        "NodeId": "daxcluster-c",
        "NodeStatus": "creating"
    }
],
"PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
"SubnetGroup": "default",
"SecurityGroups": [
    {
        "SecurityGroupIdentifier": "sg-1af6e36e",
        "Status": "active"
    }
],
"IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
"ParameterGroup": {
    "ParameterGroupName": "default.dax1.0",
    "ParameterApplyStatus": "in-sync",
    "NodeIdsToReboot": []
},
"SSEDescription": {
    "Status": "ENABLED"
}
}
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [IncreaseReplicationFactor](#) in der AWS CLI Befehlsreferenz.

list-tags

Das folgende Codebeispiel zeigt die Verwendung `list-tags`.

AWS CLI

Um Tags auf einer DAX Ressource aufzulisten

Das folgende `list-tags` Beispiel listet die Tag-Schlüssel und -Werte auf, die dem angegebenen DAX Cluster zugeordnet sind.

```
aws dax list-tags \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "ClusterUsage",  
      "Value": "prod"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [ListTags](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine DAX Ressource zu taggen

Im folgenden `tag-resource` Beispiel werden der angegebene Tag-Schlüsselname und der zugehörige Wert an den angegebenen DAX Cluster angehängt, um die Clusternutzung zu beschreiben.

```
aws dax tag-resource \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \  
  --tags="Key=ClusterUsage,Value=prod"
```

Ausgabe:

```
{
```

```
"Tags": [  
  {  
    "Key": "ClusterUsage",  
    "Value": "prod"  
  }  
]
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer DAX-Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem angegebenen Schlüsselnamen aus einem DAX Cluster entfernt.

```
aws dax untag-resource \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \  
  --tag-keys="ClusterUsage"
```

Ausgabe:

```
{  
  "Tags": []  
}
```

Weitere Informationen finden Sie unter [Managing DAX Clusters](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

Beispiele für Detective mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Detective Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-invitation

Das folgende Codebeispiel zeigt die Verwendung `accept-invitation`.

AWS CLI

Um eine Einladung anzunehmen, ein Mitgliedskonto zu werden, in einem Verhaltensdiagramm

Das folgende `accept-invitation` Beispiel akzeptiert eine Einladung, ein Mitgliedskonto zu werden, im Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234`.

```
aws detective accept-invitation \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auf eine Einladung mit einem Verhaltensdiagramm antworten](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [AcceptInvitation](#) unter AWS CLI Befehlsreferenz.

create-graph

Das folgende Codebeispiel zeigt die Verwendung `create-graph`.

AWS CLI

Um Amazon Detective zu aktivieren und ein neues Verhaltensdiagramm zu erstellen

Im folgenden `create-graph` Beispiel wird Detective für das AWS Konto aktiviert, das den Befehl in der Region ausführt, in der der Befehl ausgeführt wird. Es wird ein neues Verhaltensdiagramm erstellt, das dieses Konto als Administratorkonto verwendet. Der Befehl weist dem Department-Tag außerdem den Wert Finanzen zu.

```
aws detective create-graph \  
  --tags '{"Department": "Finance"}'
```

Ausgabe:

```
{  
  "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"  
}
```

Weitere Informationen finden Sie unter [Amazon Detective aktivieren](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [CreateGraph](#) in der AWS CLI Befehlsreferenz.

create-members

Das folgende Codebeispiel zeigt die Verwendung `create-members`.

AWS CLI

Um Mitgliedskonten zu einem Verhaltensdiagramm einzuladen

Im folgenden `create-members` Beispiel werden zwei AWS Konten eingeladen, Mitgliedskonten im Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` zu werden. Für jedes Konto enthält AWS die Anfrage die Konto-ID und die E-Mail-Adresse des Root-Benutzers des Kontos. Die Anfrage enthält eine benutzerdefinierte Nachricht, die in die Einladungs-E-Mail eingefügt werden kann.

```
aws detective create-members \
  --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
\
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
  --message "This is Paul Santos. I need to add your account to the data we use
for security investigation in Amazon Detective. If you have any questions, contact
me at psantos@example.com."
```

Ausgabe:

```
{
  "Members": [
    {
      "AccountId": "444455556666",
      "AdministratorId": "111122223333",
      "EmailAddress": "mmajor@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "INVITED",
      "UpdatedTime": 1579826107000
    },
    {
      "AccountId": "123456789012",
      "AdministratorId": "111122223333",
      "EmailAddress": "jstiles@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "VERIFICATION_IN_PROGRESS",
      "UpdatedTime": 1579826107000
    }
  ],
  "UnprocessedAccounts": [ ]
}
```

Weitere Informationen finden Sie unter Mitgliedskonten zu einem Verhaltensdiagramm einladen<
<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>> im Amazon Detective Administration Guide.

So laden Sie Mitgliedskonten ein, ohne Einladungs-E-Mails zu senden

Im folgenden `create-members` Beispiel werden zwei AWS Konten eingeladen, Mitgliedskonten im Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` zu werden. Für jedes Konto enthält AWS die Anfrage die Konto-ID und die E-Mail-Adresse des Root-Benutzers des Kontos. Die Mitgliedskonten erhalten keine Einladungs-E-Mails.

```
aws detective create-members \
  --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
\
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
  --disable-email-notification
```

Ausgabe:

```
{
  "Members": [
    {
      "AccountId": "444455556666",
      "AdministratorId": "111122223333",
      "EmailAddress": "mmajor@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "INVITED",
      "UpdatedTime": 1579826107000
    },
    {
      "AccountId": "123456789012",
      "AdministratorId": "111122223333",
      "EmailAddress": "jstiles@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "VERIFICATION_IN_PROGRESS",
      "UpdatedTime": 1579826107000
    }
  ],
  "UnprocessedAccounts": [ ]
}
```

Weitere Informationen finden Sie unter [Mitgliedskonten zu einem Verhaltensdiagramm einladen](https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html)< <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>> im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [CreateMembers](#) in AWS CLI der Befehlsreferenz.

delete-graph

Das folgende Codebeispiel zeigt die Verwendung `delete-graph`.

AWS CLI

Um Detective zu deaktivieren und das Verhaltensdiagramm zu löschen

Das folgende `delete-graph` Beispiel deaktiviert Detective und löscht das angegebene Verhaltensdiagramm.

```
aws detective delete-graph \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon Detective deaktivieren](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [DeleteGraph](#) in der AWS CLI Befehlsreferenz.

delete-members

Das folgende Codebeispiel zeigt die Verwendung `delete-members`.

AWS CLI

Um Mitgliedskonten aus einem Verhaltensdiagramm zu entfernen

Im folgenden `delete-members` Beispiel werden zwei Mitgliedskonten aus dem Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` entfernt. Um die Konten zu AWS identifizieren IDs, wird in der Anfrage das Konto bereitgestellt.

```
aws detective delete-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Ausgabe:

```
{
  "AccountIds": [ "444455556666", "123456789012" ],
  "UnprocessedAccounts": [ ]
}
```

Weitere Informationen finden Sie unter Mitgliedskonten aus einem Verhaltensdiagramm entfernen < <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html> > im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [DeleteMembers](#) in AWS CLI der Befehlsreferenz.

disassociate-membership

Das folgende Codebeispiel zeigt die Verwendung `disassociate-membership`.

AWS CLI

So kündigen Sie die Mitgliedschaft in einem Verhaltensdiagramm

Im folgenden Beispiel für eine Disassociate-Mitgliedschaft wird das AWS Konto, das den Befehl ausführt, aus dem Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` entfernt.

```
aws detective disassociate-membership \
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Weitere Informationen finden Sie unter Entfernen Ihres Kontos aus einem Verhaltensdiagramm < <https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html> > im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [DisassociateMembership](#) in AWS CLI der Befehlsreferenz.

get-members

Das folgende Codebeispiel zeigt die Verwendung `get-members`.

AWS CLI

Um Informationen über ausgewähltes Verhalten abzurufen, können Sie Mitgliedskonten grafisch darstellen

Im folgenden `get-members` Beispiel werden Informationen über zwei Mitgliedskonten im Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` abgerufen. Für die beiden Konten AWS stellt IDs die Anfrage das Konto bereit.

```
aws detective get-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Ausgabe:

```
{  
  "MemberDetails": [  
    {  
      "AccountId": "444455556666",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "mmajor@example.com",  
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "INVITED",  
      "UpdatedTime": 1579826107000  
    }  
    {  
      "AccountId": "123456789012",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "jstiles@example.com",  
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "INVITED",  
      "UpdatedTime": 1579826107000  
    }  
  ],  
  "UnprocessedAccounts": [ ]  
}
```

Weitere Informationen finden Sie unter [Kontenliste in einem Verhaltensdiagramm anzeigen](https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html) <<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html>> im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [GetMembers](#) in AWS CLI der Befehlsreferenz.

list-graphs

Das folgende Codebeispiel zeigt die Verwendung `list-graphs`.

AWS CLI

Um eine Liste von Verhaltensdiagrammen anzuzeigen, für die Ihr Konto der Administrator ist

Im folgenden `list-graphs` Beispiel werden die Verhaltensdiagramme abgerufen, für die das anrufende Konto der Administrator in der aktuellen Region ist.

```
aws detective list-graphs
```

Ausgabe:

```
{
  "GraphList": [
    {
      "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "CreatedTime": 1579736111000
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListGraphs AWS CLI Befehlsreferenz](#).

list-invitations

Das folgende Codebeispiel zeigt die Verwendung `list-invitations`.

AWS CLI

Um eine Liste von Verhaltensdiagrammen anzuzeigen, bei denen ein Account Mitglied ist oder zu denen er eingeladen wurde

Im folgenden `list-invitations` Beispiel werden die Verhaltensdiagramme abgerufen, zu denen das anrufende Konto eingeladen wurde. Die Ergebnisse enthalten nur offene und angenommene Einladungen. Sie enthalten keine abgelehnten Einladungen oder gelöschte Mitgliedschaften.

```
aws detective list-invitations
```


Ausgabe:

```
{
  "Invitations": [
    {
      "AccountId": "4444455556666",
      "AdministratorId": "111122223333",
      "EmailAddress": "mmajor@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "INVITED",
      "UpdatedTime": 1579826107000
    }
  ]
}
```

Weitere Informationen finden Sie unter Ihre Liste mit Einladungen zum Verhaltensdiagramm anzeigen < <https://docs.aws.amazon.com/detective/latest/adminguide/member-view-graph-invitations.html> > im Amazon Detective Administration Guide.

- API-Einheiten finden Sie in der Befehlsreferenz [ListInvitations](#).AWS CLI

list-members

Das folgende Codebeispiel zeigt die Verwendung `list-members`.

AWS CLI

Um die Mitgliedskonten in einem Verhaltensdiagramm aufzulisten

Im folgenden `list-members` Beispiel werden die eingeladenen und aktivierten Mitgliedskonten für das Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` abgerufen. Die Ergebnisse beinhalten keine Mitgliedskonten, die entfernt wurden.

```
aws detective list-members \
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Ausgabe:

```
{
```

```
"MemberDetails": [  
  {  
    "AccountId": "444455556666",  
    "AdministratorId": "111122223333",  
    "EmailAddress": "mmajor@example.com",  
    "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:123412341234",  
    "InvitedTime": 1579826107000,  
    "MasterId": "111122223333",  
    "Status": "INVITED",  
    "UpdatedTime": 1579826107000  
  },  
  {  
    "AccountId": "123456789012",  
    "AdministratorId": "111122223333",  
    "EmailAddress": "jstiles@example.com",  
    "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:123412341234",  
    "InvitedTime": 1579826107000,  
    "MasterId": "111122223333",  
    "PercentOfGraphUtilization": 2,  
    "PercentOfGraphUtilizationUpdatedTime": 1586287843,  
    "Status": "ENABLED",  
    "UpdatedTime": 1579973711000,  
    "VolumeUsageInBytes": 200,  
    "VolumeUsageUpdatedTime": 1586287843  
  }  
]  
}
```

Weitere Informationen finden Sie unter [Kontenliste in einem Verhaltensdiagramm anzeigen](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [ListMembers](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die einem Verhaltensdiagramm zugewiesenen Tags abzurufen

Das folgende `list-tags-for-resource` Beispiel gibt die Tags zurück, die dem angegebenen Verhaltensgraphen zugewiesen sind.

```
aws detective list-tags-for-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Ausgabe:

```
{  
  "Tags": {  
    "Department" : "Finance"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Tags für ein Verhaltensdiagramm](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

reject-invitation

Das folgende Codebeispiel zeigt die Verwendung `reject-invitation`.

AWS CLI

Um eine Einladung, ein Mitgliedskonto zu werden, in einem Verhaltensdiagramm abzulehnen

Im folgenden `reject-invitation` Beispiel wird eine Einladung, ein Mitgliedskonto zu werden, im Verhaltensdiagramm `arn:aws:detective:us-east-1:111122223333:graph:123412341234` abgelehnt.

```
aws detective reject-invitation \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auf eine Einladung mit einem Verhaltensdiagramm antworten < https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html >](https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html) im Amazon Detective Administration Guide.

- API-Einheiten [RejectInvitation AWS CLI](#) finden Sie in der Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer Ressource ein Tag zuzuweisen

Im folgenden `tag-resource` Beispiel wird dem angegebenen Verhaltensdiagramm ein Wert für das Department-Tag zugewiesen.

```
aws detective tag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tags '{"Department":"Finance"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Tags für ein Verhaltensdiagramm](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um einen Tag-Wert aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Department-Tag aus dem angegebenen Verhaltensdiagramm entfernt.

```
aws detective untag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tag-keys "Department"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Tags für ein Verhaltensdiagramm](#) im Amazon Detective Administration Guide.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

Beispiele für Device Farm mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Device Farm Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-device-pool

Das folgende Codebeispiel zeigt die Verwendung `create-device-pool`.

AWS CLI

Um einen Gerätepool zu erstellen

Der folgende Befehl erstellt einen Android-Gerätepool für ein Projekt:

```
aws devicefarm create-device-pool --name pool1 --rules file://  
device-pool-rules.json --project-arn "arn:aws:devicefarm:us-  
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506"
```

Sie können das Projekt ARN aus der Ausgabe von `create-project` oder abrufen `list-projects`. Die Datei `device-pool-rules.json` ist ein JSON Dokument im aktuellen Ordner, das die Geräteplattform angibt:

```
[  
  {  
    "attribute": "PLATFORM",  
    "operator": "EQUALS",  
    "value": "\"ANDROID\""  }  
]
```

```
}
]
```

Ausgabe:

```
{
  "devicePool": {
    "rules": [
      {
        "operator": "EQUALS",
        "attribute": "PLATFORM",
        "value": "\"ANDROID\""
      }
    ],
    "type": "PRIVATE",
    "name": "pool1",
    "arn": "arn:aws:devicefarm:us-west-2:123456789012:devicepool:070fc3ca-7ec1-4741-9c1f-d3e044efc506/2aa8d2a9-5e73-47ca-b929-659cb34b7dcd"
  }
}
```

- API-Einheiten finden Sie [CreateDevicePool](#) in der AWS CLI Befehlsreferenz.

create-project

Das folgende Codebeispiel zeigt die Verwendung `create-project`.

AWS CLI

Um ein Projekt zu erstellen

Der folgende Befehl erstellt ein neues Projekt mit dem Namen `my-project`:

```
aws devicefarm create-project --name my-project
```

Ausgabe:

```
{
  "project": {
    "name": "myproject",
  }
}
```

```

    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
    "created": 1503612890.057
  }
}

```

- API-Einheiten finden Sie [CreateProject](#) in der AWS CLI Befehlsreferenz.

create-upload

Das folgende Codebeispiel zeigt die Verwendung `create-upload`.

AWS CLI

Um einen Upload zu erstellen

Der folgende Befehl erstellt einen Upload für eine Android-App:

```

aws devicefarm create-upload --project-arn "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506" --name app.apk --
type ANDROID_APP

```

Sie können das Projekt ARN aus der Ausgabe von `create-project` oder `list-projects` abrufen.

Ausgabe:

```

{
  "upload": {
    "status": "INITIALIZED",
    "name": "app.apk",
    "created": 1503614408.769,
    "url": "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs%32Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f",
    "type": "ANDROID_APP",
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
  }
}

```

```
}
}
```

Verwenden Sie das URL in der Ausgabe signierte Zeichen, um eine Datei auf Device Farm hochzuladen:

```
curl -T app.apk "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f"
```

- API-Einheiten finden Sie [CreateUpload](#) in der AWS CLI Befehlsreferenz.

get-upload

Das folgende Codebeispiel zeigt die Verwendung `get-upload`.

AWS CLI

Um einen Upload anzusehen

Der folgende Befehl ruft Informationen über einen Upload ab:

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-ae9e-4087-09e6-f4cea3599514"
```

Sie können den Upload ARN aus der Ausgabe von `create-upload` abrufen.

Ausgabe:

```
{
  "upload": {
    "status": "SUCCEEDED",
    "name": "app.apk",
    "created": 1505262773.186,
    "type": "ANDROID_APP",
```



```

    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514",
    "metadata": {"device_admin":false,"activity_name":
\\"com.example.client.LauncherActivity\\",\\"version_name\\":\\"1.0.2.94\\",\\"screens
\\":[\\"small\\",\\"normal\\",\\"large\\",\\"xlarge\\"],\\"error_type\\":null,\\"sdk_version
\\":\\"16\\",\\"package_name\\":\\"com.example.client\\",\\"version_code\\":\\"20994\\",
\\"native_code\\":[\\"armeabi-v7a\\"],\\"target_sdk_version\\":\\"25\\"}"}
  }
}

```

- API-Einheiten finden Sie [GetUpload](#) in der AWS CLI Befehlsreferenz.

list-projects

Das folgende Codebeispiel zeigt die Verwendung `list-projects`.

AWS CLI

Um Projekte aufzulisten

Im Folgenden wird eine Liste von Projekten abgerufen:

```
aws devicefarm list-projects
```

Ausgabe:

```

{
  "projects": [
    {
      "name": "myproject",
      "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
      "created": 1503612890.057
    },
    {
      "name": "otherproject",
      "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:a5f5b752-8098-49d1-86bf-5f7682c1c77e",
      "created": 1505257519.337
    }
  ]
}

```

```
}
```

- API-Einheiten finden Sie [ListProjects](#) in der AWS CLI Befehlsreferenz.

AWS Direct Connect Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Direct Connect.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-direct-connect-gateway-association-proposal

Das folgende Codebeispiel zeigt die Verwendung `accept-direct-connect-gateway-association-proposal`.

AWS CLI

Um einen Vorschlag für eine Gateway-Zuordnung anzunehmen

Im Folgenden wird der angegebene Vorschlag `accept-direct-connect-gateway-association-proposal` akzeptiert.

```
aws directconnect accept-direct-connect-gateway-association-proposal \  
  --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \  
  --proposal-id cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE \  
  --associated-gateway-owner-account 111122223333  
  
{  
  "directConnectGatewayAssociation": {  
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
```

```

"directConnectGatewayOwnerAccount": "111122223333",
"associationState": "associating",
"associatedGateway": {
  "id": "tgw-02f776b1a7EXAMPLE",
  "type": "transitGateway",
  "ownerAccount": "111122223333",
  "region": "us-east-1"
},
"associationId": "6441f8bf-5917-4279-ade1-9708bEXAMPLE",
"allowedPrefixesToDirectConnectGateway": [
  {
    "cidr": "192.168.1.0/30"
  }
]
}
}

```

Weitere Informationen finden Sie unter [Akzeptieren oder Ablehnen eines Transit Gateway Gateway-Zuordnungsvorschlags](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [AcceptDirectConnectGatewayAssociationProposal](#) in der AWS CLI Befehlsreferenz.

allocate-connection-on-interconnect

Das folgende Codebeispiel zeigt die Verwendung `allocate-connection-on-interconnect`.

AWS CLI

Um eine gehostete Verbindung auf einer Interconnect-Verbindung zu erstellen

Mit dem folgenden `allocate-connection-on-interconnect` Befehl wird eine gehostete Verbindung auf einer Interconnect-Verbindung erstellt:

```

aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101

```

Ausgabe:

```

{
  "partnerName": "TIVIT",

```

```

    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
  }

```

- API-Einheiten finden Sie [AllocateConnectionOnInterconnect](#) in der AWS CLI Befehlsreferenz.

allocate-hosted-connection

Das folgende Codebeispiel zeigt die Verwendung `allocate-hosted-connection`.

AWS CLI

Um eine gehostete Verbindung auf einer Interconnect-Verbindung zu erstellen

Im folgenden `allocate-hosted-connection` Beispiel wird eine gehostete Verbindung auf der angegebenen Verbindungsleitung erstellt.

```

aws directconnect allocate-hosted-connection \
  --bandwidth 500Mbps \
  --connection-name mydcinterconnect \
  --owner-account 123456789012 \
  -connection-id dxcon-fgktov66 \
  -vlan 101

```

Ausgabe:

```

{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffzc51m1",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}

```

```
}

```

- API-Einheiten finden Sie unter [AllocateHostedConnection AWS CLI](#) Befehlsreferenz.

allocate-private-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `allocate-private-virtual-interface`.

AWS CLI

Um eine private virtuelle Schnittstelle bereitzustellen

Mit dem folgenden `allocate-private-virtual-interface` Befehl wird eine private virtuelle Schnittstelle bereitgestellt, die einem anderen Kunden gehört:

```
aws directconnect allocate-private-virtual-interface --connection-id dxcon-ffjrnx17 --owner-account 123456789012 --new-private-virtual-interface-allocation virtualInterfaceName=PrivateVirtualInterface,vlan=1000,asn=65000,authKey=asdf34example
```

Ausgabe:

```
{
  "virtualInterfaceState": "confirming",
  "asn": 65000,
  "vlan": 1000,
  "customerAddress": "192.168.1.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrnx17",
  "virtualInterfaceId": "dxvif-fgy8orxu",
  "authKey": "asdf34example",
  "routeFilterPrefixes": [],
  "location": "TIVIT",
  "customerRouterConfig": "<?xml version='1.0' encoding='UTF-8'?>
>\n <logical_connection id='dxvif-fgy8orxu'>\n <vlan>1000</
vlan>\n <customer_address>192.168.1.2/30</customer_address>\n
<amazon_address>192.168.1.1/30</amazon_address>\n <bgp_asn>65000</bgp_asn>\n
<bgp_auth_key>asdf34example</bgp_auth_key>\n <amazon_bgp_asn>7224</amazon_bgp_asn>
\n <connection_type>private</connection_type>\n</logical_connection>\n",
  "amazonAddress": "192.168.1.1/30",
  "virtualInterfaceType": "private",
  "virtualInterfaceName": "PrivateVirtualInterface"
}
```

- API-Einheiten finden Sie [AllocatePrivateVirtualInterface](#) in der AWS CLI Befehlsreferenz.

allocate-public-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `allocate-public-virtual-interface`.

AWS CLI

Um eine öffentliche virtuelle Schnittstelle bereitzustellen

Mit dem folgenden `allocate-public-virtual-interface` Befehl wird eine öffentliche virtuelle Schnittstelle bereitgestellt, die einem anderen Kunden gehört:

```
aws directconnect allocate-public-virtual-interface --connection-id dxcon-ffjrnx17 --owner-account 123456789012 --new-public-virtual-interface-allocation virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34example{cidr=203.0.113.4/30}
```

Ausgabe:

```
{
  "virtualInterfaceState": "confirming",
  "asn": 65000,
  "vlan": 2000,
  "customerAddress": "203.0.113.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrnx17",
  "virtualInterfaceId": "dxvif-fg9xo9vp",
  "authKey": "asdf34example",
  "routeFilterPrefixes": [
    {
      "cidr": "203.0.113.0/30"
    },
    {
      "cidr": "203.0.113.4/30"
    }
  ],
  "location": "TIVIT",
  "customerRouterConfig": "<?xml version='1.0' encoding='UTF-8'?>\n<logical_connection id='dxvif-fg9xo9vp'>\n  <vlan>2000</vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n  <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n"
```

```
<bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
  "amazonAddress": "203.0.113.1/30",
  "virtualInterfaceType": "public",
  "virtualInterfaceName": "PublicVirtualInterface"
}
```

- API-Einheiten finden Sie [AllocatePublicVirtualInterface](#) in der AWS CLI Befehlsreferenz.

allocate-transit-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `allocate-transit-virtual-interface`.

AWS CLI

Um eine virtuelle Transitschnittstelle bereitzustellen, die dem angegebenen AWS Konto gehört

Im folgenden `allocate-transit-virtual-interface` Beispiel wird eine virtuelle Transitschnittstelle für das angegebene Konto bereitgestellt.

```
aws directconnect allocate-transit-virtual-interface \
  --connection-id dxlag-fEXAMPLE \
  --owner-account 123456789012 \
  --new-transit-virtual-interface-allocation "virtualInterfaceName=Example Transit
Virtual
Interface, vlan=126, asn=65110, mtu=1500, authKey=0xzxgA9YoW9h58u8SEXAMPLE, amazonAddress=192.16"
```

Ausgabe:

```
{
  "virtualInterface": {
    "ownerAccount": "123456789012",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "Example Transit Virtual Interface",
    "vlan": 126,
    "asn": 65110,
    "amazonSideAsn": 7224,
    "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
```

```

    "amazonAddress": "192.168.1.1/30",
    "customerAddress": "192.168.1.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "confirming",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\\\"UTF-8\\\"?>\\n<logical_connection id=\\\"dxvif-fEXAMPLE\\\">\\n  <vlan>126</
vlan>\\n  <customer_address>192.168.1.2/30</customer_address>\\n
  <amazon_address>192.168.1.1/30</amazon_address>\\n  <bgp_asn>65110</bgp_asn>\\n
  <bgp_auth_key>0xzxcgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\\n  <connection_type>transit</connection_type>\\n</logical_connection>
\\n\",
    "mtu": 1500,
    "jumboFrameCapable": true,
    "virtualGatewayId": "",
    "directConnectGatewayId": "",
    "routeFilterPrefixes": [],
    "bgpPeers": [
      {
        "bgpPeerId": "dxpeer-fEXAMPLE",
        "asn": 65110,
        "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
        "addressFamily": "ipv4",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "bgpPeerState": "pending",
        "bgpStatus": "down",
        "awsDeviceV2": "loc1-26wz6vEXAMPLE"
      }
    ],
    "region": "sa-east-1",
    "awsDeviceV2": "loc1-26wz6vEXAMPLE",
    "tags": [
      {
        "key": "Tag",
        "value": "Example"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Creating a Hosted Transit Virtual Interface](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [AllocateTransitVirtualInterface](#) in der AWS CLI Befehlsreferenz.

associate-connection-with-lag

Das folgende Codebeispiel zeigt die Verwendung `associate-connection-with-lag`.

AWS CLI

Um eine Verbindung mit einem zu verknüpfen LAG

Im folgenden Beispiel wird die angegebene Verbindung der angegebenen Verbindung zugeordnet LAG.

Befehl:

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --  
connection-id dxcon-fg9607vm
```

Ausgabe:

```
{  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-fg9607vm",  
  "lagId": "dxlag-fhccu14t",  
  "connectionState": "requested",  
  "bandwidth": "1Gbps",  
  "location": "EqDC2",  
  "connectionName": "Con2ForLag",  
  "region": "us-east-1"  
}
```

- API-Einheiten finden Sie [AssociateConnectionWithLag](#) unter AWS CLI Befehlsreferenz.

associate-hosted-connection

Das folgende Codebeispiel zeigt die Verwendung `associate-hosted-connection`.

AWS CLI

Um eine gehostete Verbindung einem zuzuordnen LAG

Das folgende Beispiel verknüpft die angegebene gehostete Verbindung mit der angegebenen LAG.

Befehl:

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t
--connection-id dxcon-fg9607vm
```

Ausgabe:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- API-Einheiten finden Sie [AssociateHostedConnection](#) unter AWS CLI Befehlsreferenz.

associate-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `associate-virtual-interface`.

AWS CLI

Um eine virtuelle Schnittstelle mit einer Verbindung zu verknüpfen

Das folgende Beispiel ordnet die angegebene virtuelle Schnittstelle der angegebenen zuLAG. Um die virtuelle Schnittstelle einer Verbindung zuzuordnen, geben Sie alternativ die ID einer AWS Direct Connect-Verbindung für `--connection-id` an, zum Beispiel `dxcon-ffnikghc`.

Befehl:

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --
virtual-interface-id dxvif-fgputw0j
```

Ausgabe:

```
{
  "virtualInterfaceState": "pending",
}
```

```

"asn": 65000,
"vlan": 123,
"customerAddress": "169.254.255.2/30",
"ownerAccount": "123456789012",
"connectionId": "dxlag-ffjhj91x",
"addressFamily": "ipv4",
"virtualGatewayId": "vgw-38e90b51",
"virtualInterfaceId": "dxvif-fgputw0j",
"authKey": "0x123pK5_VBqv.UQ3kJ4123_",
"routeFilterPrefixes": [],
"location": "CSVA1",
"bgpPeers": [
  {
    "bgpStatus": "down",
    "customerAddress": "169.254.255.2/30",
    "addressFamily": "ipv4",
    "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
    "bgpPeerState": "deleting",
    "amazonAddress": "169.254.255.1/30",
    "asn": 65000
  },
  {
    "bgpStatus": "down",
    "customerAddress": "169.254.255.2/30",
    "addressFamily": "ipv4",
    "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
    "bgpPeerState": "pending",
    "amazonAddress": "169.254.255.1/30",
    "asn": 65000
  }
],
"customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgputw0j\">\n  <vlan>123</vlan>
\n  <customer_address>169.254.255.2/30</customer_address>\n
  <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0x123pK5_VBqv.UQ3kJ4123_</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
"amazonAddress": "169.254.255.1/30",
"virtualInterfaceType": "private",
"virtualInterfaceName": "VIF1A"
}

```

- API-Einheiten finden Sie [AssociateVirtualInterface](#) in der AWS CLI Befehlsreferenz.

confirm-connection

Das folgende Codebeispiel zeigt die Verwendung `confirm-connection`.

AWS CLI

Um die Erstellung einer gehosteten Verbindung auf einer Interconnect-Verbindung zu bestätigen

Der folgende `confirm-connection` Befehl bestätigt die Erstellung einer gehosteten Verbindung auf einer Interconnect:

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

Ausgabe:

```
{
  "connectionState": "pending"
}
```

- API-Einheiten finden Sie [ConfirmConnection](#) in der AWS CLI Befehlsreferenz.

confirm-private-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `confirm-private-virtual-interface`.

AWS CLI

Um den Besitz einer privaten virtuellen Schnittstelle zu akzeptieren

Der folgende `confirm-private-virtual-interface` Befehl akzeptiert den Besitz einer privaten virtuellen Schnittstelle, die von einem anderen Kunden erstellt wurde:

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

Ausgabe:

```
{
  "virtualInterfaceState": "pending"
}
```

- API-Einheiten finden Sie [ConfirmPrivateVirtualInterface](#) in der AWS CLI Befehlsreferenz.

confirm-public-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `confirm-public-virtual-interface`.

AWS CLI

Um den Besitz einer öffentlichen virtuellen Schnittstelle zu akzeptieren

Der folgende `confirm-public-virtual-interface` Befehl akzeptiert den Besitz einer öffentlichen virtuellen Schnittstelle, die von einem anderen Kunden erstellt wurde:

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

Ausgabe:

```
{
  "virtualInterfaceState": "verifying"
}
```

- API-Einheiten finden Sie [ConfirmPublicVirtualInterface](#) in der AWS CLI Befehlsreferenz.

confirm-transit-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `confirm-transit-virtual-interface`.

AWS CLI

Um den Besitz einer virtuellen Transitschnittstelle zu akzeptieren

Im Folgenden wird der Besitz einer virtuellen Transitschnittstelle `confirm-transit-virtual-interface` akzeptiert, die von einem anderen Kunden erstellt wurde.

```
aws directconnect confirm-transit-virtual-interface \
  --virtual-interface-id dxvif-fEXAMPLE \
  --direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```

Ausgabe:

```
{
  "virtualInterfaceState": "pending"
}
```

```
}
```

Weitere Informationen finden Sie unter [Akzeptieren einer gehosteten virtuellen Schnittstelle](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [ConfirmTransitVirtualInterface](#) in der AWS CLI Befehlsreferenz.

create-bgp-peer

Das folgende Codebeispiel zeigt die Verwendung `create-bgp-peer`.

AWS CLI

Um eine IPv6 BGP Peering-Sitzung zu erstellen

Das folgende Beispiel erstellt eine IPv6 BGP Peering-Sitzung auf einer privaten virtuellen Schnittstelle. `dxvif-fg1vuj3d` Die IPv6 Peer-Adressen werden automatisch von Amazon zugewiesen.

Befehl:

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

Ausgabe:

```
{
  "virtualInterface": {
    "virtualInterfaceState": "available",
    "asn": 65000,
    "vlan": 125,
    "customerAddress": "169.254.255.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fguhmqlc",
    "addressFamily": "ipv4",
    "virtualGatewayId": "vgw-f9eb0c90",
    "virtualInterfaceId": "dxvif-fg1vuj3d",
    "authKey": "0xC_ukbCerl6EYA0example",
    "routeFilterPrefixes": [],
    "location": "EqDC2",
    "bgpPeers": [
      {
```

```

        "bgpStatus": "down",
        "customerAddress": "169.254.255.2/30",
        "addressFamily": "ipv4",
        "authKey": "0xC_ukbCerl6EYA0uexample",
        "bgpPeerState": "available",
        "amazonAddress": "169.254.255.1/30",
        "asn": 65000
    },
    {
        "bgpStatus": "down",
        "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
        "addressFamily": "ipv6",
        "authKey": "0xS27kAIU_VHPjjAexample",
        "bgpPeerState": "pending",
        "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
        "asn": 64600
    }
],
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=
  \<UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
  vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
  <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</
  bgp_asn>\n  <bgp_auth_key>0xC_ukbCerl6EYA0uexample</bgp_auth_key>\n
  <ipv6_customer_address>2001:db8:1100:2f0:0:1:9cb4:4216/125</ipv6_customer_address>
  \n  <ipv6_amazon_address>2001:db8:1100:2f0:0:1:9cb4:4211/125</ipv6_amazon_address>\n
  <ipv6_bgp_asn>64600</ipv6_bgp_asn>\n  <ipv6_bgp_auth_key>0xS27kAIU_VHPjjAexample</
  ipv6_bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "Test"
  }
}

```

- API-Einheiten finden Sie [CreateBgpPeer](#) in der AWS CLI Befehlsreferenz.

create-connection

Das folgende Codebeispiel zeigt die Verwendung `create-connection`.

AWS CLI

So stellen Sie eine Verbindung von Ihrem Netzwerk zu einem AWS Direct Connect-Standort her

Mit dem folgenden `create-connection` Befehl wird eine Verbindung von Ihrem Netzwerk zu einem AWS Direct Connect-Standort hergestellt:

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-name "Connection to AWS"
```

Ausgabe:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "connectionName": "Connection to AWS",
  "region": "sa-east-1"
}
```

- API-Einheiten finden Sie [CreateConnection](#) in der AWS CLI Befehlsreferenz.

create-direct-connect-gateway-association-proposal

Das folgende Codebeispiel zeigt die Verwendung `create-direct-connect-gateway-association-proposal`.

AWS CLI

Um einen Vorschlag zur Verknüpfung des angegebenen Transit-Gateways mit dem angegebenen Direct Connect-Gateway zu erstellen

Im folgenden `create-direct-connect-gateway-association-proposal` Beispiel wird ein Vorschlag erstellt, der das angegebene Transit-Gateway dem angegebenen Direct Connect-Gateway zuordnet.

```
aws directconnect create-direct-connect-gateway-association-proposal \
  --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
  --direct-connect-gateway-owner-account 111122223333 \
  --gateway-id tgw-02f776b1a7EXAMPLE \
  --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.1.0/30
```


Ausgabe:

```
{
  "directConnectGatewayAssociationProposal": {
    "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "111122223333",
    "proposalState": "requested",
    "associatedGateway": {
      "id": "tgw-02f776b1a7EXAMPLE",
      "type": "transitGateway",
      "ownerAccount": "111122223333",
      "region": "us-east-1"
    },
    "requestedAllowedPrefixesToDirectConnectGateway": [
      {
        "cidr": "192.168.1.0/30"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Creating a Transit Gateway Association Proposal](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDirectConnectGatewayAssociationProposal](#) unter AWS CLI Befehlsreferenz.

create-direct-connect-gateway-association

Das folgende Codebeispiel zeigt die Verwendung `create-direct-connect-gateway-association`.

AWS CLI

So verknüpfen Sie ein Virtual Private Gateway mit einem Direct Connect-Gateway

Im folgenden Beispiel wird Virtual Private Gateway `vgw-6efe725e` mit Direct Connect Gateway verknüpft `5f294f92-bafb-4011-916d-9b0bexample`. Sie müssen den Befehl in der Region ausführen, in der sich das Virtual Private Gateway befindet.

Befehl:

```
aws directconnect create-direct-connect-gateway-association --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

Ausgabe:

```
{
  "directConnectGatewayAssociation": {
    "associationState": "associating",
    "virtualGatewayOwnerAccount": "123456789012",
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
    "virtualGatewayId": "vgw-6efe725e",
    "virtualGatewayRegion": "us-east-2"
  }
}
```

- API-Einheiten finden Sie [CreateDirectConnectGatewayAssociation](#) in der AWS CLI Befehlsreferenz.

create-direct-connect-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-direct-connect-gateway`.

AWS CLI

So erstellen Sie ein Direct Connect-Gateway

Im folgenden Beispiel wird ein Direct Connect-Gateway mit dem Namen `DxGateway1` erstellt.

Befehl:

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-name "DxGateway1"
```

Ausgabe:

```
{
  "directConnectGateway": {
    "amazonSideAsn": 64512,
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
    "ownerAccount": "123456789012",
    "directConnectGatewayName": "DxGateway1",
  }
}
```

```
    "directConnectGatewayState": "available"
  }
}
```

- API-Einheiten finden Sie [CreateDirectConnectGateway](#) unter AWS CLI Befehlsreferenz.

create-interconnect

Das folgende Codebeispiel zeigt die Verwendung `create-interconnect`.

AWS CLI

Um eine Verbindung zwischen dem Netzwerk eines Partners herzustellen und AWS

Mit dem folgenden `create-interconnect` Befehl wird eine Verbindung zwischen dem Netzwerk eines AWS Direct Connect-Partners und einem bestimmten AWS Direct Connect-Standort hergestellt:

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
--bandwidth 1Gbps --location TIVIT
```

Ausgabe:

```
{
  "region": "sa-east-1",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "interconnectName": "1G Interconnect to AWS",
  "interconnectId": "dxcon-fgktov66",
  "interconnectState": "requested"
}
```

- API-Einheiten finden Sie [CreateInterconnect](#) in der AWS CLI Befehlsreferenz.

create-lag

Das folgende Codebeispiel zeigt die Verwendung `create-lag`.

AWS CLI

Um eine LAG mit neuen Verbindungen zu erstellen

Das folgende Beispiel erstellt eine LAG und fordert zwei neue AWS Direct Connect-Verbindungen für die LAG mit einer Bandbreite von 1 Gbit/s an.

Befehl:

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --  
connections-bandwidth 1Gbps --lag-name 1GBLag
```

Ausgabe:

```
{  
  "awsDevice": "CSVA1-23u8t1paz8iks",  
  "numberOfConnections": 2,  
  "lagState": "pending",  
  "ownerAccount": "123456789012",  
  "lagName": "1GBLag",  
  "connections": [  
    {  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-ffqr6x5q",  
      "lagId": "dxlag-ffjhj91x",  
      "connectionState": "requested",  
      "bandwidth": "1Gbps",  
      "location": "CSVA1",  
      "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj91x",  
      "region": "us-east-1"  
    },  
    {  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-fflqyj95",  
      "lagId": "dxlag-ffjhj91x",  
      "connectionState": "requested",  
      "bandwidth": "1Gbps",  
      "location": "CSVA1",  
      "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj91x",  
      "region": "us-east-1"  
    }  
  ],  
  "lagId": "dxlag-ffjhj91x",  
  "minimumLinks": 0,  
  "connectionsBandwidth": "1Gbps",  
  "region": "us-east-1",  
  "location": "CSVA1"
```

```
}
```

Um eine Verbindung LAG unter Verwendung einer vorhandenen Verbindung zu erstellen

Im folgenden Beispiel wird eine LAG aus einer bestehenden Verbindung in Ihrem Konto erstellt und eine zweite neue Verbindung für die LAG mit derselben Bandbreite und demselben Standort wie die bestehende Verbindung angefordert.

Befehl:

```
aws directconnect create-lag --location EqDC2 --number-of-connections 2 --  
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dr
```

Ausgabe:

```
{  
  "awsDevice": "EqDC2-4h6ce2r1bes6",  
  "numberOfConnections": 2,  
  "lagState": "pending",  
  "ownerAccount": "123456789012",  
  "lagName": "2ConnLAG",  
  "connections": [  
    {  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-fh6ljcvo",  
      "lagId": "dxlag-fhccu14t",  
      "connectionState": "requested",  
      "bandwidth": "1Gbps",  
      "location": "EqDC2",  
      "connectionName": "Requested Connection 1 for Lag dxlag-fhccu14t",  
      "region": "us-east-1"  
    },  
    {  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-fgk145dr",  
      "lagId": "dxlag-fhccu14t",  
      "connectionState": "down",  
      "bandwidth": "1Gbps",  
      "location": "EqDC2",  
      "connectionName": "VAConn1",  
      "region": "us-east-1"  
    }  
  ],  
}
```

```

"lagId": "dxlag-fhccu14t",
"minimumLinks": 0,
"connectionsBandwidth": "1Gbps",
"region": "us-east-1",
"location": "EqDC2"
}

```

- API-Einheiten finden Sie [CreateLag](#) in der AWS CLI Befehlsreferenz.

create-private-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `create-private-virtual-interface`.

AWS CLI

Um eine private virtuelle Schnittstelle zu erstellen

Der folgende `create-private-virtual-interface` Befehl erstellt eine private virtuelle Schnittstelle:

```

aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrkx17 --
new-private-virtual-
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam
aba37db6

```

Ausgabe:

```

{
  "virtualInterfaceState": "pending",
  "asn": 65000,
  "vlan": 101,
  "customerAddress": "192.168.1.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrkx17",
  "virtualGatewayId": "vgw-aba37db6",
  "virtualInterfaceId": "dxvif-ffhkh74f",
  "authKey": "asdf34example",
  "routeFilterPrefixes": [],
  "location": "TIVIT",
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhkh74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n

```

```
<amazon_address>192.168.1.1/30</amazon_address>\n <bgp_asn>65000</bgp_asn>\n
<bgp_auth_key>asdf34example</bgp_auth_key>\n <amazon_bgp_asn>7224</amazon_bgp_asn>
\n <connection_type>private</connection_type>\n</logical_connection>\n",
  "amazonAddress": "192.168.1.1/30",
  "virtualInterfaceType": "private",
  "virtualInterfaceName": "PrivateVirtualInterface"
}
```

- API-Einheiten finden Sie [CreatePrivateVirtualInterface](#) in der AWS CLI Befehlsreferenz.

create-public-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `create-public-virtual-interface`.

AWS CLI

Um eine öffentliche virtuelle Schnittstelle zu erstellen

Der folgende `create-public-virtual-interface` Befehl erstellt eine öffentliche virtuelle Schnittstelle:

```
aws directconnect create-public-virtual-interface --connection-id dxcon-ffjrnx17 --
new-public-virtual-
interface virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exam
{cidr=203.0.113.4/30}
```

Ausgabe:

```
{
  "virtualInterfaceState": "verifying",
  "asn": 65000,
  "vlan": 2000,
  "customerAddress": "203.0.113.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrnx17",
  "virtualInterfaceId": "dxvif-fgh0hcrk",
  "authKey": "asdf34example",
  "routeFilterPrefixes": [
    {
      "cidr": "203.0.113.0/30"
    },
    {
```

```

        "cidr": "203.0.113.4/30"
    }
],
"location": "TIVIT",
"customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
  <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
"amazonAddress": "203.0.113.1/30",
"virtualInterfaceType": "public",
"virtualInterfaceName": "PublicVirtualInterface"
}

```

- API-Einheiten finden Sie [CreatePublicVirtualInterface](#) in der AWS CLI Befehlsreferenz.

create-transit-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `create-transit-virtual-interface`.

AWS CLI

Um eine virtuelle Transitschnittstelle zu erstellen

Im folgenden `create-transit-virtual-interface` Beispiel wird eine virtuelle Transitschnittstelle für die angegebene Verbindung erstellt.

```

aws directconnect create-transit-virtual-interface \
  --connection-id dxlag-fEXAMPLE \
  --new-transit-virtual-interface "virtualInterfaceName=Example Transit Virtual
Interface, vlan=126, asn=65110, mtu=1500, authKey=0xzxgA9YoW9h58u8SvEXAMPLE, amazonAddress=192.1
aada-5a1baEXAMPLE, tags=[{key=Tag, value=Example}]"

```

Ausgabe:

```

{
  "virtualInterface": {
    "ownerAccount": "1111222233333",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
  }
}

```



```

    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "Example Transit Virtual Interface",
    "vlan": 126,
    "asn": 65110,
    "amazonSideAsn": 4200000000,
    "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
    "amazonAddress": "192.168.1.1/30",
    "customerAddress": "192.168.1.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "pending",
    "customerRouterConfig": "<?xml version='1.0' encoding=
\\\"UTF-8\\\"?>\\n<logical_connection id='dxvif-fEXAMPLE'>\\n  <vlan>126</
vlan>\\n  <customer_address>192.168.1.2/30</customer_address>\\n
  <amazon_address>192.168.1.1/30</amazon_address>\\n  <bgp_asn>65110</
bgp_asn>\\n  <bgp_auth_key>0xzxcgA9YoW9h58u8Sv0mXRTw</bgp_auth_key>\\n
  <amazon_bgp_asn>4200000000</amazon_bgp_asn>\\n  <connection_type>transit</
connection_type>\\n</logical_connection>\\n\",
    "mtu": 1500,
    "jumboFrameCapable": true,
    "virtualGatewayId": "",
    "directConnectGatewayId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
    "routeFilterPrefixes": [],
    "bgpPeers": [
      {
        "bgpPeerId": "dxpeer-EXAMPLE",
        "asn": 65110,
        "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
        "addressFamily": "ipv4",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "bgpPeerState": "pending",
        "bgpStatus": "down",
        "awsDeviceV2": "loc1-26wz6vEXAMPLE"
      }
    ],
    "region": "sa-east-1",
    "awsDeviceV2": "loc1-26wz6vEXAMPLE",
    "tags": [
      {
        "key": "Tag",
        "value": "Example"
      }
    ]
  }
}

```

```
}
```

Weitere Informationen finden Sie unter [Erstellen einer virtuellen Transit-Schnittstelle zum Direct Connect Gateway](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateTransitVirtualInterface](#) in der AWS CLI Befehlsreferenz.

delete-bgp-peer

Das folgende Codebeispiel zeigt die Verwendung `delete-bgp-peer`.

AWS CLI

Um einen BGP Peer von einer virtuellen Schnittstelle zu löschen

Im folgenden Beispiel wird der IPv6 BGP Peer von der virtuellen Schnittstelle `dxvif-fg1vuj3d` gelöscht.

Befehl:

```
aws directconnect delete-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --asn 64600
--customer-address 2001:db8:1100:2f0:0:1:9cb4:4216/125
```

Ausgabe:

```
{
  "virtualInterface": {
    "virtualInterfaceState": "available",
    "asn": 65000,
    "vlan": 125,
    "customerAddress": "169.254.255.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fguhmq1c",
    "addressFamily": "ipv4",
    "virtualGatewayId": "vgw-f9eb0c90",
    "virtualInterfaceId": "dxvif-fg1vuj3d",
    "authKey": "0xC_ukbCerl6EYA0example",
    "routeFilterPrefixes": [],
    "location": "EqDC2",
    "bgpPeers": [
      {
        "bgpStatus": "down",
        "customerAddress": "169.254.255.2/30",
```

```

        "addressFamily": "ipv4",
        "authKey": "0xC_ukbCerl6EYA0uexample",
        "bgpPeerState": "available",
        "amazonAddress": "169.254.255.1/30",
        "asn": 65000
    },
    {
        "bgpStatus": "down",
        "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
        "addressFamily": "ipv6",
        "authKey": "0xS27kAIU_VHPjjAexample",
        "bgpPeerState": "deleting",
        "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
        "asn": 64600
    }
],
"customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
  <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0xC_ukbCerl6EYA0example</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "Test"
  }
}

```

- API-Einheiten finden Sie [DeleteBgpPeer](#) in der AWS CLI Befehlsreferenz.

delete-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-connection`.

AWS CLI

Um eine Verbindung zu löschen

Der folgende `delete-connection` Befehl löscht die angegebene Verbindung:

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

Ausgabe:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "deleted",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "connectionName": "Connection to AWS",
  "region": "sa-east-1"
}
```

- API-Einheiten finden Sie [DeleteConnection](#) in der AWS CLI Befehlsreferenz.

delete-direct-connect-gateway-association

Das folgende Codebeispiel zeigt die Verwendung `delete-direct-connect-gateway-association`.

AWS CLI

So löschen Sie eine Direct Connect-Gateway-Zuordnung

Im folgenden `delete-direct-connect-gateway-association` Beispiel wird die Direct Connect-Gateway-Zuordnung zu einem Transit-Gateway gelöscht, das die angegebene Zuordnungs-ID hat.

```
aws directconnect delete-direct-connect-gateway-association --association-id
be85116d-46eb-4b43-a27a-da0c2ad648de
```

Ausgabe:

```
{
  "directConnectGatewayAssociation": {
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "123456789012",
    "associationState": "disassociating",
    "associatedGateway": {
      "id": "tgw-095b3b0b54EXAMPLE",
      "type": "transitGateway",
      "ownerAccount": "123456789012",
      "region": "us-east-1"
    }
  }
}
```

```

    },
    "associationId": " be85116d-46eb-4b43-a27a-da0c2ad648deEXAMPLE ",
    "allowedPrefixesToDirectConnectGateway": [
      {
        "cidr": "192.0.1.0/28"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Zuordnen und Trennen von Transit-Gateways](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteDirectConnectGatewayAssociation](#) Befehlsreferenz.AWS CLI

delete-direct-connect-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-direct-connect-gateway`.

AWS CLI

Um ein Direct Connect-Gateway zu löschen

Im folgenden Beispiel wird das Direct Connect-Gateway `5f294f92-bafb-4011-916d-9b0bexample` gelöscht.

Befehl:

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Ausgabe:

```

{
  "directConnectGateway": {
    "amazonSideAsn": 64512,
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
    "ownerAccount": "123456789012",
    "directConnectGatewayName": "DxGateway1",
    "directConnectGatewayState": "deleting"
  }
}

```

```
}
```

- API-Einheiten finden Sie [DeleteDirectConnectGateway](#) in der AWS CLI Befehlsreferenz.

delete-interconnect

Das folgende Codebeispiel zeigt die Verwendung `delete-interconnect`.

AWS CLI

Um eine Verbindung zu löschen

Der folgende `delete-interconnect` Befehl löscht die angegebene Verbindung:

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

Ausgabe:

```
{
  "interconnectState": "deleted"
}
```

- API-Einheiten finden Sie [DeleteInterconnect](#) in der AWS CLI Befehlsreferenz.

delete-lag

Das folgende Codebeispiel zeigt die Verwendung `delete-lag`.

AWS CLI

Um ein zu löschen LAG

Das folgende Beispiel löscht das angegebene LAG.

Befehl:

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

Ausgabe:

```
{
  "awsDevice": "EqDC2-4h6ce2r1bes6",
}
```

```
"numberOfConnections": 0,  
"lagState": "deleted",  
"ownerAccount": "123456789012",  
"lagName": "TestLAG",  
"connections": [],  
"lagId": "dxlag-ffrhowd9",  
"minimumLinks": 0,  
"connectionsBandwidth": "1Gbps",  
"region": "us-east-1",  
"location": "EqDC2"  
}
```

- API-Einheiten finden Sie [DeleteLag](#) in der AWS CLI Befehlsreferenz.

delete-virtual-interface

Das folgende Codebeispiel zeigt die Verwendung `delete-virtual-interface`.

AWS CLI

Um eine virtuelle Schnittstelle zu löschen

Der folgende `delete-virtual-interface` Befehl löscht die angegebene virtuelle Schnittstelle:

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhkh74f
```

Ausgabe:

```
{  
  "virtualInterfaceState": "deleting"  
}
```

- API-Einheiten finden Sie [DeleteVirtualInterface](#) in der AWS CLI Befehlsreferenz.

describe-connection-loa

Das folgende Codebeispiel zeigt die Verwendung `describe-connection-loa`.

AWS CLI

Zur Beschreibung Ihres LOA — CFA für eine Verbindung mit Linux oder Mac OS X

Das folgende Beispiel beschreibt Ihre LOA - CFA für Verbindung dxcon-fh6ayh1d. Der Inhalt von LOA - CFA ist Base64-kodiert. Dieser Befehl verwendet die `--query` Parameter `--output` und, um die Ausgabe zu steuern und den Inhalt der Struktur zu extrahieren. `loaContent` Der letzte Teil des Befehls dekodiert den Inhalt mithilfe des `base64` Dienstprogramms und sendet die Ausgabe in eine PDF Datei.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --output text --query loa.loaContent/base64 --decode > myLoaCfa.pdf
```

Um Ihre LOA - CFA für eine Verbindung unter Windows zu beschreiben

Das vorherige Beispiel erfordert die Verwendung des `base64` Dienstprogramms, um die Ausgabe zu dekodieren. Auf einem Windows-Computer können Sie `certutil` stattdessen verwenden. Im folgenden Beispiel beschreibt der erste Befehl Ihre LOA CFA -for-Verbindung dxcon-fh6ayh1d und verwendet die `--query` Parameter `--output` und, um die Ausgabe zu steuern und den Inhalt der `loaContent` Struktur in eine Datei mit dem Namen zu extrahieren `myLoaCfa.base64`. Der zweite Befehl verwendet das `certutil` Hilfsprogramm, um die Datei zu dekodieren und die Ausgabe in eine PDF Datei zu senden.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Weitere Informationen zur Steuerung der AWS CLI Ausgabe finden Sie unter [Steuern der Befehlsausgabe über die AWS Befehlszeilenschnittstelle](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DescribeConnectionLoa](#) in der AWS CLI Befehlsreferenz.

describe-connections-on-interconnect

Das folgende Codebeispiel zeigt die Verwendung `describe-connections-on-interconnect`.

AWS CLI

Um Verbindungen auf einer Interconnect-Verbindung aufzulisten

Der folgende `describe-connections-on-interconnect` Befehl listet Verbindungen auf, die auf der angegebenen Interconnect bereitgestellt wurden:


```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-  
fgktov66
```

Ausgabe:

```
{  
  "connections": [  
    {  
      "partnerName": "TIVIT",  
      "vlan": 101,  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-ffzc51m1",  
      "connectionState": "ordering",  
      "bandwidth": "500Mbps",  
      "location": "TIVIT",  
      "connectionName": "mydcinterconnect",  
      "region": "sa-east-1"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribeConnectionsOnInterconnect](#) in der AWS CLI Befehlsreferenz.

describe-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-connections`.

AWS CLI

Um alle Verbindungen in der aktuellen Region aufzulisten

Der folgende `describe-connections` Befehl listet alle Verbindungen in der aktuellen Region auf:

```
aws directconnect describe-connections
```

Ausgabe:

```
{  
  "connections": [  
    {
```

```
{
  "awsDevice": "EqDC2-123h49s71dabc",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fguhmq1c",
  "lagId": "dxlag-ffrz71kw",
  "connectionState": "down",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "My_Connection",
  "loaIssueTime": 1491568964.0,
  "region": "us-east-1"
}
]
```

- API-Einheiten finden Sie [DescribeConnections](#) in der AWS CLI Befehlsreferenz.

describe-direct-connect-gateway-association-proposals

Das folgende Codebeispiel zeigt die Verwendung `describe-direct-connect-gateway-association-proposals`.

AWS CLI

Um Ihre Vorschläge für die Direct Connect-Gateway-Zuordnung zu beschreiben

Im folgenden `describe-direct-connect-gateway-association-proposals` Beispiel werden Details zu Ihren Vorschlägen für die Direct Connect-Gateway-Zuordnung angezeigt.

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

Ausgabe:

```
{
  "directConnectGatewayAssociationProposals": [
    {
      "proposalId": "c2ede9b4-bbc6-4d33-923c-bc4feEXAMPLE",
      "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
      "directConnectGatewayOwnerAccount": "111122223333",
      "proposalState": "requested",
      "associatedGateway": {
        "id": "tgw-02f776b1a7EXAMPLE",
```

```
        "type": "transitGateway",
        "ownerAccount": "111122223333",
        "region": "us-east-1"
    },
    "existingAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.2.0/30"
        },
        {
            "cidr": "192.168.1.0/30"
        }
    ],
    "requestedAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.1.0/30"
        }
    ]
},
{
    "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
    "directConnectGatewayId": "11560968-4ac1-4fd3-bcb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "111122223333",
    "proposalState": "accepted",
    "associatedGateway": {
        "id": "tgw-045776b1a7EXAMPLE",
        "type": "transitGateway",
        "ownerAccount": "111122223333",
        "region": "us-east-1"
    },
    "existingAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.4.0/30"
        },
        {
            "cidr": "192.168.5.0/30"
        }
    ],
    "requestedAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.5.0/30"
        }
    ]
}
]
```

```
}
```

Weitere Informationen finden Sie unter [Zuordnen und Trennen von Transit-Gateways](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeDirectConnectGatewayAssociationProposals](#) Befehlsreferenz.AWS CLI

describe-direct-connect-gateway-associations

Das folgende Codebeispiel zeigt die Verwendung `describe-direct-connect-gateway-associations`.

AWS CLI

Um Direct Connect-Gateway-Zuordnungen zu beschreiben

Das folgende Beispiel beschreibt alle Verknüpfungen mit dem Direct Connect-Gateway `5f294f92-bafb-4011-916d-9b0bexample`.

Befehl:

```
aws directconnect describe-direct-connect-gateway-associations --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Ausgabe:

```
{
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoiOU83OTFodzdycnZCbkn4MExHeHVwQT09IiwieYyI6InIxTEN0UEVHV0I1UF1kaWFmNjI",
  "directConnectGatewayAssociations": [
    {
      "associationState": "associating",
      "virtualGatewayOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualGatewayId": "vgw-6efe725e",
      "virtualGatewayRegion": "us-east-2"
    },
    {
      "associationState": "disassociating",
      "virtualGatewayOwnerAccount": "123456789012",
```

```

    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
    "virtualGatewayId": "vgw-ebaa27db",
    "virtualGatewayRegion": "us-east-2"
  }
]
}

```

- API-Einheiten finden Sie [DescribeDirectConnectGatewayAssociations](#) in der AWS CLI Befehlsreferenz.

describe-direct-connect-gateway-attachments

Das folgende Codebeispiel zeigt die Verwendung `describe-direct-connect-gateway-attachments`.

AWS CLI

Um Direct Connect-Gateway-Anlagen zu beschreiben

Das folgende Beispiel beschreibt die virtuellen Schnittstellen, die an das Direct Connect-Gateway angeschlossen sind `5f294f92-bafb-4011-916d-9b0bexample`.

Befehl:

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Ausgabe:

```

{
  "directConnectGatewayAttachments": [
    {
      "virtualInterfaceOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualInterfaceRegion": "us-east-2",
      "attachmentState": "attaching",
      "virtualInterfaceId": "dxvif-fg9zyabc"
    }
  ],
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoibEhXd1NpUXF5RzhoL1JyUW52S1V2QT09IiwieyI6Im5wQjFHQ0RyQUdRS3puNnNXcU

```

```
}
```

- API-Einheiten finden Sie [DescribeDirectConnectGatewayAttachments](#) in der AWS CLI Befehlsreferenz.

describe-direct-connect-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-direct-connect-gateways`.

AWS CLI

Um Ihre Direct Connect-Gateways zu beschreiben

Das folgende Beispiel beschreibt alle Ihre Direct Connect-Gateways.

Befehl:

```
aws directconnect describe-direct-connect-gateways
```

Ausgabe:

```
{
  "directConnectGateways": [
    {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "cf68415c-f4ae-48f2-87a7-3b52cexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway2",
      "directConnectGatewayState": "available"
    },
    {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "available"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeDirectConnectGateways](#) in der AWS CLI Befehlsreferenz.

describe-hosted-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-hosted-connections`.

AWS CLI

Um Verbindungen auf einer Interconnect-Verbindung aufzulisten

Das folgende Beispiel listet Verbindungen auf, die auf der angegebenen Verbindungsleitung bereitgestellt wurden.

Befehl:

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

Ausgabe:

```
{
  "connections": [
    {
      "partnerName": "TIVIT",
      "vlan": 101,
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffzc51m1",
      "connectionState": "ordering",
      "bandwidth": "500Mbps",
      "location": "TIVIT",
      "connectionName": "mydcinterconnect",
      "region": "sa-east-1"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeHostedConnections](#) in der AWS CLI Befehlsreferenz.

describe-interconnect-loa

Das folgende Codebeispiel zeigt die Verwendung `describe-interconnect-loa`.

AWS CLI

Zur Beschreibung Ihres LOA — CFA für eine Verbindung unter Linux oder Mac OS X

Das folgende Beispiel beschreibt Ihre LOA — CFA für `dxcon-fh6ayh1d` Interconnect. Der Inhalt von LOA - ist CFA Base64-kodiert. Dieser Befehl verwendet die `--query` Parameter `--output` und, um die Ausgabe zu steuern und den Inhalt der Struktur zu extrahieren. `loaContent` Der letzte Teil des Befehls dekodiert den Inhalt mithilfe des `base64` Dienstprogramms und sendet die Ausgabe in eine PDF Datei.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --output text --query Loa.LoaContent/base64 --decode > myLoaCfa.pdf
```

Um Ihren LOA - CFA für eine Verbindung unter Windows zu beschreiben

Das vorherige Beispiel erfordert die Verwendung des `base64` Dienstprogramms, um die Ausgabe zu dekodieren. Auf einem Windows-Computer können Sie `certutil` stattdessen verwenden. Im folgenden Beispiel beschreibt der erste Befehl Ihren LOA - CFA for interconnect `dxcon-fh6ayh1d` und verwendet die `--query` Parameter `--output` und, um die Ausgabe zu steuern und den Inhalt der `loaContent` Struktur in eine Datei mit dem Namen `myLoaCfa.base64` zu extrahieren. Der zweite Befehl verwendet das `certutil` Hilfsprogramm, um die Datei zu dekodieren und die Ausgabe in eine PDF Datei zu senden.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --output text --query Loa.LoaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Weitere Informationen zur Steuerung der AWS CLI Ausgabe finden Sie unter [Steuern der Befehlsausgabe über die AWS Befehlszeilenschnittstelle](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DescribeInterconnectLoa](#) in der AWS CLI Befehlsreferenz.

describe-interconnects

Das folgende Codebeispiel zeigt die Verwendung `describe-interconnects`.

AWS CLI

Um Verbindungen aufzulisten

Der folgende `describe-interconnects` Befehl listet die Interconnects auf, die Ihrem AWS Konto gehören:


```
aws directconnect describe-interconnects
```

Ausgabe:

```
{
  "interconnects": [
    {
      "region": "sa-east-1",
      "bandwidth": "1Gbps",
      "location": "TIVIT",
      "interconnectName": "1G Interconnect to AWS",
      "interconnectId": "dxcon-fgktov66",
      "interconnectState": "down"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeInterconnects](#) in der AWS CLI Befehlsreferenz.

describe-lags

Das folgende Codebeispiel zeigt die Verwendung `describe-lags`.

AWS CLI

Um deine zu beschreiben LAGs

Der folgende Befehl beschreibt alle Ihre LAGs für die aktuelle Region.

Befehl:

```
aws directconnect describe-lags
```

Ausgabe:

```
{
  "lags": [
    {
      "awsDevice": "EqDC2-19y7z3m17xpuz",
      "numberOfConnections": 2,

```

```

    "lagState": "down",
    "ownerAccount": "123456789012",
    "lagName": "DA-LAG",
    "connections": [
      {
        "ownerAccount": "123456789012",
        "connectionId": "dxcon-ffnikghc",
        "lagId": "dxlag-fgsu9erb",
        "connectionState": "requested",
        "bandwidth": "10Gbps",
        "location": "EqDC2",
        "connectionName": "Requested Connection 1 for Lag dxlag-fgsu9erb",
        "region": "us-east-1"
      },
      {
        "ownerAccount": "123456789012",
        "connectionId": "dxcon-fglgbdea",
        "lagId": "dxlag-fgsu9erb",
        "connectionState": "requested",
        "bandwidth": "10Gbps",
        "location": "EqDC2",
        "connectionName": "Requested Connection 2 for Lag dxlag-fgsu9erb",
        "region": "us-east-1"
      }
    ],
    "lagId": "dxlag-fgsu9erb",
    "minimumLinks": 0,
    "connectionsBandwidth": "10Gbps",
    "region": "us-east-1",
    "location": "EqDC2"
  }
]
}

```

- API-Einheiten finden Sie [DescribeLags](#) in der AWS CLI Befehlsreferenz.

describe-loa

Das folgende Codebeispiel zeigt die Verwendung `describe-loa`.

AWS CLI

Zur Beschreibung Ihres LOA — CFA für eine Verbindung mit Linux oder Mac OS X

Das folgende Beispiel beschreibt Ihre LOA - CFA für Verbindung `dxcon-fh6ayh1d`. Der Inhalt von LOA - CFA ist Base64-kodiert. Dieser Befehl verwendet die `--query` Parameter `--output` und, um die Ausgabe zu steuern und den Inhalt der Struktur zu extrahieren. `loaContent` Der letzte Teil des Befehls dekodiert den Inhalt mithilfe des `base64` Dienstprogramms und sendet die Ausgabe in eine PDF Datei.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query Loa.LoaContent/base64 --decode > myLoaCfa.pdf
```

Um Ihre LOA - CFA für eine Verbindung unter Windows zu beschreiben

Das vorherige Beispiel erfordert die Verwendung des `base64` Dienstprogramms, um die Ausgabe zu dekodieren. Auf einem Windows-Computer können Sie `certutil` stattdessen verwenden. Im folgenden Beispiel beschreibt der erste Befehl Ihre LOA CFA -for-Verbindung `dxcon-fh6ayh1d` und verwendet die `--query` Parameter `--output` und, um die Ausgabe zu steuern und den Inhalt der `loaContent` Struktur in eine Datei mit dem Namen zu extrahieren `myLoaCfa.base64`. Der zweite Befehl verwendet das `certutil` Hilfsprogramm, um die Datei zu dekodieren und die Ausgabe in eine PDF Datei zu senden.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query Loa.LoaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Weitere Informationen zur Steuerung der AWS CLI Ausgabe finden Sie unter [Steuern der Befehlsausgabe über die AWS Befehlszeilenschnittstelle](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DescribeLoa](#) in der AWS CLI Befehlsreferenz.

describe-locations

Das folgende Codebeispiel zeigt die Verwendung `describe-locations`.

AWS CLI

Um AWS Direct Connect-Partner und Standorte aufzulisten

Der folgende `describe-locations` Befehl listet AWS Direct Connect-Partner und -Standorte in der aktuellen Region auf:

```
aws directconnect describe-locations
```

Ausgabe:

```
{
  "locations": [
    {
      "locationName": "NAP do Brasil, Barueri, Sao Paulo",
      "locationCode": "TNDB"
    },
    {
      "locationName": "Tivit - Site Transamerica (Sao Paulo)",
      "locationCode": "TIVIT"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeLocations](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Um Tags für Ihre AWS Direct Connect-Ressourcen zu beschreiben

Der folgende Befehl beschreibt die Tags für die Verbindung `dxcon-abcabc12`.

Befehl:

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12
```

Ausgabe:

```
{
  "resourceTags": [
    {
      "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12",
```

```
    "tags": [
      {
        "value": "VAConnection",
        "key": "Name"
      }
    ]
  }
]
```

- API-Einheiten finden Sie [DescribeTags](#) in der AWS CLI Befehlsreferenz.

describe-virtual-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-virtual-gateways`.

AWS CLI

Um virtuelle private Gateways aufzulisten

Der folgende `describe-virtual-gateways` Befehl listet virtuelle private Gateways auf, die Ihrem AWS Konto gehören:

```
aws directconnect describe-virtual-gateways
```

Ausgabe:

```
{
  "virtualGateways": [
    {
      "virtualGatewayId": "vgw-aba37db6",
      "virtualGatewayState": "available"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeVirtualGateways](#) in der AWS CLI Befehlsreferenz.

describe-virtual-interfaces

Das folgende Codebeispiel zeigt die Verwendung `describe-virtual-interfaces`.

AWS CLI

Um alle virtuellen Schnittstellen aufzulisten

Der folgende `describe-virtual-interfaces` Befehl listet die Informationen zu allen virtuellen Schnittstellen auf, die Ihrem AWS Konto zugeordnet sind:

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrkx17
```

Ausgabe:

```
{
  "virtualInterfaces": [
    {
      "virtualInterfaceState": "down",
      "asn": 65000,
      "vlan": 101,
      "customerAddress": "192.168.1.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffjrkx17",
      "virtualGatewayId": "vgw-aba37db6",
      "virtualInterfaceId": "dxvif-ffhkh74f",
      "authKey": "asdf34example",
      "routeFilterPrefixes": [],
      "location": "TIVIT",
      "customerRouterConfig": "<?xml version='1.0' encoding='
UTF-8'><logical_connection id='dxvif-ffhkh74f'>
  <vlan>101</vlan>
  <customer_address>192.168.1.2/30</customer_address>
  <amazon_address>192.168.1.1/30</amazon_address>
  <bgp_asn>65000</bgp_asn>
  <bgp_auth_key>asdf34example</bgp_auth_key>
  <amazon_bgp_asn>7224</amazon_bgp_asn>
  <connection_type>private</connection_type>
</logical_connection>
",
      "amazonAddress": "192.168.1.1/30",
      "virtualInterfaceType": "private",
      "virtualInterfaceName": "PrivateVirtualInterface"
    },
    {
      "virtualInterfaceState": "verifying",
      "asn": 65000,
      "vlan": 2000,
      "customerAddress": "203.0.113.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffjrkx17",
      "virtualGatewayId": "",
    }
  ]
}
```

```

    "virtualInterfaceId": "dxvif-fgh0hcrk",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
      {
        "cidr": "203.0.113.4/30"
      },
      {
        "cidr": "203.0.113.0/30"
      }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
UTF-8\"?>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
  <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
  }
]
}

```

- API-Einheiten finden Sie [DescribeVirtualInterfaces](#) in der AWS CLI Befehlsreferenz.

disassociate-connection-from-lag

Das folgende Codebeispiel zeigt die Verwendung `disassociate-connection-from-lag`.

AWS CLI

Um eine Verbindung von einem zu trennen LAG

Im folgenden Beispiel wird die Verbindung zwischen der angegebenen Verbindung und der angegebenen Verbindung getrennt. LAG

Befehl:

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --
connection-id dxcon-fg9607vm
```

Ausgabe:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
}
```

- API-Einheiten finden Sie unter [DisassociateConnectionFromLag AWS CLI](#) Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einer AWS Direct Connect-Ressource ein Tag hinzu

Mit dem folgenden Befehl wird der Verbindung ein Tag mit dem Schlüssel `Name` und `VAConnection` dem Wert von `dxcon-abcabc12` hinzugefügt. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

So entfernen Sie ein Tag aus einer AWS Direct Connect-Ressource

Mit dem folgenden Befehl wird das Tag mit dem Schlüssel Name aus der Verbindung entfernt `dxcon-abcabc12`. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-direct-connect-gateway-association

Das folgende Codebeispiel zeigt die Verwendung `update-direct-connect-gateway-association`.

AWS CLI

Um die angegebenen Attribute der Direct Connect-Gateway-Zuordnung zu aktualisieren

Im folgenden `update-direct-connect-gateway-association` Beispiel wird der angegebene CIDR Block zu einer Direct Connect-Gateway-Zuordnung hinzugefügt.

```
aws directconnect update-direct-connect-gateway-association \  
  --association-id 820a6e4f-5374-4004-8317-3f64bEXAMPLE \  
  --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.2.0/30
```

Ausgabe:

```
{  
  "directConnectGatewayAssociation": {  
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",  
    "directConnectGatewayOwnerAccount": "111122223333",  
    "associationState": "updating",  
    "associatedGateway": {  
      "id": "tgw-02f776b1a7EXAMPLE",  
      "type": "transitGateway",  
      "ownerAccount": "111122223333",  
      "region": "us-east-1"  
    },  
    "associationId": "820a6e4f-5374-4004-8317-3f64bEXAMPLE",  
  },  
}
```

```
    "allowedPrefixesToDirectConnectGateway": [  
      {  
        "cidr": "192.168.2.0/30"  
      },  
      {  
        "cidr": "192.168.1.0/30"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Direct Connect Gateways](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDirectConnectGatewayAssociation](#) in der AWS CLI Befehlsreferenz.

update-lag

Das folgende Codebeispiel zeigt die Verwendung `update-lag`.

AWS CLI

Um ein zu aktualisieren LAG

Im folgenden Beispiel wird der Name des angegebenen Objekts geändert LAG.

Befehl:

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

Ausgabe:

```
{  
  "awsDevice": "CSVA1-23u8tlpaz8iks",  
  "numberOfConnections": 2,  
  "lagState": "down",  
  "ownerAccount": "123456789012",  
  "lagName": "2ConnLag",  
  "connections": [  
    {  
      "ownerAccount": "123456789012",
```

```

    "connectionId": "dxcon-fflqyj95",
    "lagId": "dxlag-ffjhj9lx",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "CSVA1",
    "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
    "region": "us-east-1"
  },
  {
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffqr6x5q",
    "lagId": "dxlag-ffjhj9lx",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "CSVA1",
    "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
    "region": "us-east-1"
  }
],
"lagId": "dxlag-ffjhj9lx",
"minimumLinks": 0,
"connectionsBandwidth": "1Gbps",
"region": "us-east-1",
"location": "CSVA1"
}

```

- API-Einheiten finden Sie [UpdateLag](#) unter AWS CLI Befehlsreferenz.

update-virtual-interface-attributes

Das folgende Codebeispiel zeigt die Verwendung `update-virtual-interface-attributes`.

AWS CLI

Um die MTU einer virtuellen Schnittstelle zu aktualisieren

Im folgenden `update-virtual-interface-attributes` Beispiel wird die MTU der angegebenen virtuellen Schnittstelle aktualisiert.

```

aws directconnect update-virtual-interface-attributes \
  --virtual-interface-id dxvif-fEXAMPLE \
  --mtu 1500

```

Ausgabe:

```

{
  "ownerAccount": "1111222233333",
  "virtualInterfaceId": "dxvif-fEXAMPLE",
  "location": "loc1",
  "connectionId": "dxlag-fEXAMPLE",
  "virtualInterfaceType": "transit",
  "virtualInterfaceName": "example transit virtual interface",
  "vlan": 125,
  "asn": 650001,
  "amazonSideAsn": 64512,
  "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
  "amazonAddress": "169.254.248.1/30",
  "customerAddress": "169.254.248.2/30",
  "addressFamily": "ipv4",
  "virtualInterfaceState": "down",
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>125</vlan>
\n  <customer_address>169.254.248.2/30</customer_address>\n
  <amazon_address>169.254.248.1/30</amazon_address>\n  <bgp_asn>650001</bgp_asn>\n
  <bgp_auth_key>0xzxgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>64512</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
  "mtu": 1500,
  "jumboFrameCapable": true,
  "virtualGatewayId": "",
  "directConnectGatewayId": "879b76a1-403d-4700-8b53-4a56ed85436e",
  "routeFilterPrefixes": [],
  "bgpPeers": [
    {
      "bgpPeerId": "dxpeer-fEXAMPLE",
      "asn": 650001,
      "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
      "addressFamily": "ipv4",
      "amazonAddress": "169.254.248.1/30",
      "customerAddress": "169.254.248.2/30",
      "bgpPeerState": "available",
      "bgpStatus": "down",
      "awsDeviceV2": "loc1-26wz6vEXAMPLE"
    }
  ],
  "region": "sa-east-1",
  "awsDeviceV2": "loc1-26wz6vEXAMPLE",

```

```
"tags": []  
}
```

Weitere Informationen finden Sie unter [Setting Network MTU for Private Virtual Interfaces oder Transit Virtual Interfaces](#) im AWS Direct Connect-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateVirtualInterfaceAttributes](#) in der AWS CLI Befehlsreferenz.

AWS Directory Service Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Directory Service.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-directories

Das folgende Codebeispiel zeigt die Verwendung `describe-directories`.

AWS CLI

Um Details zu Ihren Verzeichnissen zu erhalten

Im folgenden `describe-directories` Beispiel werden Details zum angegebenen Verzeichnis angezeigt.

```
aws ds describe-directories \  
  --directory-id d-a1b2c3d4e5
```

Ausgabe:

```
{
  "DirectoryDescriptions": [
    {
      "DirectoryId": "d-a1b2c3d4e5",
      "Name": "mydirectory.example.com",
      "ShortName": "mydirectory",
      "Size": "Small",
      "Edition": "Standard",
      "Alias": "d-a1b2c3d4e5",
      "AccessUrl": "d-a1b2c3d4e5.awsapps.com",
      "Stage": "Active",
      "ShareStatus": "Shared",
      "ShareMethod": "HANDSHAKE",
      "ShareNotes": "These are my share notes",
      "LaunchTime": "2019-07-08T15:33:46.327000-07:00",
      "StageLastUpdatedDateTime": "2019-07-08T15:59:12.307000-07:00",
      "Type": "SharedMicrosoftAD",
      "SsoEnabled": false,
      "DesiredNumberOfDomainControllers": 0,
      "OwnerDirectoryDescription": {
        "DirectoryId": "d-b2c3d4e5f6",
        "AccountId": "123456789111",
        "DnsIpAddr": [
          "203.113.0.248",
          "203.113.0.253"
        ],
        "VpcSettings": {
          "VpcId": "vpc-a1b2c3d4",
          "SubnetIds": [
            "subnet-a1b2c3d4",
            "subnet-d4c3b2a1"
          ],
          "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c"
          ]
        }
      }
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeDirectories](#) unter AWS CLI Befehlsreferenz.

describe-trusts

Das folgende Codebeispiel zeigt die Verwendung `describe-trusts`.

AWS CLI

Um Einzelheiten zu Ihren Vertrauensbeziehungen zu erhalten

Im folgenden `describe-trusts` Beispiel werden Details zu den Vertrauensstellungen für das angegebene Verzeichnis angezeigt.

```
aws ds describe-trusts \  
  --directory-id d-a1b2c3d4e5
```

Ausgabe:

```
{  
  "Trusts": [  
    {  
      "DirectoryId": "d-a1b2c3d4e5",  
      "TrustId": "t-9a8b7c6d5e",  
      "RemoteDomainName": "other.example.com",  
      "TrustType": "Forest",  
      "TrustDirection": "Two-Way",  
      "TrustState": "Verified",  
      "CreatedDateTime": "2017-06-20T18:08:45.614000-07:00",  
      "LastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",  
      "StateLastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",  
      "SelectiveAuth": "Disabled"  
    }  
  ]  
}
```

- API Einzelheiten finden Sie [DescribeTrusts](#) unter AWS CLI Befehlsreferenz.

AWS DMS Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS DMS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-resource

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-resource`.

AWS CLI

Um einer Ressource Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel werden einer Replikationsinstanz Tags hinzugefügt.

```
aws dms add-tags-to-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  \  
  --tags Key=Environment,Value=PROD Key=Project,Value=dbMigration
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Resources](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [AddTagsToResource](#) in der AWS CLI Befehlsreferenz.

create-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-endpoint`.

AWS CLI

Um einen Endpunkt zu erstellen

Das folgende create-endpoint Beispiel erstellt einen Endpunkt für eine Amazon S3 S3-Quelle.

```
aws dms create-endpoint \
  --endpoint-type source \
  --engine-name s3 \
  --endpoint-identifier src-endpoint \
  --s3-settings file://s3-settings.json
```

Inhalt von s3-settings.json:

```
{
  "BucketName": "my-corp-data",
  "BucketFolder": "sourcedata",
  "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role"
}
```

Ausgabe:

```
{
  "Endpoint": {
    "EndpointIdentifier": "src-endpoint",
    "EndpointType": "SOURCE",
    "EngineName": "s3",
    "EngineDisplayName": "Amazon S3",
    "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",
    "Status": "active",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U",
    "SslMode": "none",
    "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
    "S3Settings": {
      "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
      "CsvRowDelimiter": "\\n",
      "CsvDelimiter": ",",
      "BucketFolder": "sourcedata",
      "BucketName": "my-corp-data",
      "CompressionType": "NONE",
      "EnableStatistics": true
    }
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Endpunkten](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateEndpoint AWS CLI](#) Befehlsreferenz.

create-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-event-subscription`.

AWS CLI

Um Veranstaltungsabonnements aufzulisten

Im folgenden `create-event-subscription` Beispiel wird ein Event-Abonnement für ein SNS Amazon-Thema (`my-sns-topic`) erstellt.

```
aws dms create-event-subscription \  
  --subscription-name my-dms-events \  
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:my-sns-topic
```

Ausgabe:

```
{  
  "EventSubscription": {  
    "CustomerAwsId": "123456789012",  
    "CustSubscriptionId": "my-dms-events",  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",  
    "Status": "creating",  
    "SubscriptionCreationTime": "2020-05-21 21:58:38.598",  
    "Enabled": true  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Ereignissen und Benachrichtigungen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateEventSubscription](#) unter AWS CLI Befehlsreferenz.

create-replication-instance

Das folgende Codebeispiel zeigt die Verwendung `create-replication-instance`.

AWS CLI

Um eine Replikationsinstanz zu erstellen

Im folgenden `create-replication-instance` Beispiel wird eine Replikationsinstanz erstellt.

```
aws dms create-replication-instance \  
  --replication-instance-identifier my-repl-instance \  
  --replication-instance-class dms.t2.micro \  
  --allocated-storage 5
```

Ausgabe:

```
{  
  "ReplicationInstance": {  
    "ReplicationInstanceIdentifier": "my-repl-instance",  
    "ReplicationInstanceClass": "dms.t2.micro",  
    "ReplicationInstanceStatus": "creating",  
    "AllocatedStorage": 5,  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-f839b688",  
        "Status": "active"  
      }  
    ],  
    "ReplicationSubnetGroup": {  
      "ReplicationSubnetGroupIdentifier": "default",  
      "ReplicationSubnetGroupDescription": "default",  
      "VpcId": "vpc-136a4c6a",  
      "SubnetGroupStatus": "Complete",  
      "Subnets": [  
        {  
          "SubnetIdentifier": "subnet-da327bf6",  
          "SubnetAvailabilityZone": {  
            "Name": "us-east-1a"  
          },  
          "SubnetStatus": "Active"  
        },  
        {  
          "SubnetIdentifier": "subnet-42599426",
```

```
        "SubnetAvailabilityZone": {
            "Name": "us-east-1d"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-bac383e0",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1c"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-6746046b",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-d7c825e8",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-cbfff283",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
        },
        "SubnetStatus": "Active"
    }
]
},
"PreferredMaintenanceWindow": "sat:12:35-sat:13:05",
"PendingModifiedValues": {},
"MultiAZ": false,
"EngineVersion": "3.3.2",
"AutoMinorVersionUpgrade": true,
"KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
"ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:ZK2VQBUWFDBAWHIXHAYG5G2PKY",
```

```

    "PubliclyAccessible": true
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit einer AWS DMS Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateReplicationInstance](#) unter AWS CLI Befehlsreferenz.

create-replication-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `create-replication-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu erstellen

Im folgenden `create-replication-subnet-group` Beispiel wird eine Gruppe erstellt, die aus 3 Subnetzen besteht.

```

aws dms create-replication-subnet-group \
  --replication-subnet-group-identifier my-subnet-group \
  --replication-subnet-group-description "my subnet group" \
  --subnet-ids subnet-da327bf6 subnet-bac383e0 subnet-d7c825e8

```

Ausgabe:

```

{
  "ReplicationSubnetGroup": {
    "ReplicationSubnetGroupIdentifier": "my-subnet-group",
    "ReplicationSubnetGroupDescription": "my subnet group",
    "VpcId": "vpc-136a4c6a",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-da327bf6",
        "SubnetAvailabilityZone": {
          "Name": "us-east-1a"
        },
        "SubnetStatus": "Active"
      },
    ],
  },
}

```

```

    {
      "SubnetIdentifier": "subnet-bac383e0",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1c"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-d7c825e8",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1e"
      },
      "SubnetStatus": "Active"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Einrichten eines Netzwerks für eine Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateReplicationSubnetGroup](#) unter AWS CLI Befehlsreferenz.

create-replication-task

Das folgende Codebeispiel zeigt die Verwendung `create-replication-task`.

AWS CLI

Um eine Replikationsaufgabe zu erstellen

Im folgenden `create-replication-task` Beispiel wird eine Replikationsaufgabe erstellt.

```

aws dms create-replication-task \
  --replication-task-identifier movedata \
  --source-endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWNGAYUVLKIB732KEVWA \
  --target-endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U \
  --replication-instance-arn $RI_ARN \
  --migration-type full-load \
  --table-mappings file://table-mappings.json

```

Inhalt von table-mappings.json:

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "prodrep",
        "table-name": "%"
      },
      "rule-action": "include",
      "filters": []
    }
  ]
}
```

Ausgabe:

```
{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": "...output omitted... ",
    "ReplicationTaskSettings": "...output omitted... ",
    "Status": "creating",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskArn": "arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Aufgaben](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateReplicationTask](#) unter AWS CLI Befehlsreferenz.

delete-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-connection`.

AWS CLI

Um eine Verbindung zu löschen

Im folgenden `delete-connection` Beispiel wird die Zuordnung eines Endpunkts zu einer Replikationsinstanz aufgehoben.

```
aws dms delete-connection \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Ausgabe:

```
{  
  "Connection": {  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "Status": "deleting",  
    "EndpointIdentifier": "src-database-1",  
    "ReplicationInstanceIdentifier": "my-repl-instance"  
  }  
}
```

Weitere Informationen finden Sie unter https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.Creating.html im Database Migration Service Service-Benutzerhandbuch.AWS

- API-Einheiten finden Sie unter Befehlsreferenz. [DeleteConnection](#) AWS CLI

delete-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-endpoint`.

AWS CLI

Um einen Endpunkt zu löschen

Im folgenden delete-endpoint Beispiel wird ein Endpunkt gelöscht.

```
aws dms delete-endpoint \
  --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVX04XZ4CYTSEG5XGMN2R3Y
```

Ausgabe:

```
{
  "Endpoint": {
    "EndpointIdentifier": "src-endpoint",
    "EndpointType": "SOURCE",
    "EngineName": "s3",
    "EngineDisplayName": "Amazon S3",
    "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",
    "Status": "deleting",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVX04XZ4CYTSEG5XGMN2R3Y",
    "SslMode": "none",
    "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
    "S3Settings": {
      "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
      "CsvRowDelimiter": "\\n",
      "CsvDelimiter": ",",
      "BucketFolder": "sourcedata",
      "BucketName": "my-corp-data",
      "CompressionType": "NONE",
      "EnableStatistics": true
    }
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Endpunkten](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteEndpoint AWS CLI Befehlsreferenz](#).

delete-event-subscription

Das folgende Codebeispiel zeigt die Verwendung delete-event-subscription.

AWS CLI

Um ein Event-Abonnement zu löschen

Im folgenden `delete-event-subscription` Beispiel wird ein Abonnement für ein SNS Amazon-Thema gelöscht.

```
aws dms delete-event-subscription \  
  --subscription-name "my-dms-events"
```

Ausgabe:

```
{  
  "EventSubscription": {  
    "CustomerAwsId": "123456789012",  
    "CustSubscriptionId": "my-dms-events",  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",  
    "Status": "deleting",  
    "SubscriptionCreationTime": "2020-05-21 21:58:38.598",  
    "Enabled": true  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Ereignissen und Benachrichtigungen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteEventSubscription](#) unter AWS CLI Befehlsreferenz.

`delete-replication-instance`

Das folgende Codebeispiel zeigt die Verwendung `delete-replication-instance`.

AWS CLI

Um eine Replikationsinstanz zu löschen

Im folgenden `delete-replication-instance`-Beispiel wird eine Replikations-Instanz gelöscht.

```
aws dms delete-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

Ausgabe:

```
{
  "ReplicationInstance": {
    "ReplicationInstanceIdentifier": "my-repl-instance",
    "ReplicationInstanceClass": "dms.t2.micro",
    "ReplicationInstanceStatus": "deleting",
    "AllocatedStorage": 5,
    "InstanceCreateTime": 1590011235.952,
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-f839b688",
        "Status": "active"
      }
    ],
    "AvailabilityZone": "us-east-1e",
    "ReplicationSubnetGroup": {
      "ReplicationSubnetGroupIdentifier": "default",
      "ReplicationSubnetGroupDescription": "default",
      "VpcId": "vpc-136a4c6a",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-da327bf6",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-42599426",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1d"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-bac383e0",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1c"
          },
          "SubnetStatus": "Active"
        }
      ]
    }
  }
}
```

```

        "SubnetIdentifier": "subnet-6746046b",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-d7c825e8",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-cbfff283",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
        },
        "SubnetStatus": "Active"
    }
]
},
"PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
"PendingModifiedValues": {},
"MultiAZ": true,
"EngineVersion": "3.3.2",
"AutoMinorVersionUpgrade": true,
"KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
"ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
"ReplicationInstancePublicIpAddress": "54.225.120.92",
"ReplicationInstancePrivateIpAddress": "172.31.30.121",
"ReplicationInstancePublicIpAddresses": [
    "54.225.120.92",
    "3.230.18.248"
],
"ReplicationInstancePrivateIpAddresses": [
    "172.31.30.121",
    "172.31.75.90"
],
"PubliclyAccessible": true,
"SecondaryAvailabilityZone": "us-east-1b"
}

```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit einer AWS DMS Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteReplicationInstance](#) unter AWS CLI Befehlsreferenz.

delete-replication-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `delete-replication-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu löschen

Im folgenden `delete-replication-subnet-group` Beispiel wird eine Subnetzgruppe gelöscht.

```
aws dms delete-replication-subnet-group \  
--replication-subnet-group-identifier my-subnet-group
```

Ausgabe:

```
(none)
```

Weitere Informationen finden Sie unter [Einrichten eines Netzwerks für eine Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteReplicationSubnetGroup](#) unter AWS CLI Befehlsreferenz.

delete-replication-task

Das folgende Codebeispiel zeigt die Verwendung `delete-replication-task`.

AWS CLI

Um eine Replikationsaufgabe zu löschen

Im folgenden `delete-replication-task` Beispiel wird eine Replikationsaufgabe gelöscht.

```
aws dms delete-replication-task \  

```

```
--replication-task-arn arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Ausgabe:

```
{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": "...output omitted...",
    "ReplicationTaskSettings": "...output omitted...",
    "Status": "deleting",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590789988.677,
    "ReplicationTaskArn": "arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Aufgaben](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteReplicationTask](#) unter AWS CLI Befehlsreferenz.

describe-account-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-account-attributes`.

AWS CLI

Zur Beschreibung von Kontoattributen

Das folgende `describe-account-attributes` Beispiel listet die Attribute für Ihr AWS Konto auf.

```
aws dms describe-account-attributes
```

Ausgabe:

```
{
  "AccountQuotas": [
    {
      "AccountQuotaName": "ReplicationInstances",
      "Used": 1,
      "Max": 20
    },
    {
      "AccountQuotaName": "AllocatedStorage",
      "Used": 5,
      "Max": 10000
    },
    ...remaining output omitted...
  ],
  "UniqueAccountIdentifier": "cqahfbfy5xee"
}
```

- API-Einheiten finden Sie [DescribeAccountAttributes](#) in der AWS CLI Befehlsreferenz.

describe-certificates

Das folgende Codebeispiel zeigt die Verwendung `describe-certificates`.

AWS CLI

Um die verfügbaren Zertifikate aufzulisten

Das folgende `describe-certificates` Beispiel listet die verfügbaren Zertifikate in Ihrem AWS Konto auf.

```
aws dms describe-certificates
```

Ausgabe:

```
{
```

```

    "Certificates": [
      {
        "CertificateIdentifier": "my-cert",
        "CertificateCreationDate": 1543259542.506,
        "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"

        ... remaining output omitted ...

      }
    ]
  }

```

Weitere Informationen finden Sie unter [Verwenden SSL](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCertificates](#) unter AWS CLI Befehlsreferenz.

describe-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-connections`.

AWS CLI

Um Verbindungen zu beschreiben

Das folgende `describe-connections` Beispiel listet die Verbindungen auf, die Sie zwischen einer Replikationsinstanz und einem Endpunkt getestet haben.

```
aws dms describe-connections
```

Ausgabe:

```

{
  "Connections": [
    {
      "Status": "successful",
      "ReplicationInstanceIdentifier": "test",
      "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:ZW5UAN6P4E77EC7YWHK4RZZ3BE",
      "EndpointIdentifier": "testsrc1",
      "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:6UTDJGB0US3VI3SUWA66XFJCJQ"
    }
  ]
}

```



```

    }
  ]
}

```

Weitere Informationen finden Sie unter [Quell- und Zielendpunkte erstellen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeConnections AWS CLI Befehlsreferenz](#).

describe-endpoint-types

Das folgende Codebeispiel zeigt die Verwendung `describe-endpoint-types`.

AWS CLI

Um die verfügbaren Endpunkttypen aufzulisten

Das folgende `describe-endpoint-types` Beispiel listet die verfügbaren My SQL Endpunkt-Typen auf.

```

aws dms describe-endpoint-types \
  --filters "Name=engine-name,Values=mysql"

```

Ausgabe:

```

{
  "SupportedEndpointTypes": [
    {
      "EngineName": "mysql",
      "SupportsCDC": true,
      "EndpointType": "source",
      "EngineDisplayName": "MySQL"
    },
    {
      "EngineName": "mysql",
      "SupportsCDC": true,
      "EndpointType": "target",
      "EngineDisplayName": "MySQL"
    }
  ]
}

```

Weitere Informationen finden Sie unter Working with AWS DMS Endpoints < https://docs.aws.amazon.com/dms/latest/userguide/CHAP__endpoints.html >`__ im Database Migration Service Service-Benutzerhandbuch.AWS

- Einzelheiten finden Sie in der Befehlsreferenz. API [DescribeEndpointTypes](#)AWS CLI

describe-endpoints

Das folgende Codebeispiel zeigt die Verwendung describe-endpoints.

AWS CLI

Um Endpunkte zu beschreiben

Das folgende describe-endpoints Beispiel listet die Endpunkte in Ihrem AWS Konto auf.

```
aws dms describe-endpoints
```

Ausgabe:

```
{
  "Endpoints": [
    {
      "Username": "dms",
      "Status": "active",
      "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:SF2W0FLWYWKVE0HID2EKLP3SJI",
      "ServerName": "ec2-52-32-48-61.us-west-2.compute.amazonaws.com",
      "EndpointType": "SOURCE",
      "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/94d5c4e7-4e4c-44be-b58a-c8da7adf57cd",
      "DatabaseName": "test",
      "EngineName": "mysql",
      "EndpointIdentifier": "pri100",
      "Port": 8193
    },
    {
      "Username": "admin",
      "Status": "active",
      "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:TJJZCIH3CJ24TJRU4VC32WEWFR",
      "ServerName": "test.example.com",
      "EndpointType": "SOURCE",
    }
  ]
}
```

```
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/2431021b-1cf2-
a2d4-77b2-59a9e4bce323",
        "DatabaseName": "EMPL",
        "EngineName": "oracle",
        "EndpointIdentifier": "test",
        "Port": 1521
    }
]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Endpunkten](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeEndpoints AWS CLI Befehlsreferenz](#).

describe-event-categories

Das folgende Codebeispiel zeigt die Verwendung `describe-event-categories`.

AWS CLI

Um Ereigniskategorien zu beschreiben

Das folgende `describe-event-categories` Beispiel listet die verfügbaren Ereigniskategorien auf.

```
aws dms describe-event-categories
```

Ausgabe:

```
{
  "EventCategoryGroupList": [
    {
      "SourceType": "replication-instance",
      "EventCategories": [
        "low storage",
        "configuration change",
        "maintenance",
        "deletion",
        "creation",
        "failover",
        "failure"
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "SourceType": "replication-task",
    "EventCategories": [
      "configuration change",
      "state change",
      "deletion",
      "creation",
      "failure"
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Ereignissen und Benachrichtigungen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEventCategories](#) unter AWS CLI Befehlsreferenz.

describe-event-subscriptions

Das folgende Codebeispiel zeigt die Verwendung `describe-event-subscriptions`.

AWS CLI

Um Veranstaltungsabonnements zu beschreiben

Das folgende `describe-event-subscriptions` Beispiel listet die Veranstaltungsabonnements für ein SNS Amazon-Thema auf.

```
aws dms describe-event-subscriptions
```

Ausgabe:

```

{
  "EventSubscriptionsList": [
    {
      "CustomerAwsId": "123456789012",
      "CustSubscriptionId": "my-dms-events",
      "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
      "Status": "deleting",
      "SubscriptionCreationTime": "2020-05-21 22:28:51.924",

```

```
        "Enabled": true
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Arbeiten mit Ereignissen und Benachrichtigungen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEventSubscriptions](#) unter AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um DMS Ereignisse aufzulisten

Das folgende `describe-events` Beispiel listet die Ereignisse auf, die von einer Replikationsinstanz herrührten.

```
aws dms describe-events \
  --source-type "replication-instance"
```

Ausgabe:

```
{
  "Events": [
    {
      "SourceIdentifier": "my-repl-instance",
      "SourceType": "replication-instance",
      "Message": "Replication application shutdown",
      "EventCategories": [],
      "Date": 1590771645.776
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Ereignissen und Benachrichtigungen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEvents](#) unter AWS CLI Befehlsreferenz.

describe-orderable-replication-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-orderable-replication-instances`.

AWS CLI

Um bestellbare Replikationsinstanzen zu beschreiben

Das folgende `describe-orderable-replication-instances` Beispiel listet die Typen von Replikationsinstanzen auf, die Sie bestellen können.

```
aws dms describe-orderable-replication-instances
```

Ausgabe:

```
{
  "OrderableReplicationInstances": [
    {
      "EngineVersion": "3.3.2",
      "ReplicationInstanceClass": "dms.c4.2xlarge",
      "StorageType": "gp2",
      "MinAllocatedStorage": 5,
      "MaxAllocatedStorage": 6144,
      "DefaultAllocatedStorage": 100,
      "IncludedAllocatedStorage": 100,
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ]
    },
    {
      "EngineVersion": "3.3.2",
      "ReplicationInstanceClass": "dms.c4.4xlarge",
      "StorageType": "gp2",
      "MinAllocatedStorage": 5,
      "MaxAllocatedStorage": 6144,
      "DefaultAllocatedStorage": 100,
      "IncludedAllocatedStorage": 100,

```

```

    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1c",
      "us-east-1d",
      "us-east-1e",
      "us-east-1f"
    ]
  },
  ...remaining output omitted...
}

```

Weitere Informationen finden Sie unter [Arbeiten mit einer AWS DMS Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeOrderableReplicationInstances](#) unter AWS CLI Befehlsreferenz.

describe-refresh-schemas-status

Das folgende Codebeispiel zeigt die Verwendung `describe-refresh-schemas-status`.

AWS CLI

Um den Aktualisierungsstatus für einen Endpunkt aufzulisten

Das folgende `describe-refresh-schemas-status` Beispiel gibt den Status einer vorherigen Aktualisierungsanforderung zurück.

```

aws dms describe-refresh-schemas-status \
  --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA

```

Ausgabe:

```

{
  "RefreshSchemasStatus": {
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",

```

```
    "Status": "successful",
    "LastRefreshDate": 1590786544.605
  }
}
```

- API-Einheiten finden Sie [DescribeRefreshSchemasStatus](#) in der AWS CLI Befehlsreferenz.

describe-replication-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-replication-instances`.

AWS CLI

Um Replikationsinstanzen zu beschreiben

Das folgende `describe-replication-instances` Beispiel listet die Replikationsinstanzen in Ihrem AWS Konto auf.

```
aws dms describe-replication-instances
```

Ausgabe:

```
{
  "ReplicationInstances": [
    {
      "ReplicationInstanceIdentifier": "my-repl-instance",
      "ReplicationInstanceClass": "dms.t2.micro",
      "ReplicationInstanceStatus": "available",
      "AllocatedStorage": 5,
      "InstanceCreateTime": 1590011235.952,
      "VpcSecurityGroups": [
        {
          "VpcSecurityGroupId": "sg-f839b688",
          "Status": "active"
        }
      ],
      "AvailabilityZone": "us-east-1e",
      "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "default",
        "ReplicationSubnetGroupDescription": "default",
        "VpcId": "vpc-136a4c6a",
        "SubnetGroupStatus": "Complete",

```



```
"Subnets": [  
  {  
    "SubnetIdentifier": "subnet-da327bf6",  
    "SubnetAvailabilityZone": {  
      "Name": "us-east-1a"  
    },  
    "SubnetStatus": "Active"  
  },  
  {  
    "SubnetIdentifier": "subnet-42599426",  
    "SubnetAvailabilityZone": {  
      "Name": "us-east-1d"  
    },  
    "SubnetStatus": "Active"  
  },  
  {  
    "SubnetIdentifier": "subnet-bac383e0",  
    "SubnetAvailabilityZone": {  
      "Name": "us-east-1c"  
    },  
    "SubnetStatus": "Active"  
  },  
  {  
    "SubnetIdentifier": "subnet-6746046b",  
    "SubnetAvailabilityZone": {  
      "Name": "us-east-1f"  
    },  
    "SubnetStatus": "Active"  
  },  
  {  
    "SubnetIdentifier": "subnet-d7c825e8",  
    "SubnetAvailabilityZone": {  
      "Name": "us-east-1e"  
    },  
    "SubnetStatus": "Active"  
  },  
  {  
    "SubnetIdentifier": "subnet-cbfff283",  
    "SubnetAvailabilityZone": {  
      "Name": "us-east-1b"  
    },  
    "SubnetStatus": "Active"  
  }  
]
```

```

    },
    "PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
    "PendingModifiedValues": {
      "MultiAZ": true
    },
    },
    "MultiAZ": false,
    "EngineVersion": "3.3.2",
    "AutoMinorVersionUpgrade": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
f7bc0f8e-1a3a-4ace-9faa-e8494fa3921a",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "ReplicationInstancePublicIpAddress": "3.230.18.248",
    "ReplicationInstancePrivateIpAddress": "172.31.75.90",
    "ReplicationInstancePublicIpAddresses": [
      "3.230.18.248"
    ],
    "ReplicationInstancePrivateIpAddresses": [
      "172.31.75.90"
    ],
    },
    "PubliclyAccessible": true,
    "FreeUntil": 1590194829.267
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit einer AWS DMS Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReplicationInstances](#) unter AWS CLI Befehlsreferenz.

describe-replication-subnet-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-replication-subnet-groups`.

AWS CLI

Um die verfügbaren Subnetzgruppen anzuzeigen

Das folgende `describe-replication-subnet-groups` Beispiel listet die verfügbaren Subnetzgruppen auf.

```
aws dms describe-replication-subnet-groups \
```

```
--filter "Name=replication-subnet-group-id,Values=my-subnet-group"
```

Ausgabe:

```
{
  "ReplicationSubnetGroups": [
    {
      "ReplicationSubnetGroupIdentifier": "my-subnet-group",
      "ReplicationSubnetGroupDescription": "my subnet group",
      "VpcId": "vpc-136a4c6a",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-da327bf6",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-bac383e0",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1c"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-d7c825e8",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
          },
          "SubnetStatus": "Active"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Einrichten eines Netzwerks für eine Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReplicationSubnetGroups](#) unter AWS CLI Befehlsreferenz.

describe-replication-task-assessment-results

Das folgende Codebeispiel zeigt die Verwendung `describe-replication-task-assessment-results`.

AWS CLI

Um die Ergebnisse der Bewertungen von Replikationsaufgaben aufzulisten

Im folgenden `describe-replication-task-assessment-results` Beispiel werden die Ergebnisse einer früheren Aufgabenbeurteilung aufgeführt.

```
aws dms describe-replication-task-assessment-results
```

Ausgabe:

```
{
  "ReplicationTaskAssessmentResults": [
    {
      "ReplicationTaskIdentifier": "moveit2",
      "ReplicationTaskArn": "arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
      "ReplicationTaskLastAssessmentDate": 1590790230.0,
      "AssessmentStatus": "No issues found",
      "AssessmentResultsFile": "moveit2/2020-05-29-22-10"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erstellen eines Aufgabenbewertungsberichts](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReplicationTaskAssessmentResults](#) unter AWS CLI Befehlsreferenz.

describe-replication-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-replication-tasks`.

AWS CLI

Um eine Replikationsaufgabe zu beschreiben

Im folgenden `describe-replication-tasks` Beispiel werden aktuelle Replikationsaufgaben beschrieben.

```
aws dms describe-replication-tasks
```

Ausgabe:

```
{
  "ReplicationTasks": [
    {
      "ReplicationTaskIdentifier": "moveit2",
      "SourceEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
      "TargetEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
      "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
      "MigrationType": "full-load",
      "TableMappings": "...output omitted... ",
      "ReplicationTaskSettings": "...output omitted... ",
      "Status": "stopped",
      "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
      "ReplicationTaskCreationDate": 1590524772.505,
      "ReplicationTaskStartDate": 1590619805.212,
      "ReplicationTaskArn": "arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
      "ReplicationTaskStats": {
        "FullLoadProgressPercent": 100,
        "ElapsedTimeMillis": 0,
        "TablesLoaded": 0,
        "TablesLoading": 0,
        "TablesQueued": 0,
        "TablesErrored": 0,
        "FreshStartDate": 1590619811.528,
        "StartDate": 1590619811.528,
        "StopDate": 1590619842.068
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Aufgaben](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReplicationTasks](#) unter AWS CLI Befehlsreferenz.

describe-schemas

Das folgende Codebeispiel zeigt die Verwendung `describe-schemas`.

AWS CLI

Um Datenbankschemas zu beschreiben

Das folgende `describe-schemas` Beispiel listet die verfügbaren Tabellen an einem Endpunkt auf.

```
aws dms describe-schemas \
  --endpoint-arn "arn:aws:dms:us-
  east-1:123456789012:endpoint:6GGI6YPMWGAYUVLKIB732KEVWA"
```

Ausgabe:

```
{
  "Schemas": [
    "prodrep"
  ]
}
```

Weitere Informationen finden Sie unter [Dies ist der Thementitel](#) im AWS Database Migration Service User Guide.

- API-Einheiten finden Sie [DescribeSchemas](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für eine Replikationsinstanz auf.

```
aws dms list-tags-for-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Ausgabe:

```
{  
  "TagList": [  
    {  
      "Key": "Project",  
      "Value": "dbMigration"  
    },  
    {  
      "Key": "Environment",  
      "Value": "PROD"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Tagging Resources](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

modify-endpoint

Das folgende Codebeispiel zeigt die Verwendung `modify-endpoint`.

AWS CLI

Um einen Endpunkt zu ändern

Das folgende `modify-endpoint` Beispiel fügt einem Endpunkt ein zusätzliches Verbindungsattribut hinzu.

```
aws dms modify-endpoint \  
  --endpoint-arn "arn:aws:dms:us-east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U" \  
  --extra-connection-attributes "compressionType=GZIP"
```

Ausgabe:

```
{
  "Endpoint": {
    "EndpointIdentifier": "src-endpoint",
    "EndpointType": "SOURCE",
    "EngineName": "s3",
    "EngineDisplayName": "Amazon S3",
    "ExtraConnectionAttributes":
"compressionType=GZIP;csvDelimiter=,;csvRowDelimiter=\n;",
    "Status": "active",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U",
    "SslMode": "none",
    "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
    "S3Settings": {
      "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
      "CsvRowDelimiter": "\n",
      "CsvDelimiter": ",",
      "BucketFolder": "",
      "BucketName": "",
      "CompressionType": "GZIP",
      "EnableStatistics": true
    }
  }
}
```

Weitere Informationen finden Sie unter Working with AWS DMS Endpoints < https://docs.aws.amazon.com/dms/latest/userguide/CHAP__endpoints.html > im Database Migration Service Service-Benutzerhandbuch.AWS

- Einzelheiten finden Sie in der Befehlsreferenz. API [ModifyEndpoint](#) AWS CLI

modify-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `modify-event-subscription`.

AWS CLI

Um ein Event-Abonnement zu ändern

Im folgenden `modify-event-subscription` Beispiel wird der Quelltyp eines Ereignisabonnements geändert.

```
aws dms modify-event-subscription \  
  --subscription-name "my-dms-events" \  
  --source-type replication-task
```

Ausgabe:

```
{  
  "EventSubscription": {  
    "CustomerAwsId": "123456789012",  
    "CustSubscriptionId": "my-dms-events",  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",  
    "Status": "modifying",  
    "SubscriptionCreationTime": "2020-05-29 17:04:40.262",  
    "SourceType": "replication-task",  
    "Enabled": true  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Ereignissen und Benachrichtigungen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyEventSubscription](#) unter AWS CLI Befehlsreferenz.

modify-replication-instance

Das folgende Codebeispiel zeigt die Verwendung `modify-replication-instance`.

AWS CLI

Um eine Replikationsinstanz zu ändern

Im folgenden `modify-replication-instance` Beispiel wird eine Replikationsinstanz so geändert, dass sie eine Multi-AZ-Bereitstellung verwendet.

```
aws dms modify-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --multi-az
```

Ausgabe:

```
{
  "ReplicationInstance": {
    "ReplicationInstanceIdentifier": "my-repl-instance",
    "ReplicationInstanceClass": "dms.t2.micro",
    "ReplicationInstanceStatus": "available",
    "AllocatedStorage": 5,
    "InstanceCreateTime": 1590011235.952,

    ...output omitted...

    "PendingModifiedValues": {
      "MultiAZ": true
    },
    "MultiAZ": false,
    "EngineVersion": "3.3.2",
    "AutoMinorVersionUpgrade": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",

    ...output omitted...
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit einer AWS DMS Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyReplicationInstance](#) unter AWS CLI Befehlsreferenz.

modify-replication-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `modify-replication-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu ändern

Im folgenden `modify-replication-subnet-group` Beispiel werden die Listen der Subnetze geändert, die einer Subnetzgruppe zugeordnet sind.

```
aws dms modify-replication-subnet-group \
```

```
--replication-subnet-group-identifizier my-subnet-group \  
--subnet-id subnet-da327bf6 subnet-bac383e0
```

Ausgabe:

```
{  
  "ReplicationSubnetGroup": {  
    "ReplicationSubnetGroupIdentifizier": "my-subnet-group",  
    "ReplicationSubnetGroupDescription": "my subnet group",  
    "VpcId": "vpc-136a4c6a",  
    "SubnetGroupStatus": "Complete",  
    "Subnets": [  
      {  
        "SubnetIdentifizier": "subnet-da327bf6",  
        "SubnetAvailabilityZone": {  
          "Name": "us-east-1a"  
        },  
        "SubnetStatus": "Active"  
      },  
      {  
        "SubnetIdentifizier": "subnet-bac383e0",  
        "SubnetAvailabilityZone": {  
          "Name": "us-east-1c"  
        },  
        "SubnetStatus": "Active"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Einrichten eines Netzwerks für eine Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyReplicationSubnetGroup](#) unter AWS CLI Befehlsreferenz.

modify-replication-task

Das folgende Codebeispiel zeigt die Verwendung `modify-replication-task`.

AWS CLI

Um eine Replikationsaufgabe zu ändern

Im folgenden `modify-replication-task` Beispiel werden die Tabellenzuordnungen für eine Aufgabe geändert.

```
aws dms modify-replication-task \
  --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
  --table-mappings file://table-mappings.json
```

Inhalt von `table-mappings.json`:

```
{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "prodrep",
        "table-name": "ACCT_%"
      },
      "rule-action": "include",
      "filters": []
    }
  ]
}
```

Ausgabe:

```
{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": "...output omitted...",
    "ReplicationTaskSettings": "...output omitted...",
    "Status": "modifying",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
```

```

    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590789424.653,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Aufgaben](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyReplicationTask](#) unter AWS CLI Befehlsreferenz.

reboot-replication-instance

Das folgende Codebeispiel zeigt die Verwendung `reboot-replication-instance`.

AWS CLI

Um eine Replikationsinstanz neu zu starten

Im folgenden `reboot-replication-instance`-Beispiel wird eine Replikations-Instanz neu gestartet.

```

aws dms reboot-replication-instance \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE

```

Ausgabe:

```

{
  "ReplicationInstance": {
    "ReplicationInstanceIdentifier": "my-repl-instance",
    "ReplicationInstanceClass": "dms.t2.micro",
    "ReplicationInstanceStatus": "rebooting",
    "AllocatedStorage": 5,
    "InstanceCreateTime": 1590011235.952,
    ... output omitted ...
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit einer AWS DMS Replikationsinstanz](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [RebootReplicationInstance](#) unter AWS CLI Befehlsreferenz.

refresh-schemas

Das folgende Codebeispiel zeigt die Verwendung `refresh-schemas`.

AWS CLI

Um Datenbankschemas zu aktualisieren

Im folgenden `refresh-schemas` Beispiel werden Anfragen gestellt, die Liste der Schemas an einem Endpunkt zu AWS DMS aktualisieren.

```
aws dms refresh-schemas \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \
  --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

Ausgabe:

```
{
  "RefreshSchemasStatus": {
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "Status": "refreshing",
    "LastRefreshDate": 1590019949.103
  }
}
```

- API-Einheiten finden Sie unter [RefreshSchemas AWS CLI](#) Befehlsreferenz.

reload-tables

Das folgende Codebeispiel zeigt die Verwendung `reload-tables`.

AWS CLI

Um die Liste der an einem Endpunkt verfügbaren Tabellen zu aktualisieren

Im folgenden `reload-tables` Beispiel wird die Liste der verfügbaren Tabellen an einem Endpunkt neu geladen.

```
aws dms reload-tables \
  --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
  --tables-to-reload "SchemaName=prodrep,TableName=ACCT_BAL"
```

Ausgabe:

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- API-Einheiten finden Sie [ReloadTables](#) in der AWS CLI Befehlsreferenz.

remove-tags-from-resource

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-resource`.

AWS CLI

Um Tags aus einer Replikationsinstanz zu entfernen

Im folgenden `remove-tags-from-resource` Beispiel werden Tags aus einer Replikationsinstanz entfernt.

```
aws dms remove-tags-from-resource \
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
 \
  --tag-keys Environment Project
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Resources](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [RemoveTagsFromResource](#) in der AWS CLI Befehlsreferenz.

start-replication-task-assessment

Das folgende Codebeispiel zeigt die Verwendung `start-replication-task-assessment`.

AWS CLI

Um eine Aufgabenbeurteilung zu starten

Im folgenden `start-replication-task-assessment` Beispiel wird eine Bewertung der Replikationsaufgabe gestartet.

```
aws dms start-replication-task-assessment \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Ausgabe:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": "...output omitted...",  
    "ReplicationTaskSettings": "...output omitted...",  
    "Status": "testing",  
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590789988.677,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Aufgabenbewertungsberichts](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [StartReplicationTaskAssessment](#) unter AWS CLI Befehlsreferenz.

start-replication-task

Das folgende Codebeispiel zeigt die Verwendung `start-replication-task`.

AWS CLI

Um eine Replikationsaufgabe zu starten

Das folgende `command-name` Beispiel listet die verfügbaren Widgets in Ihrem AWS Konto auf.

```
aws dms start-replication-task \
  --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \
  --start-replication-task-type reload-target
```

Ausgabe:

```
{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": "...output omitted... ",
    "ReplicationTaskSettings": "...output omitted... ",
    "Status": "starting",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590619805.212,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Aufgaben](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [StartReplicationTask](#) unter AWS CLI Befehlsreferenz.

stop-replication-task

Das folgende Codebeispiel zeigt die Verwendung stop-replication-task.

AWS CLI

So beenden Sie eine Aufgabe

Im folgenden stop-replication-task Beispiel wird eine Aufgabe beendet.

```
aws dms stop-replication-task \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Ausgabe:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": "...output omitted...",  
    "ReplicationTaskSettings": "...output omitted...",  
    "Status": "stopping",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590789424.653,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS DMS Aufgaben](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie [StopReplicationTask](#) unter AWS CLI Befehlsreferenz.

test-connection

Das folgende Codebeispiel zeigt die Verwendung `test-connection`.

AWS CLI

Um eine Verbindung zu einem Endpunkt zu testen

Im folgenden `test-connection` Beispiel wird getestet, ob von einer Replikationsinstanz aus auf einen Endpunkt zugegriffen werden kann.

```
aws dms test-connection \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

Ausgabe:

```
{  
  "Connection": {  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "Status": "testing",  
    "EndpointIdentifier": "src-database-1",  
    "ReplicationInstanceIdentifier": "my-repl-instance"  
  }  
}
```

Weitere Informationen finden Sie unter [Quell- und Zielendpunkte erstellen](#) im AWS Database Migration Service Service-Benutzerhandbuch.

- API-Einheiten finden Sie unter [TestConnection AWS CLI](#) Befehlsreferenz.

Amazon DocumentDB DocumentDB-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon DocumentDB Aktionen ausführen und allgemeine Szenarien implementieren können.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-resource

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-resource`.

AWS CLI

Um einer bestimmten Ressource ein oder mehrere Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel werden drei Tags hinzugefügt `sample-cluster`. Ein Tag (`CropB`) hat einen Schlüsselnamen, aber keinen Wert.

```
aws docdb add-tags-to-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
  --tags Key="CropA",Value="Apple" Key="CropB" Key="CropC",Value="Corn"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Amazon DocumentDB DocumentDB-Ressourcen im Amazon DocumentDB DocumentDB-Entwicklerhandbuch](#).

- API-Einheiten finden Sie unter Befehlsreferenz [AddTagsToResource](#).AWS CLI

apply-pending-maintenance-action

Das folgende Codebeispiel zeigt die Verwendung `apply-pending-maintenance-action`.

AWS CLI

Damit ausstehende Wartungsaktionen während des nächsten Wartungsfensters ausgeführt werden

Im folgenden `apply-pending-maintenance-action` Beispiel werden alle Systemaktualisierungsaktionen während des nächsten geplanten Wartungsfensters ausgeführt.

```
aws docdb apply-pending-maintenance-action \  
--resource-identifier arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
--apply-action system-update \  
--opt-in-type next-maintenance
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Anwenden von Amazon DocumentDB DocumentDB-Updates](#) im Amazon DocumentDB DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [ApplyPendingMaintenanceAction AWS CLI Befehlsreferenz](#).

copy-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `copy-db-cluster-parameter-group`.

AWS CLI

Um eine bestehende DB-Cluster-Parametergruppe zu duplizieren

Im folgenden `copy-db-cluster-parameter-group` Beispiel wird eine Kopie der Parametergruppe `custom-docdb3-6` mit dem Namen `custom-docdb3-6-copy` erstellt. Beim Kopieren werden der neuen Parametergruppe Tags hinzugefügt.

```
aws docdb copy-db-cluster-parameter-group \  
--source-db-cluster-parameter-group-identifier custom-docdb3-6 \  
--target-db-cluster-parameter-group-identifier custom-docdb3-6-copy \  
--target-db-cluster-parameter-group-description "Copy of custom-docdb3-6" \  
--tags Key="CopyNumber",Value="1" Key="Modifiable",Value="Yes"
```

Ausgabe:

```
{  
  "DBClusterParameterGroup": {  
    "DBParameterGroupFamily": "docdb3.6",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:12345678901:cluster-  
pg:custom-docdb3-6-copy",  
    "DBClusterParameterGroupName": "custom-docdb3-6-copy",  
    "Description": "Copy of custom-docdb3-6"  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Kopieren einer Amazon DocumentDB-Cluster-Parametergruppe im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [CopyDbClusterParameterGroup AWS CLI](#) Befehlsreferenz.

copy-db-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-db-cluster-snapshot`.

AWS CLI

Um eine Kopie eines Snapshots zu erstellen

Das folgende `copy-db-cluster-snapshot`-Beispiel erstellt eine Kopie von `sample-cluster-snapshot` mit dem Namen `sample-cluster-snapshot-copy`. Die Kopie enthält alle Tags des Originals sowie ein neues Tag mit dem Schlüsselnamen `CopyNumber`.

```
aws docdb copy-db-cluster-snapshot \  
  --source-db-cluster-snapshot-identifizier sample-cluster-snapshot \  
  --target-db-cluster-snapshot-identifizier sample-cluster-snapshot-copy \  
  --copy-tags \  
  --tags Key="CopyNumber",Value="1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kopieren eines Cluster-Snapshots](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [CopyDbClusterSnapshot](#) in der AWS CLI Befehlsreferenz.

create-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster-parameter-group`.

AWS CLI

So erstellen Sie eine Amazon DocumentDB-Cluster-Parametergruppe

Im folgenden `create-db-cluster-parameter-group` Beispiel wird die DB-Cluster-Parametergruppe `sample-parameter-group` mithilfe der `docdb3.6` Familie erstellt.

```
aws docdb create-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --db-parameter-group-family docdb3.6 \  
  --description "Sample parameter group based on docdb3.6"
```

Ausgabe:

```
{  
  "DBClusterParameterGroup": {  
    "Description": "Sample parameter group based on docdb3.6",  
    "DBParameterGroupFamily": "docdb3.6",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-west-2:123456789012:cluster-  
pg:sample-parameter-group",  
    "DBClusterParameterGroupName": "sample-parameter-group"  
  }  
}
```

Weitere Informationen finden Sie unter [Creating an Amazon DocumentDB Cluster-Parametergruppe](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [CreateDbClusterParameterGroup AWS CLI Befehlsreferenz](#).

create-db-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster-snapshot`.

AWS CLI

Um einen manuellen Amazon DocumentDB-Cluster-Snapshot zu erstellen

Das folgende `create-db-cluster-snapshot` Beispiel erstellt einen Amazon DB-Cluster-Snapshot mit dem Namen `sample-cluster-snapshot`.

```
aws docdb create-db-cluster-snapshot \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Ausgabe:

```
{  
  "DBClusterSnapshot": {  
    "MasterUsername": "master-user",
```

```
"SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
"AvailabilityZones": [
  "us-west-2a",
  "us-west-2b",
  "us-west-2c",
  "us-west-2d",
  "us-west-2e",
  "us-west-2f"
],
"SnapshotType": "manual",
"DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
"EngineVersion": "3.6.0",
"PercentProgress": 0,
"DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
"Engine": "docdb",
"DBClusterIdentifier": "sample-cluster",
"Status": "creating",
"ClusterCreateTime": "2019-03-15T20:29:58.836Z",
"Port": 0,
"StorageEncrypted": false,
"VpcId": "vpc-91280df6"
}
}
```

Weitere Informationen finden Sie unter [Erstellen eines manuellen Cluster-Snapshots](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [CreateDbClusterSnapshot AWS CLI Befehlsreferenz](#).

create-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster`.

AWS CLI

So erstellen Sie einen Amazon DocumentDB-Cluster

Das folgende `create-db-cluster` Beispiel erstellt einen Amazon DocumentDB-Cluster `sample-cluster` mit dem Namen des bevorzugten Wartungsfensters an Sonntagen zwischen 20:30 und 11:00 Uhr.

```
aws docdb create-db-cluster \
```



```

--db-cluster-identifizier sample-cluster \
--engine docdb \
--master-username master-user \
--master-user-password password \
--preferred-maintenance-window Sun:20:30-Sun:21:00

```

Ausgabe:

```

{
  "DBCluster": {
    "DBClusterParameterGroup": "default.docdb3.6",
    "AssociatedRoles": [],
    "DBSubnetGroup": "default",
    "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
    "Status": "creating",
    "Port": 27017,
    "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "DBClusterMembers": [],
    "Engine": "docdb",
    "DBClusterIdentifier": "sample-cluster",
    "PreferredBackupWindow": "10:12-10:42",
    "AvailabilityZones": [
      "us-west-2d",
      "us-west-2f",
      "us-west-2e"
    ],
    "MasterUsername": "master-user",
    "BackupRetentionPeriod": 1,
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ],
    "StorageEncrypted": false,
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
    "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
    "MultiAZ": false,
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",

```

```
    "EngineVersion": "3.6.0"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen eines Amazon DocumentDB-Clusters im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [CreateDbCluster AWS CLI](#) Befehlsreferenz.

create-db-instance

Das folgende Codebeispiel zeigt die Verwendung `create-db-instance`.

AWS CLI

So erstellen Sie eine Amazon DocumentDB-Cluster-Instanz

Der folgende `create-db-instance` Beispielpcode erstellt die Instanz `sample-cluster-instance-2` im Amazon DocumentDB-Cluster `sample-cluster`.

```
aws docdb create-db-instance \
  --db-cluster-identifier sample-cluster \
  --db-instance-class db.r4.xlarge \
  --db-instance-identifier sample-cluster-instance-2 \
  --engine docdb
```

Ausgabe:

```
{
  "DBInstance": {
    "DBInstanceStatus": "creating",
    "PendingModifiedValues": {
      "PendingCloudwatchLogsExports": {
        "LogTypesToEnable": [
          "audit"
        ]
      }
    },
    "PubliclyAccessible": false,
    "PreferredBackupWindow": "00:00-00:30",
    "PromotionTier": 1,
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
```

```
"DBInstanceIdentifier": "sample-cluster-instance-2",
"PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
"StorageEncrypted": false,
"Engine": "docdb",
"DBClusterIdentifier": "sample-cluster",
"DBSubnetGroup": {
  "Subnets": [
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-4e26d263"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-afc329f4"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-53ab3636"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-991cb8d0"
    }
  ],
  "DBSubnetGroupDescription": "default",
  "SubnetGroupStatus": "Complete",
  "VpcId": "vpc-91280df6",
  "DBSubnetGroupName": "default"
},
"DBInstanceClass": "db.r4.xlarge",
"VpcSecurityGroups": [
  {
```

```

        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
    }
  ],
  "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
  "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4H03ITE"
}
}

```

Weitere Informationen finden Sie unter [Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [CreateDbInstance AWS CLI](#) Befehlsreferenz.

create-db-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `create-db-subnet-group`.

AWS CLI

So erstellen Sie eine Amazon DocumentDB-Subnetzgruppe

Im folgenden `create-db-subnet-group` Beispiel wird eine Amazon DocumentDB-Subnetzgruppe mit dem Namen `sample-subnet-group` erstellt.

```

aws docdb create-db-subnet-group \
  --db-subnet-group-description "a sample subnet group" \
  --db-subnet-group-name sample-subnet-group \
  --subnet-ids "subnet-29ab1025" "subnet-991cb8d0" "subnet-53ab3636"

```

Ausgabe:

```

{
  "DBSubnetGroup": {
    "SubnetGroupStatus": "Complete",
    "DBSubnetGroupName": "sample-subnet-group",
    "DBSubnetGroupDescription": "a sample subnet group",
    "VpcId": "vpc-91280df6",
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
  }
}

```

```
"Subnets": [  
  {  
    "SubnetStatus": "Active",  
    "SubnetIdentifier": "subnet-53ab3636",  
    "SubnetAvailabilityZone": {  
      "Name": "us-west-2d"  
    }  
  },  
  {  
    "SubnetStatus": "Active",  
    "SubnetIdentifier": "subnet-991cb8d0",  
    "SubnetAvailabilityZone": {  
      "Name": "us-west-2b"  
    }  
  },  
  {  
    "SubnetStatus": "Active",  
    "SubnetIdentifier": "subnet-29ab1025",  
    "SubnetAvailabilityZone": {  
      "Name": "us-west-2c"  
    }  
  }  
]
```

Weitere Informationen finden Sie unter [Creating a Amazon DocumentDB Subnet Group](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateDbSubnetGroup](#).AWS CLI

delete-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster-parameter-group`.

AWS CLI

Um eine Amazon DocumentDB-Cluster-Parametergruppe zu löschen

Im folgenden `delete-db-cluster-parameter-group` Beispiel wird die Amazon DocumentDB DocumentDB-Parametergruppe `sample-parameter-group`

```
aws docdb delete-db-cluster-parameter-group \
```

```
--db-cluster-parameter-group-name sample-parameter-group
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Amazon DocumentDB-Cluster-Parametergruppe](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [DeleteDbClusterParameterGroup AWS CLI-Befehlsreferenz](#).

delete-db-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster-snapshot`.

AWS CLI

Um einen Amazon DocumentDB-Cluster-Snapshot zu löschen

Im folgenden `delete-db-cluster-snapshot` Beispiel wird der Amazon DocumentDB-Cluster-Snapshot `sample-cluster-snapshot` gelöscht.

```
aws docdb delete-db-cluster-snapshot \  
--db-cluster-snapshot-identifizier sample-cluster-snapshot
```

Ausgabe:

```
{  
  "DBClusterSnapshot": {  
    "DBClusterIdentifizier": "sample-cluster",  
    "AvailabilityZones": [  
      "us-west-2a",  
      "us-west-2b",  
      "us-west-2c",  
      "us-west-2d"  
    ],  
    "DBClusterSnapshotIdentifizier": "sample-cluster-snapshot",  
    "VpcId": "vpc-91280df6",  
    "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-snapshot:sample-cluster-snapshot",  
    "EngineVersion": "3.6.0",  
    "Engine": "docdb",  
    "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",  
    "Status": "available",
```

```
    "MasterUsername": "master-user",
    "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
    "PercentProgress": 100,
    "StorageEncrypted": false,
    "SnapshotType": "manual",
    "Port": 0
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines Cluster-Snapshots](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DeleteDbClusterSnapshot](#) in der AWS CLI Befehlsreferenz.

delete-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster`.

AWS CLI

Um einen Amazon DocumentDB-Cluster zu löschen

Im folgenden `delete-db-cluster` Beispiel wird der Amazon DocumentDB-Cluster gelöscht. `sample-cluster` Vor dem Löschen des Clusters wird keine Sicherungskopie erstellt. NOTE: Sie müssen alle mit dem Cluster verknüpften Instanzen löschen, bevor Sie ihn löschen können.

```
aws docdb delete-db-cluster \
  --db-cluster-identifizier sample-cluster \
  --skip-final-snapshot
```

Ausgabe:

```
{
  "DBCluster": {
    "DBClusterIdentifizier": "sample-cluster",
    "DBSubnetGroup": "default",
    "EngineVersion": "3.6.0",
    "Engine": "docdb",
    "LatestRestorableTime": "2019-03-18T18:07:24.610Z",
    "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
    "StorageEncrypted": false,
    "EarliestRestorableTime": "2019-03-18T18:07:24.610Z",
```

```

    "Port": 27017,
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ],
    "MultiAZ": false,
    "MasterUsername": "master-user",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
    "Status": "available",
    "PreferredBackupWindow": "10:12-10:42",
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "AvailabilityZones": [
      "us-west-2c",
      "us-west-2b",
      "us-west-2a"
    ],
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
    "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
    "AssociatedRoles": [],
    "DBClusterParameterGroup": "default.docdb3.6",
    "HostedZoneId": "ZNKXH85TT8VWV",
    "BackupRetentionPeriod": 1,
    "DBClusterMembers": []
  }
}

```

Weitere Informationen finden Sie unter [Löschen eines Amazon DocumentDB-Clusters im Amazon DocumentDB DocumentDB-Entwicklerhandbuch](#).

- API-Einheiten finden Sie unter [DeleteDbCluster AWS CLI](#) Befehlsreferenz.

delete-db-instance

Das folgende Codebeispiel zeigt die Verwendung `delete-db-instance`.

AWS CLI

Um eine Amazon DocumentDB DocumentDB-Instance zu löschen

Im folgenden `delete-db-instance` Beispiel wird die Amazon DocumentDB DocumentDB-Instance gelöscht. `sample-cluster-instance-2`

```
aws docdb delete-db-instance \  
  --db-instance-identifier sample-cluster-instance-2
```

Ausgabe:

```
{  
  "DBInstance": {  
    "DBSubnetGroup": {  
      "Subnets": [  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2a"  
          },  
          "SubnetStatus": "Active",  
          "SubnetIdentifier": "subnet-4e26d263"  
        },  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2c"  
          },  
          "SubnetStatus": "Active",  
          "SubnetIdentifier": "subnet-afc329f4"  
        },  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2d"  
          },  
          "SubnetStatus": "Active",  
          "SubnetIdentifier": "subnet-53ab3636"  
        },  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2b"  
          },  
          "SubnetStatus": "Active",  
          "SubnetIdentifier": "subnet-991cb8d0"  
        }  
      ],  
      "DBSubnetGroupName": "default",  
      "DBSubnetGroupDescription": "default",
```

```
    "VpcId": "vpc-91280df6",
    "SubnetGroupStatus": "Complete"
  },
  "PreferredBackupWindow": "00:00-00:30",
  "InstanceCreateTime": "2019-03-18T18:37:33.709Z",
  "DBInstanceClass": "db.r4.xlarge",
  "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4H03ITE",
  "BackupRetentionPeriod": 3,
  "Engine": "docdb",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-77186e0d"
    }
  ],
  "AutoMinorVersionUpgrade": true,
  "PromotionTier": 1,
  "EngineVersion": "3.6.0",
  "Endpoint": {
    "Address": "sample-cluster-instance-2.corcjzrlsfc.us-
west-2.docdb.amazonaws.com",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "Port": 27017
  },
  "DBInstanceIdentifier": "sample-cluster-instance-2",
  "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
  "EnabledCloudwatchLogsExports": [
    "audit"
  ],
  "PendingModifiedValues": {},
  "DBInstanceStatus": "deleting",
  "PubliclyAccessible": false,
  "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
  "DBClusterIdentifier": "sample-cluster",
  "AvailabilityZone": "us-west-2c",
  "StorageEncrypted": false
}
}
```

Weitere Informationen finden Sie unter [Löschen einer Amazon DocumentDB DocumentDB-Instance im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DeleteDbInstance AWS CLI](#) Befehlsreferenz.

delete-db-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `delete-db-subnet-group`.

AWS CLI

Um eine Amazon DocumentDB-Subnetzgruppe zu löschen

Im folgenden `delete-db-subnet-group` Beispiel wird die Amazon DocumentDB-Subnetzgruppe `sample-subnet-group` gelöscht.

```
aws docdb delete-db-subnet-group \  
  --db-subnet-group-name sample-subnet-group
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Amazon DocumentDB-Subnetzgruppe im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteDbSubnetGroup](#).AWS CLI

describe-db-cluster-parameter-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-parameter-groups`.

AWS CLI

Um die Details einer oder mehrerer Amazon DocumentDB-Cluster-Parametergruppen anzuzeigen

Im folgenden `describe-db-cluster-parameter-groups` Beispiel werden Details für die Amazon DocumentDB-Cluster-Parametergruppe `custom3-6-param-grp` angezeigt.

```
aws docdb describe-db-cluster-parameter-groups \  
  --db-cluster-parameter-group-name custom3-6-param-grp
```

Ausgabe:

```
{  
  "DBClusterParameterGroups": [  
    {  
      "DBParameterGroupFamily": "docdb3.6",  
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:custom3-6-param-grp",
```

```
        "Description": "Custom docdb3.6 parameter group",
        "DBClusterParameterGroupName": "custom3-6-param-grp"
    }
]
}
```

Weitere Informationen finden Sie unter [Amazon DocumentDB Cluster-Parametergruppen anzeigen](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeDbClusterParameterGroups AWS CLI](#) Befehlsreferenz.

describe-db-cluster-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-parameters`.

AWS CLI

Um die detaillierte Parameterliste für eine Amazon DocumentDB-Cluster-Parametergruppe anzuzeigen.

Das folgende `describe-db-cluster-parameters` Beispiel listet die Parameter für die Amazon DocumentDB DocumentDB-Parametergruppe `custom3-6-param-grp` auf.

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name custom3-6-param-grp
```

Ausgabe:

```
{
  "Parameters": [
    {
      "DataType": "string",
      "ParameterName": "audit_logs",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot",
      "Source": "system",
      "ApplyType": "dynamic",
      "AllowedValues": "enabled,disabled",
      "Description": "Enables auditing on cluster.",
      "ParameterValue": "disabled"
    }
  ]
}
```

```
    },
    {
      "DataType": "string",
      "ParameterName": "tls",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot",
      "Source": "system",
      "ApplyType": "static",
      "AllowedValues": "disabled,enabled",
      "Description": "Config to enable/disable TLS",
      "ParameterValue": "enabled"
    },
    {
      "DataType": "string",
      "ParameterName": "ttl_monitor",
      "IsModifiable": true,
      "ApplyMethod": "pending-reboot",
      "Source": "user",
      "ApplyType": "dynamic",
      "AllowedValues": "disabled,enabled",
      "Description": "Enables TTL Monitoring",
      "ParameterValue": "enabled"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon DocumentDB-Cluster-Parameter anzeigen im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DescribeDbClusterParameters AWS CLI](#) Befehlsreferenz.

describe-db-cluster-snapshot-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-snapshot-attributes`.

AWS CLI

Um die Namen und Werte eines Amazon DocumentDB-Snapshot-Attributs aufzulisten

Das folgende `describe-db-cluster-snapshot-attributes` Beispiel listet die Attributnamen und -werte für den Amazon DocumentDB-Snapshot `sample-cluster-snapshot` auf.

```
aws docdb describe-db-cluster-snapshot-attributes \  
--db-cluster-snapshot-identifier sample-cluster-snapshot
```

Ausgabe:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": []  
      }  
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"  
  }  
}
```

Weitere Informationen finden Sie unter [DescribeDBClusterSnapshotAttributes](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribeDbClusterSnapshotAttributes](#) in der AWS CLI Befehlsreferenz.

describe-db-cluster-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-snapshots`.

AWS CLI

Um Amazon DocumentDB-Snapshots zu beschreiben

Im folgenden `describe-db-cluster-snapshots` Beispiel werden Details für den Amazon DocumentDB-Snapshot `sample-cluster-snapshot` angezeigt.

```
aws docdb describe-db-cluster-snapshots \  
--db-cluster-snapshot-identifier sample-cluster-snapshot
```

Ausgabe:

```
{
```

```

"DBClusterSnapshots": [
  {
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2b",
      "us-west-2c",
      "us-west-2d"
    ],
    "Status": "available",
    "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
    "SnapshotCreateTime": "2019-03-15T20:41:26.515Z",
    "SnapshotType": "manual",
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "DBClusterIdentifier": "sample-cluster",
    "MasterUsername": "master-user",
    "StorageEncrypted": false,
    "VpcId": "vpc-91280df6",
    "EngineVersion": "3.6.0",
    "PercentProgress": 100,
    "Port": 0,
    "Engine": "docdb",
    "ClusterCreateTime": "2019-03-15T20:29:58.836Z"
  }
]
}

```

Weitere Informationen finden Sie unter [D escribeDBCluster Snapshots](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribeDbClusterSnapshots](#) in der AWS CLI Befehlsreferenz.

describe-db-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-db-clusters`.

AWS CLI

Um detaillierte Informationen über einen oder mehrere Amazon DocumentDB-Cluster zu erhalten.

Im folgenden `describe-db-clusters` Beispiel werden Details für den Amazon DocumentDB-Cluster `sample-cluster` angezeigt. Wenn Sie den `--db-cluster-identifier` Parameter weglassen, können Sie Informationen von bis zu 100 Clustern abrufen.

```
aws docdb describe-db-clusters
--db-cluster-identifier sample-cluster
```

Ausgabe:

```
{
  "DBClusters": [
    {
      "DBClusterParameterGroup": "default.docdb3.6",
      "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
      "PreferredBackupWindow": "00:00-00:30",
      "DBClusterIdentifier": "sample-cluster",
      "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
      "LatestRestorableTime": "2019-03-18T20:28:03.239Z",
      "MasterUsername": "master-user",
      "DBClusterMembers": [
        {
          "PromotionTier": 1,
          "DBClusterParameterGroupStatus": "in-sync",
          "IsClusterWriter": false,
          "DBInstanceIdentifier": "sample-cluster"
        },
        {
          "PromotionTier": 1,
          "DBClusterParameterGroupStatus": "in-sync",
          "IsClusterWriter": true,
          "DBInstanceIdentifier": "sample-cluster2"
        }
      ],
      "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
      "VpcSecurityGroups": [
        {
          "VpcSecurityGroupId": "sg-77186e0d",
          "Status": "active"
        }
      ],
      "Engine": "docdb",
      "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
      "DBSubnetGroup": "default",
      "MultiAZ": true,
      "AvailabilityZones": [
```



```

        "us-west-2a",
        "us-west-2c",
        "us-west-2b"
    ],
    "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
    "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
    "BackupRetentionPeriod": 3,
    "HostedZoneId": "ZNKXH85TT8WVW",
    "StorageEncrypted": false,
    "EnabledCloudwatchLogsExports": [
        "audit"
    ],
    "AssociatedRoles": [],
    "EngineVersion": "3.6.0",
    "Port": 27017,
    "Status": "available"
    }
]
}

```

Weitere Informationen finden Sie unter [Beschreibung von Amazon DocumentDB-Clustern im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DescribeDbClusters AWS CLI](#) Befehlsreferenz.

describe-db-engine-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-db-engine-versions`.

AWS CLI

Um verfügbare Versionen der Amazon DocumentDB DocumentDB-Engine aufzulisten

Das folgende `describe-db-engine-versions` Beispiel listet alle verfügbaren Versionen der Amazon DocumentDB DocumentDB-Engine auf.

```
aws docdb describe-db-engine-versions \
    --engine docdb
```

Ausgabe:

```
{
  "DBEngineVersions": [
    {
      "DBEngineVersionDescription": "DocDB version 1.0.200837",
      "DBParameterGroupFamily": "docdb3.6",
      "EngineVersion": "3.6.0",
      "ValidUpgradeTarget": [],
      "DBEngineDescription": "Amazon DocumentDB (with MongoDB compatibility)",
      "SupportsLogExportsToCloudwatchLogs": true,
      "Engine": "docdb",
      "ExportableLogTypes": [
        "audit"
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [describeDBEngineD-Versionen](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribeDbEngineVersions](#) in der AWS CLI Befehlsreferenz.

describe-db-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-db-instances`.

AWS CLI

Um Informationen über bereitgestellte Amazon DocumentDB DocumentDB-Instances zu finden

Im folgenden `describe-db-instances` Beispiel werden Details zur Amazon DocumentDB DocumentDB-Instance `sample-cluster-instance` angezeigt. Wenn Sie den `--db-instance-identifizier` Parameter weglassen, erhalten Sie Informationen zu bis zu 100 Instances.

```
aws docdb describe-db-instances \
  --db-instance-identifizier sample-cluster-instance
```

Ausgabe:

```
{
  "DBInstances": [
```

```
{
  "Endpoint": {
    "HostedZoneId": "ZNKXH85TT8WW",
    "Address": "sample-cluster-instance.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "Port": 27017
  },
  "PreferredBackupWindow": "00:00-00:30",
  "DBInstanceStatus": "available",
  "DBInstanceClass": "db.r4.large",
  "EnabledCloudwatchLogsExports": [
    "audit"
  ],
  "DBInstanceIdentifier": "sample-cluster-instance",
  "DBSubnetGroup": {
    "Subnets": [
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-4e26d263",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      },
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-afc329f4",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2c"
        }
      },
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-53ab3636",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        }
      },
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-991cb8d0",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        }
      }
    ]
  }
}
```

```

    ],
    "DBSubnetGroupName": "default",
    "SubnetGroupStatus": "Complete",
    "DBSubnetGroupDescription": "default",
    "VpcId": "vpc-91280df6"
  },
  "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
  "Engine": "docdb",
  "StorageEncrypted": false,
  "AutoMinorVersionUpgrade": true,
  "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance",
  "PreferredMaintenanceWindow": "tue:08:39-tue:09:09",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-77186e0d"
    }
  ],
  "DBClusterIdentifier": "sample-cluster",
  "PendingModifiedValues": {},
  "BackupRetentionPeriod": 3,
  "PubliclyAccessible": false,
  "EngineVersion": "3.6.0",
  "PromotionTier": 1,
  "AvailabilityZone": "us-west-2c",
  "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA"
}
]
}

```

Weitere Informationen finden Sie unter [Beschreibung von Amazon DocumentDB DocumentDB-Instances](#) im Amazon DocumentDB DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DescribeDbInstances AWS CLI](#) Befehlsreferenz.

describe-db-subnet-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-subnet-groups`.

AWS CLI

Um eine Liste mit Amazon DocumentDB-Subnetzbeschreibungen abzurufen

Das folgende `describe-db-subnet-groups` Beispiel beschreibt Details für das Amazon DocumentDB-Subnetz mit dem Namen `default`

```
aws docdb describe-db-subnet-groups \  
--db-subnet-group-name default
```

Ausgabe:

```
{  
  "DBSubnetGroups": [  
    {  
      "VpcId": "vpc-91280df6",  
      "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:default",  
      "Subnets": [  
        {  
          "SubnetIdentifier": "subnet-4e26d263",  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2a"  
          }  
        },  
        {  
          "SubnetIdentifier": "subnet-afc329f4",  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2c"  
          }  
        },  
        {  
          "SubnetIdentifier": "subnet-53ab3636",  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2d"  
          }  
        },  
        {  
          "SubnetIdentifier": "subnet-991cb8d0",  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2b"  
          }  
        }  
      ],  
    }  
  ],  
}
```

```
        "DBSubnetGroupName": "default",
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupDescription": "default"
    }
]
}
```

Weitere Informationen finden Sie unter [Beschreibung von Subnetzgruppen](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribeDbSubnetGroups](#) in der AWS CLI Befehlsreferenz.

describe-engine-default-cluster-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-engine-default-cluster-parameters`.

AWS CLI

Um die Standard-Engine- und Systemparameterinformationen für Amazon DocumentDB zu beschreiben

Im folgenden `describe-engine-default-cluster-parameters` Beispiel werden Details zur Standard-Engine und Systemparameterinformationen für die Amazon DocumentDB DocumentDB-Parametergruppe `docdb3.6` angezeigt.

```
aws docdb describe-engine-default-cluster-parameters \
  --db-parameter-group-family docdb3.6
```

Ausgabe:

```
{
  "EngineDefaults": {
    "DBParameterGroupFamily": "docdb3.6",
    "Parameters": [
      {
        "ApplyType": "dynamic",
        "ParameterValue": "disabled",
        "Description": "Enables auditing on cluster.",
        "Source": "system",
        "DataType": "string",
        "MinimumEngineVersion": "3.6.0",
```

```

        "AllowedValues": "enabled,disabled",
        "ParameterName": "audit_logs",
        "IsModifiable": true
    },
    {
        "ApplyType": "static",
        "ParameterValue": "enabled",
        "Description": "Config to enable/disable TLS",
        "Source": "system",
        "DataType": "string",
        "MinimumEngineVersion": "3.6.0",
        "AllowedValues": "disabled,enabled",
        "ParameterName": "tls",
        "IsModifiable": true
    },
    {
        "ApplyType": "dynamic",
        "ParameterValue": "enabled",
        "Description": "Enables TTL Monitoring",
        "Source": "system",
        "DataType": "string",
        "MinimumEngineVersion": "3.6.0",
        "AllowedValues": "disabled,enabled",
        "ParameterName": "ttl_monitor",
        "IsModifiable": true
    }
]
}
}

```

Weitere Informationen finden Sie [DescribeEngineDefaultClusterParameters](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribeEngineDefaultClusterParameters](#) in der AWS CLI Befehlsreferenz.

describe-event-categories

Das folgende Codebeispiel zeigt die Verwendung `describe-event-categories`.

AWS CLI

Um alle Amazon DocumentDB DocumentDB-Ereigniskategorien zu beschreiben

Das folgende `describe-event-categories` Beispiel listet alle Kategorien für den Amazon DocumentDB DocumentDB-Ereignisquellentyp `db-instance` auf.

```
aws docdb describe-event-categories \  
  --source-type db-cluster
```

Ausgabe:

```
{  
  "EventCategoriesMapList": [  
    {  
      "SourceType": "db-cluster",  
      "EventCategories": [  
        "failover",  
        "maintenance",  
        "notification",  
        "failure"  
      ]  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Event-Kategorien anzeigen](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribeEventCategories](#) in der AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um Amazon DocumentDB DocumentDB-Ereignisse aufzulisten

Das folgende `describe-events` Beispiel listet alle Amazon DocumentDB DocumentDB-Ereignisse der letzten 24 Stunden (1440 Minuten) auf.

```
aws docdb describe-events \  
  --duration 1440
```


Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "Events": [
    {
      "EventCategories": [
        "failover"
      ],
      "Message": "Started cross AZ failover to DB instance: sample-cluster",
      "Date": "2019-03-18T21:36:29.807Z",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
      "SourceIdentifier": "sample-cluster",
      "SourceType": "db-cluster"
    },
    {
      "EventCategories": [
        "availability"
      ],
      "Message": "DB instance restarted",
      "Date": "2019-03-18T21:36:40.793Z",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster",
      "SourceIdentifier": "sample-cluster",
      "SourceType": "db-instance"
    },
    {
      "EventCategories": [],
      "Message": "A new writer was promoted. Restarting database as a reader.",
      "Date": "2019-03-18T21:36:43.873Z",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
      "SourceIdentifier": "sample-cluster2",
      "SourceType": "db-instance"
    },
    {
      "EventCategories": [
        "availability"
      ],
      "Message": "DB instance restarted",
      "Date": "2019-03-18T21:36:51.257Z",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
      "SourceIdentifier": "sample-cluster2",
      "SourceType": "db-instance"
    }
  ]
}
```

```
{
  "EventCategories": [
    "failover"
  ],
  "Message": "Completed failover to DB instance: sample-cluster",
  "Date": "2019-03-18T21:36:53.462Z",
  "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
  "SourceIdentifier": "sample-cluster",
  "SourceType": "db-cluster"
},
{
  "Date": "2019-03-19T16:51:48.847Z",
  "EventCategories": [
    "configuration change"
  ],
  "Message": "Updated parameter audit_logs to enabled with apply method
pending-reboot",
  "SourceIdentifier": "custom3-6-param-grp",
  "SourceType": "db-parameter-group"
},
{
  "EventCategories": [
    "configuration change"
  ],
  "Message": "Applying modification to database instance class",
  "Date": "2019-03-19T17:55:20.095Z",
  "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
  "SourceIdentifier": "sample-cluster2",
  "SourceType": "db-instance"
},
{
  "EventCategories": [
    "availability"
  ],
  "Message": "DB instance shutdown",
  "Date": "2019-03-19T17:56:31.127Z",
  "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
  "SourceIdentifier": "sample-cluster2",
  "SourceType": "db-instance"
},
{
  "EventCategories": [
    "configuration change"
```

```
    ],
    "Message": "Finished applying modification to DB instance class",
    "Date": "2019-03-19T18:00:45.822Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
  },
  {
    "EventCategories": [
      "availability"
    ],
    "Message": "DB instance restarted",
    "Date": "2019-03-19T18:00:53.397Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
  },
  {
    "EventCategories": [
      "availability"
    ],
    "Message": "DB instance shutdown",
    "Date": "2019-03-19T18:23:36.045Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
  },
  {
    "EventCategories": [
      "availability"
    ],
    "Message": "DB instance restarted",
    "Date": "2019-03-19T18:23:46.209Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
  },
  {
    "Date": "2019-03-19T18:39:05.822Z",
    "EventCategories": [
      "configuration change"
    ],
    "Message": "Updated parameter ttl_monitor to enabled with apply method
immediate",
```

```

        "SourceIdentifier": "custom3-6-param-grp",
        "SourceType": "db-parameter-group"
    },
    {
        "Date": "2019-03-19T18:39:48.067Z",
        "EventCategories": [
            "configuration change"
        ],
        "Message": "Updated parameter audit_logs to disabled with apply method
immediate",
        "SourceIdentifier": "custom3-6-param-grp",
        "SourceType": "db-parameter-group"
    }
]
}

```

Weitere Informationen finden Sie unter [Amazon DocumentDB DocumentDB-Ereignisse anzeigen im Amazon DocumentDB DocumentDB-Entwicklerhandbuch](#).

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

describe-orderable-db-instance-options

Das folgende Codebeispiel zeigt die Verwendung `describe-orderable-db-instance-options`.

AWS CLI

Um die Amazon DocumentDB DocumentDB-Instance-Optionen zu finden, können Sie bestellen

Das folgende `describe-orderable-db-instance-options` Beispiel listet alle Instance-Optionen für Amazon DocumentDB für eine Region auf.

```

aws docdb describe-orderable-db-instance-options \
  --engine docdb \
  --region us-east-1

```

Ausgabe:

```

{
  "OrderableDBInstanceOptions": [
    {
      "Vpc": true,
      "AvailabilityZones": [

```

```
        {
            "Name": "us-east-1a"
        },
        {
            "Name": "us-east-1b"
        },
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.16xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
},
{
    "Vpc": true,
    "AvailabilityZones": [
        {
            "Name": "us-east-1a"
        },
        {
            "Name": "us-east-1b"
        },
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.2xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
},
{
    "Vpc": true,
    "AvailabilityZones": [
        {
```

```
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.4xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.8xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },

```

```
        {
            "Name": "us-east-1b"
        },
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.large",
    "LicenseModel": "na",
    "Engine": "docdb"
},
{
    "Vpc": true,
    "AvailabilityZones": [
        {
            "Name": "us-east-1a"
        },
        {
            "Name": "us-east-1b"
        },
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
}
]
}
```

Weitere Informationen finden Sie unter [Hinzufügen einer Amazon DocumentDB-Instanz zu einem Cluster](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeOrderableDbInstanceOptions AWS CLI-Befehlsreferenz](#).

describe-pending-maintenance-actions

Das folgende Codebeispiel zeigt die Verwendung `describe-pending-maintenance-actions`.

AWS CLI

Um Ihre ausstehenden Amazon DocumentDB DocumentDB-Wartungsmaßnahmen aufzulisten

Das folgende `describe-pending-maintenance-actions` Beispiel listet all Ihre ausstehenden Amazon DocumentDB DocumentDB-Wartungsaktionen auf.

```
aws docdb describe-pending-maintenance-actions
```

Ausgabe:

```
{
  "PendingMaintenanceActions": []
}
```

Weitere Informationen finden Sie unter [Wartung von Amazon DocumentDB](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [DescribePendingMaintenanceActions](#) in der AWS CLI Befehlsreferenz.

failover-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `failover-db-cluster`.

AWS CLI

So erzwingen Sie ein Failover eines Amazon DocumentDB-Clusters auf ein Replikat

Im folgenden `failover-db-cluster` Beispiel wird für die primäre Instance im Amazon DocumentDB-Cluster-Beispielcluster ein Failover auf ein Replikat ausgeführt.

```
aws docdb failover-db-cluster \
  --db-cluster-identifier sample-cluster
```

Ausgabe:

```
{
```



```
"DBCluster": {
  "AssociatedRoles": [],
  "DBClusterIdentifier": "sample-cluster",
  "EngineVersion": "3.6.0",
  "DBSubnetGroup": "default",
  "MasterUsername": "master-user",
  "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
  "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
  "AvailabilityZones": [
    "us-west-2a",
    "us-west-2c",
    "us-west-2b"
  ],
  "LatestRestorableTime": "2019-03-18T21:35:23.548Z",
  "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
  "PreferredBackupWindow": "00:00-00:30",
  "Port": 27017,
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sg-77186e0d",
      "Status": "active"
    }
  ],
  "StorageEncrypted": false,
  "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
  "MultiAZ": true,
  "Status": "available",
  "DBClusterMembers": [
    {
      "DBClusterParameterGroupStatus": "in-sync",
      "IsClusterWriter": false,
      "DBInstanceIdentifier": "sample-cluster",
      "PromotionTier": 1
    },
    {
      "DBClusterParameterGroupStatus": "in-sync",
      "IsClusterWriter": true,
      "DBInstanceIdentifier": "sample-cluster2",
      "PromotionTier": 2
    }
  ],
  "EnabledCloudwatchLogsExports": [
    "audit"
  ]
}
```

```

    ],
    "DBClusterParameterGroup": "default.docdb3.6",
    "HostedZoneId": "ZNKXH85TT8WW",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
    "BackupRetentionPeriod": 3,
    "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "Engine": "docdb"
  }
}

```

Weitere Informationen finden Sie unter [Amazon DocumentDB Failover](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [FailoverDbCluster](#) in AWS CLI der Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um alle Tags in einer Amazon DocumentDB DocumentDB-Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags im Amazon DocumentDB-Cluster `sample-cluster` auf.

```

aws docdb list-tags-for-resource \
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster

```

Ausgabe:

```

{
  "TagList": [
    {
      "Key": "A",
      "Value": "ALPHA"
    },
    {
      "Key": "B",
      "Value": ""
    }
  ],
}

```

```
{
  "Key": "C",
  "Value": "CHARLIE"
}
]
```

Weitere Informationen finden Sie unter [Auflisten von Tags in einer Amazon DocumentDB DocumentDB-Ressource im Amazon DocumentDB DocumentDB-Entwicklerhandbuch](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

modify-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `modify-db-cluster-parameter-group`.

AWS CLI

So ändern Sie eine Amazon DocumentDB-DB-Cluster-Parametergruppe

Im folgenden `modify-db-cluster-parameter-group` Beispiel wird die Amazon DocumentDB-Cluster-Parametergruppe geändert, `custom3-6-param-grp` indem die beiden Parameter `audit_logs` und `ttl_monitor` auf `enabled` gesetzt werden. Die Änderungen werden beim nächsten Neustart übernommen.

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name custom3-6-param-grp \
  --
parameters ParameterName=audit_logs,ParameterValue=enabled,ApplyMethod=pending-reboot \
ParameterName=ttl_monitor,ParameterValue=enabled,ApplyMethod=pending-reboot
```

Ausgabe:

```
{
  "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

Weitere Informationen finden Sie unter [Ändern einer Amazon DocumentDB-Cluster-Parametergruppe](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie unter [ModifyDbClusterParameterGroup AWS CLI Befehlsreferenz](#).

modify-db-cluster-snapshot-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-db-cluster-snapshot-attribute`.

AWS CLI

Beispiel 1: Um einem Amazon DocumentDB-Snapshot ein Attribut hinzuzufügen

Das folgende `modify-db-cluster-snapshot-attribute` Beispiel fügt einem Amazon DocumentDB-Cluster-Snapshot vier Attributwerte hinzu.

```
aws docdb modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --attribute-name restore \  
  --values-to-add 123456789011 123456789012 123456789013
```

Ausgabe:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789011",  
          "123456789012",  
          "123456789013"  
        ]  
      }  
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"  
  }  
}
```

Beispiel 2: So entfernen Sie Attribute aus einem Amazon DocumentDB-Snapshot

Das folgende `modify-db-cluster-snapshot-attribute` Beispiel entfernt zwei Attributwerte aus einem Amazon DocumentDB-Cluster-Snapshot.

```
aws docdb modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --attribute-name restore \  
  --values-to-add 123456789011 123456789012 123456789013
```

```
--db-cluster-snapshot-identifizier sample-cluster-snapshot \  
--attribute-name restore \  
--values-to-remove 123456789012
```

Ausgabe:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789011",  
          "123456789013"  
        ]  
      }  
    ],  
    "DBClusterSnapshotIdentifizier": "sample-cluster-snapshot"  
  }  
}
```

Weitere Informationen finden Sie unter [ModifyDBCluster SnapshotAttribute](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [ModifyDbClusterSnapshotAttribute](#) in der AWS CLI Befehlsreferenz.

modify-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `modify-db-cluster`.

AWS CLI

So ändern Sie einen Amazon DocumentDB-Cluster

Im folgenden `modify-db-cluster` Beispiel wird der Amazon DocumentDB-Cluster geändert, `sample-cluster` indem die Aufbewahrungsfrist für automatische Backups auf 7 Tage festgelegt und die bevorzugten Fenster für Backups und Wartung geändert werden. Alle Änderungen werden im nächsten Wartungsfenster übernommen.

```
aws docdb modify-db-cluster \  
--db-cluster-identifizier sample-cluster \  
--no-apply-immediately \  

```

```
--backup-retention-period 7 \
--preferred-backup-window 18:00-18:30 \
--preferred-maintenance-window sun:20:00-sun:20:30
```

Ausgabe:

```
{
  "DBCluster": {
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DBClusterMembers": [
      {
        "DBClusterParameterGroupStatus": "in-sync",
        "DBInstanceIdentifier": "sample-cluster",
        "IsClusterWriter": true,
        "PromotionTier": 1
      },
      {
        "DBClusterParameterGroupStatus": "in-sync",
        "DBInstanceIdentifier": "sample-cluster2",
        "IsClusterWriter": false,
        "PromotionTier": 2
      }
    ],
    "HostedZoneId": "ZNKXH85TT8WVW",
    "StorageEncrypted": false,
    "PreferredBackupWindow": "18:00-18:30",
    "MultiAZ": true,
    "EngineVersion": "3.6.0",
    "MasterUsername": "master-user",
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DBSubnetGroup": "default",
    "LatestRestorableTime": "2019-03-18T22:08:13.408Z",
    "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
    "PreferredMaintenanceWindow": "sun:20:00-sun:20:30",
    "AssociatedRoles": [],
    "EnabledCloudwatchLogsExports": [
      "audit"
    ],
    "Engine": "docdb",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
```

```
    "BackupRetentionPeriod": 7,
    "DBClusterIdentifier": "sample-cluster",
    "AvailabilityZones": [
        "us-west-2a",
        "us-west-2c",
        "us-west-2b"
    ],
    "Status": "available",
    "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
    "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-77186e0d",
            "Status": "active"
        }
    ],
    "Port": 27017
}
}
```

Weitere Informationen finden Sie unter [Ändern eines Amazon DocumentDB-Clusters im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [ModifyDbCluster AWS CLI](#) Befehlsreferenz.

modify-db-instance

Das folgende Codebeispiel zeigt die Verwendung `modify-db-instance`.

AWS CLI

So ändern Sie eine Amazon DocumentDB DocumentDB-Instance

Im folgenden `modify-db-instance` Beispiel wird die Amazon DocumentDB DocumentDB-Instance geändert, `sample-cluster2` indem ihre Instance-Klasse auf `db.r4.4xlarge` und ihre Promotion-Stufe auf geändert wird. 5 Die Änderungen werden sofort übernommen, sind aber erst sichtbar, wenn der Instance-Status verfügbar ist.

```
aws docdb modify-db-instance \
  --db-instance-identifier sample-cluster2 \
  --apply-immediately \
  --db-instance-class db.r4.4xlarge \
```

--promotion-tier 5

Ausgabe:

```
{
  "DBInstance": {
    "EngineVersion": "3.6.0",
    "StorageEncrypted": false,
    "DBInstanceClass": "db.r4.large",
    "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
    "AutoMinorVersionUpgrade": true,
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ],
    "PreferredBackupWindow": "18:00-18:30",
    "EnabledCloudwatchLogsExports": [
      "audit"
    ],
    "AvailabilityZone": "us-west-2f",
    "DBInstanceIdentifier": "sample-cluster2",
    "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
    "Engine": "docdb",
    "BackupRetentionPeriod": 7,
    "DBSubnetGroup": {
      "DBSubnetGroupName": "default",
      "DBSubnetGroupDescription": "default",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-4e26d263",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2a"
          }
        },
        {
          "SubnetIdentifier": "subnet-afc329f4",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2c"
          }
        }
      ]
    }
  }
}
```



```

        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-53ab3636",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2d"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-991cb8d0",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
    }
],
"VpcId": "vpc-91280df6"
},
"PromotionTier": 2,
"Endpoint": {
    "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
    "HostedZoneId": "ZNKXH85TT8WW",
    "Port": 27017
},
"DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
"DBClusterIdentifier": "sample-cluster",
"DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
"PendingModifiedValues": {
    "DBInstanceClass": "db.r4.4xlarge"
},
"PubliclyAccessible": false,
"DBInstanceStatus": "available"
}
}

```

Weitere Informationen finden Sie unter [Ändern einer Amazon DocumentDB DocumentDB-Instance im Amazon DocumentDB DocumentDB-Entwicklerhandbuch](#).

- API-Einheiten finden Sie unter [ModifyDbInstance AWS CLI Befehlsreferenz](#).

modify-db-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `modify-db-subnet-group`.

AWS CLI

So ändern Sie eine Amazon DocumentDB-Subnetzgruppe

Im folgenden `modify-db-subnet-group` Beispiel wird die Subnetzgruppe geändert, `sample-subnet-group` indem die angegebenen Subnetze und eine neue Beschreibung hinzugefügt werden.

```
aws docdb modify-db-subnet-group \  
  --db-subnet-group-name sample-subnet-group \  
  --subnet-ids subnet-b3806e8f subnet-53ab3636 subnet-991cb8d0 \  
  --db-subnet-group-description "New subnet description"
```

Ausgabe:

```
{  
  "DBSubnetGroup": {  
    "DBSubnetGroupName": "sample-subnet-group",  
    "SubnetGroupStatus": "Complete",  
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-  
subnet-group",  
    "VpcId": "vpc-91280df6",  
    "DBSubnetGroupDescription": "New subnet description",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-b3806e8f",  
        "SubnetStatus": "Active",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      },  
      {  
        "SubnetIdentifier": "subnet-53ab3636",  
        "SubnetStatus": "Active",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2c"  
        }  
      }  
    ]  
  }  
}
```

```

        "SubnetIdentifizier": "subnet-991cb8d0",
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2b"
        }
    ]
}

```

Weitere Informationen finden Sie unter [Ändern einer Amazon DocumentDB-Subnetzgruppe im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [ModifyDbSubnetGroup](#).AWS CLI

reboot-db-instance

Das folgende Codebeispiel zeigt die Verwendung `reboot-db-instance`.

AWS CLI

Um eine Amazon DocumentDB DocumentDB-Instance neu zu starten

Im folgenden `reboot-db-instance` Beispiel wird die Amazon DocumentDB DocumentDB-Instance neu gestartet. `sample-cluster2`

```

aws docdb reboot-db-instance \
  --db-instance-identifizier sample-cluster2

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```

{
  "DBInstance": {
    "PreferredBackupWindow": "18:00-18:30",
    "DBInstanceIdentifizier": "sample-cluster2",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ],
    "DBSubnetGroup": {
      "VpcId": "vpc-91280df6",

```

```
    "Subnets": [
      {
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        },
        "SubnetIdentifier": "subnet-4e26d263"
      },
      {
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2c"
        },
        "SubnetIdentifier": "subnet-afc329f4"
      },
      {
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        },
        "SubnetIdentifier": "subnet-53ab3636"
      },
      {
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetIdentifier": "subnet-991cb8d0"
      }
    ],
    "SubnetGroupStatus": "Complete",
    "DBSubnetGroupName": "default",
    "DBSubnetGroupDescription": "default"
  },
  "PendingModifiedValues": {},
  "Endpoint": {
    "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "Port": 27017
  },
  "EnabledCloudwatchLogsExports": [
    "audit"
  ],
  "StorageEncrypted": false,
```

```
"DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
"AutoMinorVersionUpgrade": true,
"Engine": "docdb",
"InstanceCreateTime": "2019-03-15T20:36:06.338Z",
"EngineVersion": "3.6.0",
"PromotionTier": 5,
"BackupRetentionPeriod": 7,
"DBClusterIdentifier": "sample-cluster",
"PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
"PubliclyAccessible": false,
"DBInstanceClass": "db.r4.4xlarge",
"AvailabilityZone": "us-west-2d",
"DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
"DBInstanceStatus": "rebooting"
}
}
```

Weitere Informationen finden Sie unter Amazon DocumentDB [Rebooting an Amazon DocumentDB Instance](#) Developer Guide.

- API-Einheiten finden Sie unter [RebootDbInstance](#) Befehlsreferenz. AWS CLI

remove-tags-from-resource

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-resource`.

AWS CLI

So entfernen Sie Tags aus einer Amazon DocumentDB DocumentDB-Ressource

Im folgenden `remove-tags-from-resource` Beispiel wird das Tag mit dem Namen des Schlüssels `B` aus dem Amazon DocumentDB-Cluster `sample-cluster` entfernt.

```
aws docdb remove-tags-from-resource \
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
  --tag-keys B
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Entfernen von Tags aus einer Amazon DocumentDB Resource im Amazon DocumentDB Developerhandbuch](#).

- API-Einheiten finden Sie unter [RemoveTagsFromResource AWS CLI](#) Befehlsreferenz.

reset-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `reset-db-cluster-parameter-group`.

AWS CLI

Um den angegebenen Parameterwert in einer Amazon DocumentDB DocumentDB-Parametergruppe auf die Standardwerte zurückzusetzen

Im folgenden `reset-db-cluster-parameter-group` Beispiel wird der Parameter `ttl_monitor` in der Amazon DocumentDB DocumentDB-Parametergruppe `custom3-6-param-grp` auf seinen Standardwert zurückgesetzt.

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters ParameterName=ttl_monitor,ApplyMethod=immediate
```

Ausgabe:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Weitere Informationen finden Sie im Titel im Amazon DocumentDB Developer Guide.

Um bestimmte oder alle Parameterwerte in einer Amazon DocumentDB DocumentDB-Parametergruppe auf ihre Standardwerte zurückzusetzen

Im folgenden `reset-db-cluster-parameter-group` Beispiel werden alle Parameter in der Amazon DocumentDB DocumentDB-Parametergruppe `custom3-6-param-grp` auf ihren Standardwert zurückgesetzt.

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --reset-all-parameters
```

Ausgabe:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"
```

```
}

```

Weitere Informationen finden Sie unter [Zurücksetzen einer Amazon DocumentDB-Cluster-Parametergruppe im Amazon DocumentDB Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz [ResetDbClusterParameterGroup](#).AWS CLI

restore-db-cluster-from-snapshot

Das folgende Codebeispiel zeigt die Verwendung `restore-db-cluster-from-snapshot`.

AWS CLI

So stellen Sie einen Amazon DocumentDB-Cluster aus einem automatischen oder manuellen Snapshot wieder her

Das folgende `restore-db-cluster-from-snapshot` Beispiel erstellt einen neuen Amazon DocumentDB-Cluster, der nach `sample-cluster-2019-03-16-00-01-restored` dem Snapshot `rds:sample-cluster-2019-03-16-00-01` benannt wird.

```
aws docdb restore-db-cluster-from-snapshot \
  --db-cluster-identifizier sample-cluster-2019-03-16-00-01-restored \
  --engine docdb \
  --snapshot-identifizier rds:sample-cluster-2019-03-16-00-01
```

Ausgabe:

```
{
  "DBCluster": {
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
    "HostedZoneId": "ZKXHX85TT8WVW",
    "Engine": "docdb",
    "DBClusterMembers": [],
    "MultiAZ": false,
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2c",
      "us-west-2b"
    ],
    "StorageEncrypted": false,
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
```

```

    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
    "Port": 27017,
    "PreferredBackupWindow": "00:00-00:30",
    "DBSubnetGroup": "default",
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
    "MasterUsername": "master-user",
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Wiederherstellen aus einem Cluster-Snapshot](#) im Amazon DocumentDB Developer Guide.

- API-Einheiten finden Sie [RestoreDbClusterFromSnapshot](#) in der AWS CLI Befehlsreferenz.

restore-db-cluster-to-point-in-time

Das folgende Codebeispiel zeigt die Verwendung `restore-db-cluster-to-point-in-time`.

AWS CLI

So stellen Sie einen Amazon DocumentDB-Cluster point-in-time aus einem manuellen Snapshot wieder her

Im folgenden `restore-db-cluster-to-point-in-time` Beispiel wird der verwendete `sample-cluster-snapshot`, um einen neuen Amazon DocumentDB-Cluster zu erstellen `sample-cluster-pit`, wobei der letzte wiederherstellbare Zeitpunkt verwendet wird.


```
aws docdb restore-db-cluster-to-point-in-time \
  --db-cluster-identifier sample-cluster-pit \
  --source-db-cluster-identifier arn:aws:rds:us-
west-2:123456789012:cluster:sample-cluster \
  --use-latest-restorable-time
```

Ausgabe:

```
{
  "DBCluster": {
    "StorageEncrypted": false,
    "BackupRetentionPeriod": 3,
    "MasterUsername": "master-user",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "PreferredBackupWindow": "00:00-00:30",
    "MultiAZ": false,
    "DBClusterIdentifier": "sample-cluster-pit",
    "DBSubnetGroup": "default",
    "ClusterCreateTime": "2019-04-03T15:55:21.320Z",
    "AssociatedRoles": [],
    "DBClusterParameterGroup": "default.docdb3.6",
    "DBClusterMembers": [],
    "Status": "creating",
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2d",
      "us-west-2b"
    ],
    "ReaderEndpoint": "sample-cluster-pit.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "Port": 27017,
    "Engine": "docdb",
    "EngineVersion": "3.6.0",
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ],
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "Endpoint": "sample-cluster-pit.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DbClusterResourceId": "cluster-NLCABBX0SE2QPQ4GOLZIFWEPLM",
```

```
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
    pit"
  }
}
```

Weitere Informationen finden Sie unter [Restoring a Snapshot to a Point in Time im Amazon DocumentDB Developer Guide](#).

- API-Einheiten finden Sie unter [RestoreDbClusterToPointInTime AWS CLI Befehlsreferenz](#).

start-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `start-db-cluster`.

AWS CLI

Um einen gestoppten Amazon DocumentDB-Cluster zu starten

Im folgenden `start-db-cluster` Beispiel wird der angegebene Amazon DocumentDB-Cluster gestartet.

```
aws docdb start-db-cluster \
  --db-cluster-identifier sample-cluster
```

Ausgabe:

```
{
  "DBCluster": {
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "Engine": "docdb",
    "DBClusterMembers": [],
    "MultiAZ": false,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1c",
      "us-east-1f"
    ],
    "StorageEncrypted": false,
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
    corcjozrlsfc.us-east-1.docdb.amazonaws.com",
```

```

    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
    "Port": 27017,
    "PreferredBackupWindow": "00:00-00:30",
    "DBSubnetGroup": "default",
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
    "MasterUsername": "master-user",
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Stoppen und Starten eines Amazon DocumentDB-Clusters im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [StartDbCluster AWS CLI](#) Befehlsreferenz.

stop-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `stop-db-cluster`.

AWS CLI

Um einen laufenden Amazon DocumentDB-Cluster zu beenden

Das folgende `stop-db-cluster` Beispiel stoppt den angegebenen Amazon DocumentDB-Cluster.

```
aws docdb stop-db-cluster \
```

```
--db-cluster-identifizier sample-cluster
```

Ausgabe:

```
{
  "DBCluster": {
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "Engine": "docdb",
    "DBClusterMembers": [],
    "MultiAZ": false,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1c",
      "us-east-1f"
    ],
    "StorageEncrypted": false,
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",
    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",
    "Port": 27017,
    "PreferredBackupWindow": "00:00-00:30",
    "DBSubnetGroup": "default",
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
    "MasterUsername": "master-user",
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Stoppen und Starten eines Amazon DocumentDB-Clusters im Amazon DocumentDB](#) DocumentDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [StopDbCluster AWS CLI](#) Befehlsreferenz.

DynamoDB-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit DynamoDB Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-get-item

Das folgende Codebeispiel zeigt die Verwendung `batch-get-item`.

AWS CLI

Um mehrere Elemente aus einer Tabelle abzurufen

Im folgenden `batch-get-items` Beispiel werden mithilfe eines Stapels von drei `GetItem` Anfragen mehrere Elemente aus der `MusicCollection` Tabelle gelesen und die Anzahl der durch den Vorgang verbrauchten Lesekapazitätseinheiten abgefragt. Der Befehl gibt nur das `AlbumTitle` Attribut zurück.

```
aws dynamodb batch-get-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity TOTAL
```

Inhalt von `request-items.json`:

```
{
  "MusicCollection": {
    "Keys": [
      {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"}
      },
      {
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"}
      },
      {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Scared of My Shadow"}
      }
    ],
    "ProjectionExpression": "AlbumTitle"
  }
}
```

Ausgabe:

```
{
  "Responses": {
    "MusicCollection": [
      {
        "AlbumTitle": {
          "S": "Somewhat Famous"
        }
      },
      {
        "AlbumTitle": {
          "S": "Blue Sky Blues"
        }
      },
      {
        "AlbumTitle": {
          "S": "Louder Than Ever"
        }
      }
    ]
  },
  "UnprocessedKeys": {},
}
```

```
"ConsumedCapacity": [  
  {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 1.5  
  }  
]  
}
```

Weitere Informationen finden Sie unter [Batch Operations](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [BatchGetItem AWS CLI](#) Befehlsreferenz.

batch-write-item

Das folgende Codebeispiel zeigt die Verwendung `batch-write-item`.

AWS CLI

Um mehrere Elemente zu einer Tabelle hinzuzufügen

Im folgenden `batch-write-item` Beispiel werden der `MusicCollection` Tabelle drei neue Elemente hinzugefügt, wobei ein Stapel von drei `PutItem` Anfragen verwendet wird. Außerdem werden Informationen über die Anzahl der durch den Vorgang verbrauchten Schreibkapazitätseinheiten sowie über alle durch den Vorgang geänderten Elementsammlungen angefordert.

```
aws dynamodb batch-write-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity INDEXES \  
  --return-item-collection-metrics SIZE
```

Inhalt von `request-items.json`:

```
{  
  "MusicCollection": [  
    {  
      "PutRequest": {  
        "Item": {  
          "Artist": {"S": "No One You Know"},  
          "SongTitle": {"S": "Call Me Today"},  
          "AlbumTitle": {"S": "Somewhat Famous"}  
        }  
      }  
    }  
  ]  
}
```

```

    }
  },
  {
    "PutRequest": {
      "Item": {
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"},
        "AlbumTitle": {"S": "Songs About Life"}
      }
    }
  },
  {
    "PutRequest": {
      "Item": {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Scared of My Shadow"},
        "AlbumTitle": {"S": "Blue Sky Blues"}
      }
    }
  }
]
}

```

Ausgabe:

```

{
  "UnprocessedItems": {},
  "ItemCollectionMetrics": {
    "MusicCollection": [
      {
        "ItemCollectionKey": {
          "Artist": {
            "S": "No One You Know"
          }
        },
        "SizeEstimateRangeGB": [
          0.0,
          1.0
        ]
      },
      {
        "ItemCollectionKey": {

```



```

        "Artist": {
            "S": "Acme Band"
        }
    },
    "SizeEstimateRangeGB": [
        0.0,
        1.0
    ]
}
]
},
"ConsumedCapacity": [
    {
        "TableName": "MusicCollection",
        "CapacityUnits": 6.0,
        "Table": {
            "CapacityUnits": 3.0
        },
        "LocalSecondaryIndexes": {
            "AlbumTitleIndex": {
                "CapacityUnits": 3.0
            }
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Batch Operations](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [BatchWriteItem AWS CLI Befehlsreferenz](#).

create-backup

Das folgende Codebeispiel zeigt die Verwendung `create-backup`.

AWS CLI

Um ein Backup für eine bestehende DynamoDB-Tabelle zu erstellen

Im folgenden `create-backup` Beispiel wird eine Sicherungskopie der `MusicCollection` Tabelle erstellt.

```
aws dynamodb create-backup \  
  --table-name MusicCollection \  
  --backup-name MusicCollectionBackup
```

Ausgabe:

```
{  
  "BackupDetails": {  
    "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
backup/01576616366715-b4e58d3a",  
    "BackupName": "MusicCollectionBackup",  
    "BackupSizeBytes": 0,  
    "BackupStatus": "CREATING",  
    "BackupType": "USER",  
    "BackupCreationDateTime": 1576616366.715  
  }  
}
```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [CreateBackup](#) Befehlsreferenz.AWS CLI

create-global-table

Das folgende Codebeispiel zeigt die Verwendung `create-global-table`.

AWS CLI

Um eine globale Tabelle zu erstellen

Im folgenden `create-global-table` Beispiel wird eine globale Tabelle aus zwei identischen Tabellen in den angegebenen, separaten AWS Regionen erstellt.

```
aws dynamodb create-global-table \  
  --global-table-name MusicCollection \  
  --replication-group RegionName=us-east-2 RegionName=us-east-1 \  
  --region us-east-2
```

Ausgabe:

```
{
  "GlobalTableDescription": {
    "ReplicationGroup": [
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-east-1"
      }
    ],
    "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
    "CreationDateTime": 1576625818.532,
    "GlobalTableStatus": "CREATING",
    "GlobalTableName": "MusicCollection"
  }
}
```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [CreateGlobalTable](#) Befehlsreferenz. AWS CLI

create-table

Das folgende Codebeispiel zeigt die Verwendung `create-table`.

AWS CLI

Beispiel 1: Um eine Tabelle mit Tags zu erstellen

Im folgenden `create-table` Beispiel werden die angegebenen Attribute und das angegebene Schlüsselschema verwendet, um eine Tabelle mit dem Namen `MusicCollection` zu erstellen. Diese Tabelle verwendet den bereitgestellten Durchsatz und wird im Ruhezustand mit der Standardeinstellung AWS Owned CMK verschlüsselt. Der Befehl weist der Tabelle außerdem ein Tag mit dem Schlüssel `Owner` und dem Wert von `blueTeam` zu.

```
aws dynamodb create-table \
  --table-name MusicCollection \
  --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
\
```

```

--key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
--provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
--tags Key=Owner,Value=blueTeam

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "WriteCapacityUnits": 5,
      "ReadCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "TableName": "MusicCollection",
    "TableStatus": "CREATING",
    "KeySchema": [
      {
        "KeyType": "HASH",
        "AttributeName": "Artist"
      },
      {
        "KeyType": "RANGE",
        "AttributeName": "SongTitle"
      }
    ],
    "ItemCount": 0,
    "CreationDateTime": "2020-05-26T16:04:41.627000-07:00",
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 2: So erstellen Sie eine Tabelle im On-Demand-Modus

Im folgenden Beispiel wird eine Tabelle erstellt, die im MusicCollection On-Demand-Modus und nicht im Bereitstellungs-Durchsatzmodus aufgerufen wird. Dies ist nützlich für Tabellen mit unvorhersehbaren Workloads.

```
aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-  
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S  
 \  
  --key-  
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \  
  --billing-mode PAY_PER_REQUEST
```

Ausgabe:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
  },  
}
```

```

    "TableStatus": "CREATING",
    "CreationDateTime": "2020-05-27T11:44:10.807000-07:00",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 0,
      "WriteCapacityUnits": 0
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "BillingModeSummary": {
      "BillingMode": "PAY_PER_REQUEST"
    }
  }
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 3: So erstellen Sie eine Tabelle und verschlüsseln sie mit einem vom Kunden verwalteten CMK

Das folgende Beispiel erstellt eine Tabelle mit dem Namen `MusicCollection` und verschlüsselt sie mithilfe einer vom Kunden verwalteten Tabelle. CMK

```

aws dynamodb create-table \
  --table-name MusicCollection \
  --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
\
  --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
  --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=abcd1234-abcd-1234-
a123-ab1234a1b234

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {

```

```

        "AttributeName": "Artist",
        "AttributeType": "S"
    },
    {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
    }
],
"TableName": "MusicCollection",
"KeySchema": [
    {
        "AttributeName": "Artist",
        "KeyType": "HASH"
    },
    {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
    }
],
"TableStatus": "CREATING",
"CreationDateTime": "2020-05-27T11:12:16.431000-07:00",
"ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
"TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"SSEDescription": {
    "Status": "ENABLED",
    "SSEType": "KMS",
    "KMSMasterKeyArn": "arn:aws:kms:us-west-2:123456789012:key/abcd1234-
abcd-1234-a123-ab1234a1b234"
}
}
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 4: So erstellen Sie eine Tabelle mit einem lokalen sekundären Index

Im folgenden Beispiel werden die angegebenen Attribute und das angegebene Schlüsselschema verwendet, um eine Tabelle `MusicCollection` mit einem Namen für den lokalen sekundären Index zu erstellen `AlbumTitleIndex`.

```
aws dynamodb create-table \
  --table-name MusicCollection \
  --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S Att
\
  --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --local-secondary-indexes \
    "[
      {
        \"IndexName\": \"AlbumTitleIndex\",
        \"KeySchema\": [
          {\"AttributeName\": \"Artist\", \"KeyType\": \"HASH\"},
          {\"AttributeName\": \"AlbumTitle\", \"KeyType\": \"RANGE\"}
        ],
        \"Projection\": {
          \"ProjectionType\": \"INCLUDE\",
          \"NonKeyAttributes\": [\"Genre\", \"Year\"]
        }
      }
    ]"
```

Ausgabe:

```
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "AlbumTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ]
  }
}
```



```
    }
  ],
  "TableName": "MusicCollection",
  "KeySchema": [
    {
      "AttributeName": "Artist",
      "KeyType": "HASH"
    },
    {
      "AttributeName": "SongTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "CREATING",
  "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
  "ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
  },
  "TableSizeBytes": 0,
  "ItemCount": 0,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
  "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "LocalSecondaryIndexes": [
    {
      "IndexName": "AlbumTitleIndex",
      "KeySchema": [
        {
          "AttributeName": "Artist",
          "KeyType": "HASH"
        },
        {
          "AttributeName": "AlbumTitle",
          "KeyType": "RANGE"
        }
      ],
      "Projection": {
        "ProjectionType": "INCLUDE",
        "NonKeyAttributes": [
          "Genre",
          "Year"
        ]
      }
    }
  ],
}
```

```

        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
    }
]
}
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 5: So erstellen Sie eine Tabelle mit einem globalen sekundären Index

Im folgenden Beispiel wird eine Tabelle GameScores mit dem Namen „Globaler Sekundärindex“ erstellt GameTitleIndex. Die Basistabelle hat einen Partitionsschlüssel von UserId und einen Sortierschlüssel von GameTitle, sodass Sie effizient die beste Punktzahl eines einzelnen Benutzers für ein bestimmtes Spiel ermitteln können, wohingegen die Tabelle einen Partitionsschlüssel von GameTitle und einen Sortierschlüssel von GSI hat TopScore, sodass Sie schnell die höchste Gesamtpunktzahl für ein bestimmtes Spiel ermitteln können.

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
  \
  --key-schema AttributeName=UserId,KeyType=HASH \
                AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --global-secondary-indexes \
    "[
      {
        \"IndexName\": \"GameTitleIndex\",
        \"KeySchema\": [
          {\"AttributeName\": \"GameTitle\", \"KeyType\": \"HASH\"},
          {\"AttributeName\": \"TopScore\", \"KeyType\": \"RANGE\"}
        ],
        \"Projection\": {
          \"ProjectionType\": \"INCLUDE\",
          \"NonKeyAttributes\": [\"UserId\"]
        },
        \"ProvisionedThroughput\": {

```

```

        \ "ReadCapacityUnits\": 10,
        \ "WriteCapacityUnits\": 5
    }
}
]"

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "GameTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "TopScore",
        "AttributeType": "N"
      },
      {
        "AttributeName": "UserId",
        "AttributeType": "S"
      }
    ],
    "TableName": "GameScores",
    "KeySchema": [
      {
        "AttributeName": "UserId",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "GameTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2020-05-26T17:28:15.602000-07:00",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,

```

```

    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "GlobalSecondaryIndexes": [
      {
        "IndexName": "GameTitleIndex",
        "KeySchema": [
          {
            "AttributeName": "GameTitle",
            "KeyType": "HASH"
          },
          {
            "AttributeName": "TopScore",
            "KeyType": "RANGE"
          }
        ],
        "Projection": {
          "ProjectionType": "INCLUDE",
          "NonKeyAttributes": [
            "UserId"
          ]
        },
        "IndexStatus": "CREATING",
        "ProvisionedThroughput": {
          "NumberOfDecreasesToday": 0,
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 5
        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 6: So erstellen Sie eine Tabelle mit mehreren globalen Sekundärindizes gleichzeitig

Im folgenden Beispiel wird eine Tabelle erstellt, die GameScores mit zwei globalen sekundären Indizes benannt ist. Die GSI Schemas werden über eine Datei und nicht über die Befehlszeile übergeben.

```
aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
\
  --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --global-secondary-indexes file://gsi.json
```

Inhalt von `gsi.json`:

```
[
  {
    "IndexName": "GameTitleIndex",
    "KeySchema": [
      {
        "AttributeName": "GameTitle",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "TopScore",
        "KeyType": "RANGE"
      }
    ],
    "Projection": {
      "ProjectionType": "ALL"
    },
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    }
  },
  {
    "IndexName": "GameDateIndex",
    "KeySchema": [
      {
        "AttributeName": "GameTitle",
        "KeyType": "HASH"
      }
    ]
  }
]
```

```
    },
    {
      "AttributeName": "Date",
      "KeyType": "RANGE"
    }
  ],
  "Projection": {
    "ProjectionType": "ALL"
  },
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  }
}
```

Ausgabe:

```
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Date",
        "AttributeType": "S"
      },
      {
        "AttributeName": "GameTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "TopScore",
        "AttributeType": "N"
      },
      {
        "AttributeName": "UserId",
        "AttributeType": "S"
      }
    ],
    "TableName": "GameScores",
    "KeySchema": [
      {
        "AttributeName": "UserId",
        "KeyType": "HASH"
      }
    ]
  }
}
```

```
    },
    {
      "AttributeName": "GameTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "CREATING",
  "CreationDateTime": "2020-08-04T16:40:55.524000-07:00",
  "ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
  },
  "TableSizeBytes": 0,
  "ItemCount": 0,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
  "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "GlobalSecondaryIndexes": [
    {
      "IndexName": "GameTitleIndex",
      "KeySchema": [
        {
          "AttributeName": "GameTitle",
          "KeyType": "HASH"
        },
        {
          "AttributeName": "TopScore",
          "KeyType": "RANGE"
        }
      ],
      "Projection": {
        "ProjectionType": "ALL"
      },
      "IndexStatus": "CREATING",
      "ProvisionedThroughput": {
        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 10,
        "WriteCapacityUnits": 5
      },
      "IndexSizeBytes": 0,
      "ItemCount": 0,
      "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
    },
  ],
```

```

    {
      "IndexName": "GameDateIndex",
      "KeySchema": [
        {
          "AttributeName": "GameTitle",
          "KeyType": "HASH"
        },
        {
          "AttributeName": "Date",
          "KeyType": "RANGE"
        }
      ],
      "Projection": {
        "ProjectionType": "ALL"
      },
      "IndexStatus": "CREATING",
      "ProvisionedThroughput": {
        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
      },
      "IndexSizeBytes": 0,
      "ItemCount": 0,
      "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameDateIndex"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 7: So erstellen Sie eine Tabelle mit aktivierten Streams

Im folgenden Beispiel wird eine Tabelle `GameScores` mit aktiviertem DynamoDB Streams aufgerufen. Sowohl neue als auch alte Bilder jedes Elements werden in den Stream geschrieben.

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
\

```



```

--key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
--provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
--stream-specification StreamEnabled=TRUE,StreamViewType=NEW_AND_OLD_IMAGES

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "GameTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "UserId",
        "AttributeType": "S"
      }
    ],
    "TableName": "GameScores",
    "KeySchema": [
      {
        "AttributeName": "UserId",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "GameTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2020-05-27T10:49:34.056000-07:00",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "StreamSpecification": {
      "StreamEnabled": true,

```

```

        "StreamViewType": "NEW_AND_OLD_IMAGES"
    },
    "LatestStreamLabel": "2020-05-27T17:49:34.056",
    "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2020-05-27T17:49:34.056"
}
}

```

Weitere Informationen finden Sie unter [Basic Operations for Tables](#) im Amazon DynamoDB Developer Guide.

Beispiel 8: So erstellen Sie eine Tabelle mit aktiviertem Keys-Only-Stream

Im folgenden Beispiel wird eine Tabelle GameScores mit aktiviertem DynamoDB Streams aufgerufen. Nur die Schlüsselattribute der geänderten Elemente werden in den Stream geschrieben.

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
\
  --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --stream-specification StreamEnabled=TRUE,StreamViewType=KEYS_ONLY

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "GameTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "UserId",
        "AttributeType": "S"
      }
    ],
    "TableName": "GameScores",
    "KeySchema": [

```

```

    {
      "AttributeName": "UserId",
      "KeyType": "HASH"
    },
    {
      "AttributeName": "GameTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "CREATING",
  "CreationDateTime": "2023-05-25T18:45:34.140000+00:00",
  "ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
  },
  "TableSizeBytes": 0,
  "ItemCount": 0,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
  "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "StreamSpecification": {
    "StreamEnabled": true,
    "StreamViewType": "KEYS_ONLY"
  },
  "LatestStreamLabel": "2023-05-25T18:45:34.140",
  "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2023-05-25T18:45:34.140",
  "DeletionProtectionEnabled": false
}
}

```

Weitere Informationen finden Sie unter [Change Data Capture for DynamoDB Streams](#) im Amazon DynamoDB Developer Guide.

Beispiel 9: So erstellen Sie eine Tabelle mit der Klasse Standard Infrequent Access

Im folgenden Beispiel wird eine Tabelle mit dem Namen Standard-Infrequent Access (DynamoDB Standard-IA) erstellt GameScores und ihr zugewiesen. Diese Tabellenklasse ist für Speicher optimiert, da der Hauptkostenfaktor ist.

```

aws dynamodb create-table \
  --table-name GameScores \

```

```

--attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
\
--key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
--provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
--table-class STANDARD_INFREQUENT_ACCESS

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "GameTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "UserId",
        "AttributeType": "S"
      }
    ],
    "TableName": "GameScores",
    "KeySchema": [
      {
        "AttributeName": "UserId",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "GameTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2023-05-25T18:33:07.581000+00:00",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",

```

```

    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "TableClassSummary": {
      "TableClass": "STANDARD_INFREQUENT_ACCESS"
    },
    "DeletionProtectionEnabled": false
  }
}

```

Weitere Informationen finden Sie unter [Tabellenklassen](#) im Amazon DynamoDB Developer Guide.

Beispiel 10: So erstellen Sie eine Tabelle mit aktiviertem Löschschutz

Das folgende Beispiel erstellt eine Tabelle mit dem Namen GameScores und aktiviert den Löschschutz.

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
\
  --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --deletion-protection-enabled

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "GameTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "UserId",
        "AttributeType": "S"
      }
    ],
    "TableName": "GameScores",
    "KeySchema": [
      {
        "AttributeName": "UserId",

```

```

        "KeyType": "HASH"
    },
    {
        "AttributeName": "GameTitle",
        "KeyType": "RANGE"
    }
],
"TableStatus": "CREATING",
"CreationDateTime": "2023-05-25T23:02:17.093000+00:00",
"ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
"TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"DeletionProtectionEnabled": true
}
}

```

Weitere Informationen finden Sie unter [Verwenden des Löschschutzes](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [CreateTable AWS CLI Befehlsreferenz](#).

delete-backup

Das folgende Codebeispiel zeigt die Verwendung `delete-backup`.

AWS CLI

Um ein vorhandenes DynamoDB-Backup zu löschen

Im folgenden `delete-backup` Beispiel wird die angegebene vorhandene Sicherung gelöscht.

```

aws dynamodb delete-backup \
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a

```

Ausgabe:

```

{
  "BackupDescription": {
    "BackupDetails": {
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
      "BackupName": "MusicCollectionBackup",
      "BackupSizeBytes": 0,
      "BackupStatus": "DELETED",
      "BackupType": "USER",
      "BackupCreationDateTime": 1576616366.715
    },
    "SourceTableDetails": {
      "TableName": "MusicCollection",
      "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "TableSizeBytes": 0,
      "KeySchema": [
        {
          "AttributeName": "Artist",
          "KeyType": "HASH"
        },
        {
          "AttributeName": "SongTitle",
          "KeyType": "RANGE"
        }
      ],
      "TableCreationDateTime": 1576615228.571,
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
      },
      "ItemCount": 0,
      "BillingMode": "PROVISIONED"
    },
    "SourceTableFeatureDetails": {}
  }
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DeleteBackup](#) Befehlsreferenz.AWS CLI

delete-item

Das folgende Codebeispiel zeigt die Verwendung `delete-item`.

AWS CLI

Beispiel 1: Um ein Element zu löschen

Das folgende `delete-item` Beispiel löscht ein Element aus der `MusicCollection` Tabelle und fordert Details zu dem gelöschten Element und der von der Anforderung verwendeten Kapazität an.

```
aws dynamodb delete-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-values ALL_OLD \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

Inhalt von `key.json`:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Scared of My Shadow"}  
}
```

Ausgabe:

```
{  
  "Attributes": {  
    "AlbumTitle": {  
      "S": "Blue Sky Blues"  
    },  
    "Artist": {  
      "S": "No One You Know"  
    },  
    "SongTitle": {  
      "S": "Scared of My Shadow"  
    }  
  },  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 2.0  
  }  
}
```



```

    },
    "ItemCollectionMetrics": {
      "ItemCollectionKey": {
        "Artist": {
          "S": "No One You Know"
        }
      },
    },
    "SizeEstimateRangeGB": [
      0.0,
      1.0
    ]
  }
}

```

Weitere Informationen finden Sie unter [Artikel schreiben](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 2: Um einen Artikel unter bestimmten Bedingungen zu löschen

Im folgenden Beispiel wird ein Artikel nur dann aus der ProductCatalog Tabelle gelöscht, wenn er entweder Sporting Goods oder ProductCategory ist Gardening Supplies und sein Preis zwischen 500 und 600 liegt. Es gibt Details zu dem Element zurück, das gelöscht wurde.

```

aws dynamodb delete-item \
  --table-name ProductCatalog \
  --key '{"Id":{"N":"456"}}' \
  --condition-expression "(ProductCategory IN (:cat1, :cat2)) and (#P between :lo and :hi)" \
  --expression-attribute-names file://names.json \
  --expression-attribute-values file://values.json \
  --return-values ALL_OLD

```

Inhalt von names.json:

```

{
  "#P": "Price"
}

```

Inhalt von values.json:

```

{
  ":cat1": {"S": "Sporting Goods"},

```

```
    ":cat2": {"S": "Gardening Supplies"},  
    ":lo": {"N": "500"},  
    ":hi": {"N": "600"}  
}
```

Ausgabe:

```
{  
  "Attributes": {  
    "Id": {  
      "N": "456"  
    },  
    "Price": {  
      "N": "550"  
    },  
    "ProductCategory": {  
      "S": "Sporting Goods"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Artikel schreiben](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DeleteItem AWS CLI](#) Befehlsreferenz.

delete-table

Das folgende Codebeispiel zeigt die Verwendung `delete-table`.

AWS CLI

Um eine Tabelle zu löschen

Im folgenden `delete-table` Beispiel wird die `MusicCollection` Tabelle gelöscht.

```
aws dynamodb delete-table \  
  --table-name MusicCollection
```

Ausgabe:

```
{
```

```

    "TableDescription": {
      "TableStatus": "DELETING",
      "TableSizeBytes": 0,
      "ItemCount": 0,
      "TableName": "MusicCollection",
      "ProvisionedThroughput": {
        "NumberOfDecreasesToday": 0,
        "WriteCapacityUnits": 5,
        "ReadCapacityUnits": 5
      }
    }
  }
}

```

Weitere Informationen finden Sie unter [Löschen einer Tabelle](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DeleteTable AWS CLI Befehlsreferenz](#).

describe-backup

Das folgende Codebeispiel zeigt die Verwendung `describe-backup`.

AWS CLI

Um Informationen über eine bestehende Sicherung einer Tabelle abzurufen

Im folgenden `describe-backup` Beispiel werden Informationen über die angegebene vorhandene Sicherung angezeigt.

```

aws dynamodb describe-backup \
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a

```

Ausgabe:

```

{
  "BackupDescription": {
    "BackupDetails": {
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
      "BackupName": "MusicCollectionBackup",
      "BackupSizeBytes": 0,
      "BackupStatus": "AVAILABLE",

```

```

    "BackupType": "USER",
    "BackupCreationDateTime": 1576616366.715
  },
  "SourceTableDetails": {
    "TableName": "MusicCollection",
    "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
    "TableSizeBytes": 0,
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableCreationDateTime": 1576615228.571,
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "ItemCount": 0,
    "BillingMode": "PROVISIONED"
  },
  "SourceTableFeatureDetails": {}
}
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeBackup](#) Befehlsreferenz.AWS CLI

describe-continuous-backups

Das folgende Codebeispiel zeigt die Verwendung `describe-continuous-backups`.

AWS CLI

Um Informationen über kontinuierliche Backups für eine DynamoDB-Tabelle abzurufen

Im folgenden `describe-continuous-backups` Beispiel werden Details zu den Einstellungen für kontinuierliche Backups für die `MusicCollection` Tabelle angezeigt.

```
aws dynamodb describe-continuous-backups \  
  --table-name MusicCollection
```

Ausgabe:

```
{  
  "ContinuousBackupsDescription": {  
    "ContinuousBackupsStatus": "ENABLED",  
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "DISABLED"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Point-in-Time Recovery for DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeContinuousBackups](#) Befehlsreferenz. AWS CLI

describe-contributor-insights

Das folgende Codebeispiel zeigt die Verwendung `describe-contributor-insights`.

AWS CLI

So zeigen Sie Contributor Insights-Einstellungen für eine DynamoDB-Tabelle an

Im folgenden `describe-contributor-insights` Beispiel werden die Contributor Insights-Einstellungen für die `MusicCollection` Tabelle und den `AlbumTitle-index` globalen Sekundärindex angezeigt.

```
aws dynamodb describe-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index
```

Ausgabe:

```
{  
  "TableName": "MusicCollection",
```

```
"IndexName": "AlbumTitle-index",
"ContributorInsightsRuleList": [
  "DynamoDBContributorInsights-PKC-MusicCollection-1576629651520",
  "DynamoDBContributorInsights-SKC-MusicCollection-1576629651520",
  "DynamoDBContributorInsights-PKT-MusicCollection-1576629651520",
  "DynamoDBContributorInsights-SKT-MusicCollection-1576629651520"
],
"ContributorInsightsStatus": "ENABLED",
"LastUpdateDateTime": 1576629654.78
}
```

Weitere Informationen finden Sie unter [Analysieren des Datenzugriffs mithilfe von CloudWatch Contributor Insights for DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz. [DescribeContributorInsights](#) AWS CLI

describe-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-endpoints`.

AWS CLI

Um regionale Endpunktinformationen anzuzeigen

Im folgenden `describe-endpoints` Beispiel werden Details zu den Endpunkten für die aktuelle AWS Region angezeigt.

```
aws dynamodb describe-endpoints
```

Ausgabe:

```
{
  "Endpoints": [
    {
      "Address": "dynamodb.us-west-2.amazonaws.com",
      "CachePeriodInMinutes": 1440
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon DynamoDB Endpoints and Quotas](#) in der AWS Allgemeinen Referenz.

- API-Einheiten finden Sie unter Befehlsreferenz [DescribeEndpoints](#).AWS CLI

describe-global-table-settings

Das folgende Codebeispiel zeigt die Verwendung `describe-global-table-settings`.

AWS CLI

Um Informationen über die Einstellungen einer globalen DynamoDB-Tabelle abzurufen

Im folgenden `describe-global-table-settings` Beispiel werden die Einstellungen für die `MusicCollection` globale Tabelle angezeigt.

```
aws dynamodb describe-global-table-settings \  
  --global-table-name MusicCollection
```

Ausgabe:

```
{  
  "GlobalTableName": "MusicCollection",  
  "ReplicaSettings": [  
    {  
      "RegionName": "us-east-1",  
      "ReplicaStatus": "ACTIVE",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 5,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    },  
    {  
      "RegionName": "us-east-2",  
      "ReplicaStatus": "ACTIVE",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 5,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
```

```
        "AutoScalingDisabled": true
      }
    ]
  }
```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeGlobalTableSettings](#) Befehlsreferenz.AWS CLI

describe-global-table

Das folgende Codebeispiel zeigt die Verwendung `describe-global-table`.

AWS CLI

So zeigen Sie Informationen zu einer globalen DynamoDB-Tabelle an

Im folgenden `describe-global-table` Beispiel werden Details zur `MusicCollection` globalen Tabelle angezeigt.

```
aws dynamodb describe-global-table \
  --global-table-name MusicCollection
```

Ausgabe:

```
{
  "GlobalTableDescription": {
    "ReplicationGroup": [
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-east-1"
      }
    ],
    "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
    "CreationDateTime": 1576625818.532,
    "GlobalTableStatus": "ACTIVE",
    "GlobalTableName": "MusicCollection"
  }
}
```



```
}  
}
```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeGlobalTable](#) Befehlsreferenz.AWS CLI

describe-limits

Das folgende Codebeispiel zeigt die Verwendung `describe-limits`.

AWS CLI

Um die Limits für die bereitgestellte Kapazität anzuzeigen

Im folgenden `describe-limits` Beispiel werden die Limits der bereitgestellten Kapazität für Ihr Konto in der aktuellen Region angezeigt. AWS

```
aws dynamodb describe-limits
```

Ausgabe:

```
{  
  "AccountMaxReadCapacityUnits": 80000,  
  "AccountMaxWriteCapacityUnits": 80000,  
  "TableMaxReadCapacityUnits": 40000,  
  "TableMaxWriteCapacityUnits": 40000  
}
```

Weitere Informationen finden Sie unter [Limits in DynamoDB im](#) Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeLimits](#) Befehlsreferenz.AWS CLI

describe-table-replica-auto-scaling

Das folgende Codebeispiel zeigt die Verwendung `describe-table-replica-auto-scaling`.

AWS CLI

So zeigen Sie Auto-Scaling-Einstellungen für Replikate einer globalen Tabelle an

Im folgenden `describe-table-replica-auto-scaling` Beispiel werden Auto-Scaling-Einstellungen für alle Replikate der `MusicCollection` globalen Tabelle angezeigt.

```
aws dynamodb describe-table-replica-auto-scaling \  
--table-name MusicCollection
```

Ausgabe:

```
{  
  "TableAutoScalingDescription": {  
    "TableName": "MusicCollection",  
    "TableStatus": "ACTIVE",  
    "Replicas": [  
      {  
        "RegionName": "us-east-1",  
        "GlobalSecondaryIndexes": [],  
        "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
          "MinimumUnits": 5,  
          "MaximumUnits": 40000,  
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/  
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",  
          "ScalingPolicies": [  
            {  
              "PolicyName": "DynamoDBReadCapacityUtilization:table/  
MusicCollection",  
              "TargetTrackingScalingPolicyConfiguration": {  
                "TargetValue": 70.0  
              }  
            }  
          ]  
        },  
        "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
          "MinimumUnits": 5,  
          "MaximumUnits": 40000,  
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/  
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",  
          "ScalingPolicies": [  
            {  
              "PolicyName": "DynamoDBWriteCapacityUtilization:table/  
MusicCollection",  
              "TargetTrackingScalingPolicyConfiguration": {
```

```

        "TargetValue": 70.0
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
},
{
  "RegionName": "us-east-2",
  "GlobalSecondaryIndexes": [],
  "ReplicaProvisionedReadCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
}

```

```
    ]
  }
}
```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeTableReplicaAutoScaling](#) Befehlsreferenz. AWS CLI

describe-table

Das folgende Codebeispiel zeigt die Verwendung `describe-table`.

AWS CLI

Um eine Tabelle zu beschreiben

Das folgende `describe-table` Beispiel beschreibt die `MusicCollection` Tabelle.

```
aws dynamodb describe-table \
  --table-name MusicCollection
```

Ausgabe:

```
{
  "Table": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "WriteCapacityUnits": 5,
      "ReadCapacityUnits": 5
    },
    "TableSizeBytes": 0,
  }
}
```

```
    "TableName": "MusicCollection",
    "TableStatus": "ACTIVE",
    "KeySchema": [
      {
        "KeyType": "HASH",
        "AttributeName": "Artist"
      },
      {
        "KeyType": "RANGE",
        "AttributeName": "SongTitle"
      }
    ],
    "ItemCount": 0,
    "CreationDateTime": 1421866952.062
  }
}
```

Weitere Informationen finden Sie unter [Describing a Table](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeTable AWS CLI Befehlsreferenz](#).

describe-time-to-live

Das folgende Codebeispiel zeigt die Verwendung `describe-time-to-live`.

AWS CLI

So zeigen Sie die Time-to-Live-Einstellungen für eine Tabelle an

Im folgenden `describe-time-to-live` Beispiel werden die Time-to-Live-Einstellungen für die `MusicCollection` Tabelle angezeigt.

```
aws dynamodb describe-time-to-live \
  --table-name MusicCollection
```

Ausgabe:

```
{
  "TimeToLiveDescription": {
    "TimeToLiveStatus": "ENABLED",
    "AttributeName": "ttl"
  }
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Time to Live](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DescribeTimeToLive AWS CLI](#) Befehlsreferenz.

get-item

Das folgende Codebeispiel zeigt die Verwendung `get-item`.

AWS CLI

Beispiel 1: Um ein Element in einer Tabelle zu lesen

Im folgenden `get-item` Beispiel wird ein Element aus der `MusicCollection` Tabelle abgerufen. Die Tabelle hat einen hash-and-range Primärschlüssel (`Artist` und `SongTitle`), daher müssen Sie diese beiden Attribute angeben. Der Befehl fordert auch Informationen über die durch den Vorgang verbrauchte Lesekapazität an.

```
aws dynamodb get-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-consumed-capacity TOTAL
```

Inhalt von `key.json`:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Ausgabe:

```
{  
  "Item": {  
    "AlbumTitle": {  
      "S": "Songs About Life"  
    },  
    "SongTitle": {
```

```
        "S": "Happy Day"
    },
    "Artist": {
        "S": "Acme Band"
    }
},
"ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5
}
}
```

Weitere Informationen finden Sie unter [Artikel lesen](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 2: Um ein Element mit einem konsistenten Lesevorgang zu lesen

Im folgenden Beispiel wird mithilfe stark konsistenter Lesevorgänge ein Element aus der MusicCollection Tabelle abgerufen.

```
aws dynamodb get-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --consistent-read \  
  --return-consumed-capacity TOTAL
```

Inhalt von key.json:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Ausgabe:

```
{  
  "Item": {  
    "AlbumTitle": {  
      "S": "Songs About Life"  
    },  
    "SongTitle": {
```

```

        "S": "Happy Day"
    },
    "Artist": {
        "S": "Acme Band"
    }
},
"ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
}
}

```

Weitere Informationen finden Sie unter [Artikel lesen](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 3: Um bestimmte Attribute eines Artikels abzurufen

Im folgenden Beispiel wird ein Projektionsausdruck verwendet, um nur drei Attribute des gewünschten Elements abzurufen.

```

aws dynamodb get-item \
  --table-name ProductCatalog \
  --key '{"Id": {"N": "102"}}' \
  --projection-expression "#T, #C, #P" \
  --expression-attribute-names file://names.json

```

Inhalt von `names.json`:

```

{
  "#T": "Title",
  "#C": "ProductCategory",
  "#P": "Price"
}

```

Ausgabe:

```

{
  "Item": {
    "Price": {
      "N": "20"
    },
    "Title": {

```



```
        "S": "Book 102 Title"
      },
      "ProductCategory": {
        "S": "Book"
      }
    }
  }
}
```

Weitere Informationen finden Sie unter [Artikel lesen](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [GetItem AWS CLI Befehlsreferenz](#).

list-backups

Das folgende Codebeispiel zeigt die Verwendung `list-backups`.

AWS CLI

Beispiel 1: Um alle vorhandenen DynamoDB-Backups aufzulisten

Das folgende `list-backups` Beispiel listet alle Ihre vorhandenen Backups auf.

```
aws dynamodb list-backups
```

Ausgabe:

```
{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    },
  ],
}
```

```

    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
      "BackupName": "MusicCollectionBackup2",
      "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 400
    }
  ]
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

Beispiel 2: Um von Benutzern erstellte Backups in einem bestimmten Zeitraum aufzulisten

Im folgenden Beispiel werden nur Backups der MusicCollection Tabelle aufgeführt, die vom Benutzer erstellt wurden (nicht die automatisch von DynamoDB erstellten), deren Erstellungsdatum zwischen dem 1. Januar 2020 und dem 1. März 2020 liegt.

```

aws dynamodb list-backups \
  --table-name MusicCollection \
  --time-range-lower-bound 1577836800 \
  --time-range-upper-bound 1583020800 \
  --backup-type USER

```

Ausgabe:

```

{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",

```

```

        "BackupName": "MusicCollectionBackup1",
        "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
        "BackupStatus": "AVAILABLE",
        "BackupType": "USER",
        "BackupSizeBytes": 170
    }
]
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

Beispiel 3: Um die Seitengröße zu begrenzen

Das folgende Beispiel gibt eine Liste aller vorhandenen Backups zurück, ruft jedoch bei jedem Aufruf nur ein Element ab und führt bei Bedarf mehrere Aufrufe durch, um die gesamte Liste abzurufen. Die Begrenzung der Seitengröße ist nützlich, wenn Listenbefehle für eine große Anzahl von Ressourcen ausgeführt werden. Dies kann bei Verwendung der Standardseitengröße von 1000 zu einem Timeout-Fehler führen.

```

aws dynamodb list-backups \
  --page-size 1

```

Ausgabe:

```

{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    },
    {
      "TableName": "MusicCollection",

```

```

        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
        "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
        "BackupName": "MusicCollectionBackup2",
        "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
        "BackupStatus": "AVAILABLE",
        "BackupType": "USER",
        "BackupSizeBytes": 400
    }
]
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

Beispiel 4: Um die Anzahl der zurückgegebenen Artikel zu begrenzen

Im folgenden Beispiel wird die Anzahl der zurückgegebenen Elemente auf 1 begrenzt. Die Antwort enthält einen NextToken Wert, mit dem die nächste Ergebnisseite abgerufen werden kann.

```

aws dynamodb list-backups \
  --max-items 1

```

Ausgabe:

```

{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    }
  ]
}

```

```

    ],
    "NextToken":
    "abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0ppqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
  }

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

Beispiel 5: So rufen Sie die nächste Ergebnisseite ab

Der folgende Befehl verwendet den NextToken Wert eines vorherigen Aufrufs des list-backups Befehls, um eine weitere Ergebnisseite abzurufen. Da die Antwort in diesem Fall keinen NextToken Wert enthält, wissen wir, dass wir das Ende der Ergebnisse erreicht haben.

```

aws dynamodb list-backups \
  --starting-
  token abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0ppqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9

```

Output

```

{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/backup/01234567890123-b2abc345",
      "BackupName": "MusicCollectionBackup2",
      "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 400
    }
  ]
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

- APIEinzelheiten finden Sie unter [ListBackups](#)Befehlsreferenz.AWS CLI

list-contributor-insights

Das folgende Codebeispiel zeigt die Verwendung `list-contributor-insights`.

AWS CLI

Beispiel 1: Um eine Liste von Contributor Insights-Zusammenfassungen anzuzeigen

Im folgenden `list-contributor-insights` Beispiel wird eine Liste von Contributor Insights-Zusammenfassungen angezeigt.

```
aws dynamodb list-contributor-insights
```

Ausgabe:

```
{
  "ContributorInsightsSummaries": [
    {
      "TableName": "MusicCollection",
      "IndexName": "AlbumTitle-index",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "ProductCatalog",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Forum",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Reply",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Thread",
      "ContributorInsightsStatus": "ENABLED"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Analysieren des Datenzugriffs mithilfe von CloudWatch Contributor Insights for DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 2: Um die Anzahl der zurückgegebenen Artikel zu begrenzen

Im folgenden Beispiel wird die Anzahl der zurückgegebenen Artikel auf 4 begrenzt. Die Antwort enthält einen `NextToken` Wert, mit dem die nächste Ergebnisseite abgerufen werden kann.

```
aws dynamodb list-contributor-insights \  
  --max-results 4
```

Ausgabe:

```
{  
  "ContributorInsightsSummaries": [  
    {  
      "TableName": "MusicCollection",  
      "IndexName": "AlbumTitle-index",  
      "ContributorInsightsStatus": "ENABLED"  
    },  
    {  
      "TableName": "ProductCatalog",  
      "ContributorInsightsStatus": "ENABLED"  
    },  
    {  
      "TableName": "Forum",  
      "ContributorInsightsStatus": "ENABLED"  
    }  
  ],  
  "NextToken":  
  "abCDeFGhIJKlMnOPqrSTUvwXYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"  
}
```

Weitere Informationen finden Sie unter [Analysieren des Datenzugriffs mithilfe von CloudWatch Contributor Insights for DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 3: So rufen Sie die nächste Ergebnisseite ab

Der folgende Befehl verwendet den `NextToken` Wert eines vorherigen Aufrufs des `list-contributor-insights` Befehls, um eine weitere Ergebnisseite abzurufen. Da die Antwort in diesem Fall keinen `NextToken` Wert enthält, wissen wir, dass wir das Ende der Ergebnisse erreicht haben.

```
aws dynamodb list-contributor-insights \  
  --max-results 4 \  
  --next-token abCDeFGhIJKlMnOPqrSTUvwXYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

```
--next-
```

```
token abCDeFGhiJKLmnOPqrSTuvwXYZ1aBCdEFghijk7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9
```

Ausgabe:

```
{
  "ContributorInsightsSummaries": [
    {
      "TableName": "Reply",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Thread",
      "ContributorInsightsStatus": "ENABLED"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Analysieren des Datenzugriffs mithilfe von CloudWatch Contributor Insights for DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz. [ListContributorInsights](#) AWS CLI

list-global-tables

Das folgende Codebeispiel zeigt die Verwendung `list-global-tables`.

AWS CLI

Um bestehende globale DynamoDB-Tabellen aufzulisten

Das folgende `list-global-tables` Beispiel listet alle Ihre vorhandenen globalen Tabellen auf.

```
aws dynamodb list-global-tables
```

Ausgabe:

```
{
  "GlobalTables": [
    {
      "GlobalTableName": "MusicCollection",
      "ReplicationGroup": [
```



```
{
  "RegionName": "us-east-2"
},
{
  "RegionName": "us-east-1"
}
]
}
]
```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [ListGlobalTables](#) Befehlsreferenz.AWS CLI

list-tables

Das folgende Codebeispiel zeigt die Verwendung `list-tables`.

AWS CLI

Beispiel 1: Um Tabellen aufzulisten

Das folgende `list-tables` Beispiel listet alle Tabellen auf, die dem AWS Girokonto und der Region zugeordnet sind.

```
aws dynamodb list-tables
```

Ausgabe:

```
{
  "TableNames": [
    "Forum",
    "ProductCatalog",
    "Reply",
    "Thread"
  ]
}
```

Weitere Informationen finden Sie unter [Tabellennamen auflisten](#) im Amazon DynamoDB Developer Guide.

Beispiel 2: Um die Seitengröße zu begrenzen

Das folgende Beispiel gibt eine Liste aller vorhandenen Tabellen zurück, ruft jedoch bei jedem Aufruf nur ein Element ab und führt bei Bedarf mehrere Aufrufe durch, um die gesamte Liste abzurufen. Die Begrenzung der Seitengröße ist nützlich, wenn Listenbefehle für eine große Anzahl von Ressourcen ausgeführt werden. Dies kann bei Verwendung der Standardseitengröße von 1000 zu einem Timeout-Fehler führen.

```
aws dynamodb list-tables \  
  --page-size 1
```

Ausgabe:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog",  
    "Reply",  
    "Thread"  
  ]  
}
```

Weitere Informationen finden Sie unter [Tabellennamen auflisten](#) im Amazon DynamoDB Developer Guide.

Beispiel 3: Um die Anzahl der zurückgegebenen Artikel zu begrenzen

Im folgenden Beispiel wird die Anzahl der zurückgegebenen Artikel auf 2 begrenzt. Die Antwort enthält einen NextToken Wert, mit dem die nächste Ergebnisseite abgerufen werden kann.

```
aws dynamodb list-tables \  
  --max-items 2
```

Ausgabe:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog"  
  ],  
  "NextToken": "  
"
```

```
"NextToken":
"abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Weitere Informationen finden Sie unter [Tabellennamen auflisten](#) im Amazon DynamoDB Developer Guide.

Beispiel 4: So rufen Sie die nächste Ergebnisseite ab

Der folgende Befehl verwendet den NextToken Wert eines vorherigen Aufrufs des `list-tables` Befehls, um eine weitere Ergebnisseite abzurufen. Da die Antwort in diesem Fall keinen NextToken Wert enthält, wissen wir, dass wir das Ende der Ergebnisse erreicht haben.

```
aws dynamodb list-tables \
  --starting-
  token abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Ausgabe:

```
{
  "TableNames": [
    "Reply",
    "Thread"
  ]
}
```

Weitere Informationen finden Sie unter [Tabellennamen auflisten](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [ListTables AWS CLI Befehlsreferenz](#).

list-tags-of-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-of-resource`.

AWS CLI

Beispiel 1: So listen Sie die Tags einer DynamoDB-Ressource auf

Im folgenden `list-tags-of-resource` Beispiel werden Tags für die MusicCollection Tabelle angezeigt.

```
aws dynamodb list-tags-of-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "Owner",  
      "Value": "blueTeam"  
    },  
    {  
      "Key": "Environment",  
      "Value": "Production"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Tagging for DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 2: Um die Anzahl der zurückgegebenen Tags zu begrenzen

Im folgenden Beispiel wird die Anzahl der zurückgegebenen Tags auf 1 begrenzt. Die Antwort enthält einen NextToken Wert, mit dem die nächste Ergebnisseite abgerufen werden kann.

```
aws dynamodb list-tags-of-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --max-items 1
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "Owner",  
      "Value": "blueTeam"  
    }  
  ],  
  "NextToken":  
  "abCDeFGhiJKlmnOPqrSTuvwXYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
```

```
}
```

Weitere Informationen finden Sie unter [Tagging for DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 3: So rufen Sie die nächste Ergebnisseite ab

Der folgende Befehl verwendet den `NextToken` Wert eines vorherigen Aufrufs des `list-tags-of-resource` Befehls, um eine weitere Ergebnisseite abzurufen. Da die Antwort in diesem Fall keinen `NextToken` Wert enthält, wissen wir, dass wir das Ende der Ergebnisse erreicht haben.

```
aws dynamodb list-tags-of-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
  --starting-
token abCDeFGhiJKLmnOPqrSTuvwXYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "Environment",
      "Value": "Production"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging for DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz. [ListTagsOfResource](#) AWS CLI

put-item

Das folgende Codebeispiel zeigt die Verwendung `put-item`.

AWS CLI

Beispiel 1: Um ein Element zu einer Tabelle hinzuzufügen

Das folgende `put-item` Beispiel fügt der `MusicCollection` Tabelle ein neues Element hinzu.

```
aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item file://item.json \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

Inhalt von `item.json`:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Call Me Today"},  
  "AlbumTitle": {"S": "Greatest Hits"}  
}
```

Ausgabe:

```
{  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 1.0  
  },  
  "ItemCollectionMetrics": {  
    "ItemCollectionKey": {  
      "Artist": {  
        "S": "No One You Know"  
      }  
    },  
    "SizeEstimateRangeGB": [  
      0.0,  
      1.0  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Artikel schreiben](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 2: Um ein Element in einer Tabelle bedingt zu überschreiben

Im folgenden `put-item` Beispiel wird ein vorhandenes Element in der `MusicCollection` Tabelle nur dann überschrieben, wenn dieses vorhandene Element ein `AlbumTitle` Attribut mit dem Wert `Greatest Hits` hat. Der Befehl gibt den vorherigen Wert des Elements zurück.

```
aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item file://item.json \  
  --condition-expression "#A = :A" \  
  --expression-attribute-names file://names.json \  
  --expression-attribute-values file://values.json \  
  --return-values ALL_OLD
```

Inhalt von `item.json`:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Call Me Today"},  
  "AlbumTitle": {"S": "Somewhat Famous"}  
}
```

Inhalt von `names.json`:

```
{  
  "#A": "AlbumTitle"  
}
```

Inhalt von `values.json`:

```
{  
  ":A": {"S": "Greatest Hits"}  
}
```

Ausgabe:

```
{  
  "Attributes": {  
    "AlbumTitle": {  
      "S": "Greatest Hits"  
    },  
    "Artist": {  
      "S": "No One You Know"  
    },  
    "SongTitle": {  
      "S": "Call Me Today"  
    }  
  }  
}
```

```
}  
}
```

Wenn der Schlüssel bereits existiert, sollten Sie die folgende Ausgabe sehen:

```
A client error (ConditionalCheckFailedException) occurred when calling the PutItem  
operation: The conditional request failed.
```

Weitere Informationen finden Sie unter [Artikel schreiben](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [PutItem AWS CLI Befehlsreferenz](#).

query

Das folgende Codebeispiel zeigt die Verwendung `query`.

AWS CLI

Beispiel 1: Um eine Tabelle abzufragen

Im folgenden `query` Beispiel werden Elemente in der `MusicCollection` Tabelle abgefragt. Die Tabelle hat einen hash-and-range Primärschlüssel (`Artist` und `SongTitle`), aber diese Abfrage gibt nur den Hashschlüsselwert an. Es gibt Songtitel des Künstlers mit dem Namen „No One You Know“ zurück.

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --projection-expression "SongTitle" \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json \  
  --return-consumed-capacity TOTAL
```

Inhalt von `expression-attributes.json`:

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

Ausgabe:


```
{
  "Items": [
    {
      "SongTitle": {
        "S": "Call Me Today"
      },
      "SongTitle": {
        "S": "Scared of My Shadow"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Abfragen in DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 2: Um eine Tabelle mit stark konsistenten Lesevorgängen abzufragen und den Index in absteigender Reihenfolge zu durchlaufen

Im folgenden Beispiel wird dieselbe Abfrage wie im ersten Beispiel ausgeführt, die Ergebnisse werden jedoch in umgekehrter Reihenfolge zurückgegeben und es werden stark konsistente Lesevorgänge verwendet.

```
aws dynamodb query \
  --table-name MusicCollection \
  --projection-expression "SongTitle" \
  --key-condition-expression "Artist = :v1" \
  --expression-attribute-values file://expression-attributes.json \
  --consistent-read \
  --no-scan-index-forward \
  --return-consumed-capacity TOTAL
```

Inhalt von `expression-attributes.json`:

```
{
  ":v1": {"S": "No One You Know"}
```

```
}
```

Ausgabe:

```
{
  "Items": [
    {
      "SongTitle": {
        "S": "Scared of My Shadow"
      }
    },
    {
      "SongTitle": {
        "S": "Call Me Today"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Abfragen in DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 3: Um bestimmte Ergebnisse herauszufiltern

Das folgende Beispiel fragt die `abMusicCollection`, schließt jedoch Ergebnisse mit bestimmten Werten im `AlbumTitle` Attribut aus. Beachten Sie, dass sich dies nicht auf `ScannedCount` oder `auswirktConsumedCapacity`, da der Filter angewendet wird, nachdem die Elemente gelesen wurden.

```
aws dynamodb query \
  --table-name MusicCollection \
  --key-condition-expression "#n1 = :v1" \
  --filter-expression "NOT (#n2 IN (:v2, :v3))" \
  --expression-attribute-names file://names.json \
  --expression-attribute-values file://values.json \
  --return-consumed-capacity TOTAL
```

Inhalt von values.json:

```
{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Blue Sky Blues"},
  ":v3": {"S": "Greatest Hits"}
}
```

Inhalt von names.json:

```
{
  "#n1": "Artist",
  "#n2": "AlbumTitle"
}
```

Ausgabe:

```
{
  "Items": [
    {
      "AlbumTitle": {
        "S": "Somewhat Famous"
      },
      "Artist": {
        "S": "No One You Know"
      },
      "SongTitle": {
        "S": "Call Me Today"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Abfragen in DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 4: Um nur eine Artikelanzahl abzurufen

Das folgende Beispiel ruft eine Anzahl von Elementen ab, die der Abfrage entsprechen, ruft jedoch keines der Elemente selbst ab.

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --select COUNT \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json
```

Inhalt von `expression-attributes.json`:

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

Ausgabe:

```
{  
  "Count": 2,  
  "ScannedCount": 2,  
  "ConsumedCapacity": null  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Abfragen in DynamoDB im Amazon DynamoDB Developer Guide](#).

Beispiel 5: So fragen Sie einen Index ab

Im folgenden Beispiel wird der lokale sekundäre Index `AlbumTitleIndex` abgefragt. Die Abfrage gibt alle Attribute aus der Basistabelle zurück, die in den lokalen sekundären Index projiziert wurden. Beachten Sie, dass Sie bei der Abfrage eines lokalen Sekundärindex oder eines globalen Sekundärindex auch den Namen der Basistabelle mithilfe des `table-name` Parameters angeben müssen.

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --index-name AlbumTitleIndex \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json
```

```
--expression-attribute-values file://expression-attributes.json \  
--select ALL_PROJECTED_ATTRIBUTES \  
--return-consumed-capacity INDEXES
```

Inhalt von `expression-attributes.json`:

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

Ausgabe:

```
{  
  "Items": [  
    {  
      "AlbumTitle": {  
        "S": "Blue Sky Blues"  
      },  
      "Artist": {  
        "S": "No One You Know"  
      },  
      "SongTitle": {  
        "S": "Scared of My Shadow"  
      }  
    },  
    {  
      "AlbumTitle": {  
        "S": "Somewhat Famous"  
      },  
      "Artist": {  
        "S": "No One You Know"  
      },  
      "SongTitle": {  
        "S": "Call Me Today"  
      }  
    }  
  ],  
  "Count": 2,  
  "ScannedCount": 2,  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 0.5,  
    "Table": {
```

```

        "CapacityUnits": 0.0
    },
    "LocalSecondaryIndexes": {
        "AlbumTitleIndex": {
            "CapacityUnits": 0.5
        }
    }
}
}
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Abfragen in DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter [Query](#) in Command Reference.AWS CLI

restore-table-from-backup

Das folgende Codebeispiel zeigt die Verwendung `restore-table-from-backup`.

AWS CLI

So stellen Sie eine DynamoDB-Tabelle aus einer vorhandenen Sicherung wieder her

Im folgenden `restore-table-from-backup` Beispiel wird die angegebene Tabelle aus einer vorhandenen Sicherung wiederhergestellt.

```

aws dynamodb restore-table-from-backup \
  --target-table-name MusicCollection \
  --backup-arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
  backup/01576616366715-b4e58d3a

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",

```

```

        "AttributeType": "S"
      }
    ],
    "TableName": "MusicCollection2",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": 1576618274.326,
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection2",
    "TableId": "114865c9-5ef3-496c-b4d1-c4cbdd2d44fb",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED"
    },
    "RestoreSummary": {
      "SourceBackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
      "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "RestoreDateTime": 1576616366.715,
      "RestoreInProgress": true
    }
  }
}

```

Weitere Informationen finden Sie unter [On-Demand-Backup and Restore für DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [RestoreTableFromBackup](#) Befehlsreferenz.AWS CLI

restore-table-to-point-in-time

Das folgende Codebeispiel zeigt die Verwendung `restore-table-to-point-in-time`.

AWS CLI

So stellen Sie eine DynamoDB-Tabelle auf einen bestimmten Zeitpunkt zurück

Im folgenden `restore-table-to-point-in-time` Beispiel wird die `MusicCollection` Tabelle auf den angegebenen Zeitpunkt zurückgesetzt.

```
aws dynamodb restore-table-to-point-in-time \  
  --source-table-name MusicCollection \  
  --target-table-name MusicCollectionRestore \  
  --restore-date-time 1576622404.0
```

Ausgabe:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollectionRestore",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": 1576623311.86,  
  }  
}
```



```

    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollectionRestore",
    "TableId": "befd9e0e-1843-4dc6-a147-d6d00e85cb1f",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED"
    },
    "RestoreSummary": {
      "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "RestoreDateTime": 1576622404.0,
      "RestoreInProgress": true
    }
  }
}

```

Weitere Informationen finden Sie unter [Point-in-Time Recovery for DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [RestoreTableToPointInTime](#) Befehlsreferenz.AWS CLI

scan

Das folgende Codebeispiel zeigt die Verwendungscan.

AWS CLI

Um eine Tabelle zu scannen

Im folgenden scan Beispiel wird die gesamte MusicCollection Tabelle gescannt und die Ergebnisse dann auf Songs des Künstlers „No One You Know“ eingegrenzt. Für jedes Element werden nur der Albumtitel und der Songtitel zurückgegeben.

```

aws dynamodb scan \
  --table-name MusicCollection \
  --filter-expression "Artist = :a" \
  --projection-expression "#ST, #AT" \

```

```
--expression-attribute-names file://expression-attribute-names.json \  
--expression-attribute-values file://expression-attribute-values.json
```

Inhalt von `expression-attribute-names.json`:

```
{  
  "#ST": "SongTitle",  
  "#AT": "AlbumTitle"  
}
```

Inhalt von `expression-attribute-values.json`:

```
{  
  ":a": {"S": "No One You Know"}  
}
```

Ausgabe:

```
{  
  "Count": 2,  
  "Items": [  
    {  
      "SongTitle": {  
        "S": "Call Me Today"  
      },  
      "AlbumTitle": {  
        "S": "Somewhat Famous"  
      }  
    },  
    {  
      "SongTitle": {  
        "S": "Scared of My Shadow"  
      },  
      "AlbumTitle": {  
        "S": "Blue Sky Blues"  
      }  
    }  
  ],  
  "ScannedCount": 3,  
  "ConsumedCapacity": null  
}
```

Weitere Informationen finden Sie unter [Working with Scans in DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [Scan](#) in Command Reference.AWS CLI

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einer DynamoDB-Ressource Tags hinzu

Im folgenden `tag-resource` Beispiel wird der Tabelle ein Tag-Schlüssel/Wert-Paar hinzugefügt. `MusicCollection`

```
aws dynamodb tag-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --tags Key=Owner,Value=blueTeam
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging for DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz. [TagResource](#)AWS CLI

transact-get-items

Das folgende Codebeispiel zeigt die Verwendung `transact-get-items`.

AWS CLI

Um mehrere Elemente atomar aus einer oder mehreren Tabellen abzurufen

Im folgenden `transact-get-items` Beispiel werden mehrere Elemente atomar abgerufen.

```
aws dynamodb transact-get-items \  
  --transact-items file://transact-items.json \  
  --return-consumed-capacity TOTAL
```

Inhalt von `transact-items.json`:

```
[
  {
    "Get": {
      "Key": {
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"}
      },
      "TableName": "MusicCollection"
    }
  },
  {
    "Get": {
      "Key": {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"}
      },
      "TableName": "MusicCollection"
    }
  }
]
```

Ausgabe:

```
{
  "ConsumedCapacity": [
    {
      "TableName": "MusicCollection",
      "CapacityUnits": 4.0,
      "ReadCapacityUnits": 4.0
    }
  ],
  "Responses": [
    {
      "Item": {
        "AlbumTitle": {
          "S": "Songs About Life"
        },
        "Artist": {
          "S": "Acme Band"
        },
        "SongTitle": {
```

```

        "S": "Happy Day"
      }
    },
    {
      "Item": {
        "AlbumTitle": {
          "S": "Somewhat Famous"
        },
        "Artist": {
          "S": "No One You Know"
        },
        "SongTitle": {
          "S": "Call Me Today"
        }
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter [Managing Complex Workflows with DynamoDB Transactions](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [TransactGetItems](#) Befehlsreferenz. AWS CLI

transact-write-items

Das folgende Codebeispiel zeigt die Verwendung `transact-write-items`.

AWS CLI

Beispiel 1: Um Elemente atomar in eine oder mehrere Tabellen zu schreiben

Im folgenden `transact-write-items` Beispiel wird ein Element aktualisiert und ein anderes gelöscht. Der Vorgang schlägt fehl, wenn einer der Operationen fehlschlägt oder wenn eines der Elemente ein `Rating` Attribut enthält.

```

aws dynamodb transact-write-items \
  --transact-items file://transact-items.json \
  --return-consumed-capacity TOTAL \
  --return-item-collection-metrics SIZE

```

Inhalt der `transact-items.json` Datei:

```
[
  {
    "Update": {
      "Key": {
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"}
      },
      "UpdateExpression": "SET AlbumTitle = :newval",
      "ExpressionAttributeValues": {
        ":newval": {"S": "Updated Album Title"}
      },
      "TableName": "MusicCollection",
      "ConditionExpression": "attribute_not_exists(Rating)"
    }
  },
  {
    "Delete": {
      "Key": {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"}
      },
      "TableName": "MusicCollection",
      "ConditionExpression": "attribute_not_exists(Rating)"
    }
  }
]
```

Ausgabe:

```
{
  "ConsumedCapacity": [
    {
      "TableName": "MusicCollection",
      "CapacityUnits": 10.0,
      "WriteCapacityUnits": 10.0
    }
  ],
  "ItemCollectionMetrics": {
    "MusicCollection": [
      {
        "ItemCollectionKey": {
          "Artist": {
            "S": "No One You Know"
          }
        }
      }
    ]
  }
}
```

```

        }
      },
      "SizeEstimateRangeGB": [
        0.0,
        1.0
      ]
    },
    {
      "ItemCollectionKey": {
        "Artist": {
          "S": "Acme Band"
        }
      },
      "SizeEstimateRangeGB": [
        0.0,
        1.0
      ]
    }
  ]
}

```

Weitere Informationen finden Sie unter [Managing Complex Workflows with DynamoDB Transactions](#) im Amazon DynamoDB Developer Guide.

Beispiel 2: Um Elemente mithilfe eines Client-Request-Tokens atomar zu schreiben

Der folgende Befehl verwendet ein Client-Anforderungstoken, um den Aufruf an `transact-write-items` idempotent zu richten, was bedeutet, dass mehrere Aufrufe den gleichen Effekt haben wie ein einziger Aufruf.

```

aws dynamodb transact-write-items \
  --transact-items file://transact-items.json \
  --client-request-token abc123

```

Inhalt der Datei: `transact-items.json`

```

[
  {
    "Update": {
      "Key": {
        "Artist": {"S": "Acme Band"},

```

```

        "SongTitle": {"S": "Happy Day"}
    },
    "UpdateExpression": "SET AlbumTitle = :newval",
    "ExpressionAttributeValues": {
        ":newval": {"S": "Updated Album Title"}
    },
    "TableName": "MusicCollection",
    "ConditionExpression": "attribute_not_exists(Rating)"
}
},
{
    "Delete": {
        "Key": {
            "Artist": {"S": "No One You Know"},
            "SongTitle": {"S": "Call Me Today"}
        },
        "TableName": "MusicCollection",
        "ConditionExpression": "attribute_not_exists(Rating)"
    }
}
]

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Complex Workflows with DynamoDB Transactions](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [TransactWriteItems](#) Befehlsreferenz.AWS CLI

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

So entfernen Sie ein Tag aus einer DynamoDB-Ressource

Im folgenden `untag-resource` Beispiel wird das Tag mit dem Schlüssel `Owner` aus der `MusicCollection` Tabelle entfernt.

```

aws dynamodb untag-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \

```



```
--tag-keys Owner
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging for DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz. [UntagResource](#) AWS CLI

update-continuous-backups

Das folgende Codebeispiel zeigt die Verwendung `update-continuous-backups`.

AWS CLI

So aktualisieren Sie die Einstellungen für kontinuierliche Backups für eine DynamoDB-Tabelle

Das folgende `update-continuous-backups` Beispiel ermöglicht die point-in-time Wiederherstellung der `MusicCollection` Tabelle.

```
aws dynamodb update-continuous-backups \  
  --table-name MusicCollection \  
  --point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

Ausgabe:

```
{  
  "ContinuousBackupsDescription": {  
    "ContinuousBackupsStatus": "ENABLED",  
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "ENABLED",  
      "EarliestRestorableDateTime": 1576622404.0,  
      "LatestRestorableDateTime": 1576622404.0  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Point-in-Time Recovery for DynamoDB](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [UpdateContinuousBackups](#) Befehlsreferenz. AWS CLI

update-contributor-insights

Das folgende Codebeispiel zeigt die Verwendung `update-contributor-insights`.

AWS CLI

Um Contributor Insights für eine Tabelle zu aktivieren

Im folgenden `update-contributor-insights` Beispiel wird Contributor Insights für die `MusicCollection` Tabelle und den `AlbumTitle-index` globalen Sekundärindex aktiviert.

```
aws dynamodb update-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index \  
  --contributor-insights-action ENABLE
```

Ausgabe:

```
{  
  "TableName": "MusicCollection",  
  "IndexName": "AlbumTitle-index",  
  "ContributorInsightsStatus": "ENABLING"  
}
```

Weitere Informationen finden Sie unter [Analysieren des Datenzugriffs mithilfe von CloudWatch Contributor Insights for DynamoDB im Amazon DynamoDB Developer Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz. [UpdateContributorInsights](#) AWS CLI

update-global-table-settings

Das folgende Codebeispiel zeigt die Verwendung `update-global-table-settings`.

AWS CLI

So aktualisieren Sie die bereitgestellten Schreibkapazitätseinstellungen für eine globale DynamoDB-Tabelle

Im folgenden `update-global-table-settings` Beispiel wird die bereitgestellte Schreibkapazität der `MusicCollection` globalen Tabelle auf 15 festgelegt.

```
aws dynamodb update-global-table-settings \  
--global-table-name MusicCollection \  
--global-table-provisioned-write-capacity-units 15
```

Ausgabe:

```
{  
  "GlobalTableName": "MusicCollection",  
  "ReplicaSettings": [  
    {  
      "RegionName": "eu-west-1",  
      "ReplicaStatus": "UPDATING",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 10,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    },  
    {  
      "RegionName": "us-east-1",  
      "ReplicaStatus": "UPDATING",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 10,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    },  
    {  
      "RegionName": "us-east-2",  
      "ReplicaStatus": "UPDATING",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 10,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    }  
  ]  
}
```

```

    }
  }
]
}

```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [UpdateGlobalTableSettings](#) Befehlsreferenz. AWS CLI

update-global-table

Das folgende Codebeispiel zeigt die Verwendung `update-global-table`.

AWS CLI

So aktualisieren Sie eine globale DynamoDB-Tabelle

Im folgenden `update-global-table` Beispiel wird der globalen Tabelle ein Replikat in der angegebenen Region hinzugefügt. `MusicCollection`

```

aws dynamodb update-global-table \
  --global-table-name MusicCollection \
  --replica-updates Create={RegionName=eu-west-1}

```

Ausgabe:

```

{
  "GlobalTableDescription": {
    "ReplicationGroup": [
      {
        "RegionName": "eu-west-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-east-1"
      }
    ],
    "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
    "CreationDateTime": 1576625818.532,

```

```
    "GlobalTableStatus": "ACTIVE",
    "GlobalTableName": "MusicCollection"
  }
}
```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [UpdateGlobalTable](#) Befehlsreferenz.AWS CLI

update-item

Das folgende Codebeispiel zeigt die Verwendung `update-item`.

AWS CLI

Beispiel 1: Um ein Element in einer Tabelle zu aktualisieren

Das folgende `update-item`-Beispiel aktualisiert ein Element in der Tabelle `MusicCollection`. Es fügt ein neues Attribut (`Year`) hinzu und ändert das `AlbumTitle` Attribut. Alle Attribute im Element, so wie sie nach der Aktualisierung erscheinen, werden in der Antwort zurückgegeben.

```
aws dynamodb update-item \
  --table-name MusicCollection \
  --key file://key.json \
  --update-expression "SET #Y = :y, #AT = :t" \
  --expression-attribute-names file://expression-attribute-names.json \
  --expression-attribute-values file://expression-attribute-values.json \
  --return-values ALL_NEW \
  --return-consumed-capacity TOTAL \
  --return-item-collection-metrics SIZE
```

Inhalt von `key.json`:

```
{
  "Artist": {"S": "Acme Band"},
  "SongTitle": {"S": "Happy Day"}
}
```

Inhalt von `expression-attribute-names.json`:

```
{
```

```
"#Y":"Year", "#AT":"AlbumTitle"  
}
```

Inhalt von `expression-attribute-values.json`:

```
{  
  "y":{"N": "2015"},  
  "t":{"S": "Louder Than Ever"}  
}
```

Ausgabe:

```
{  
  "Attributes": {  
    "AlbumTitle": {  
      "S": "Louder Than Ever"  
    },  
    "Awards": {  
      "N": "10"  
    },  
    "Artist": {  
      "S": "Acme Band"  
    },  
    "Year": {  
      "N": "2015"  
    },  
    "SongTitle": {  
      "S": "Happy Day"  
    }  
  },  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 3.0  
  },  
  "ItemCollectionMetrics": {  
    "ItemCollectionKey": {  
      "Artist": {  
        "S": "Acme Band"  
      }  
    },  
    "SizeEstimateRangeGB": [  
      0.0,  
      1.0  
    ]  
  }  
}
```

```

    ]
  }
}

```

Weitere Informationen finden Sie unter [Artikel schreiben](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 2: Um einen Artikel unter bestimmten Bedingungen zu aktualisieren

Im folgenden Beispiel wird ein Element in der MusicCollection Tabelle aktualisiert, jedoch nur, wenn das vorhandene Element noch kein Year Attribut besitzt.

```

aws dynamodb update-item \
  --table-name MusicCollection \
  --key file://key.json \
  --update-expression "SET #Y = :y, #AT = :t" \
  --expression-attribute-names file://expression-attribute-names.json \
  --expression-attribute-values file://expression-attribute-values.json \
  --condition-expression "attribute_not_exists(#Y)"

```

Inhalt von key.json:

```

{
  "Artist": {"S": "Acme Band"},
  "SongTitle": {"S": "Happy Day"}
}

```

Inhalt von expression-attribute-names.json:

```

{
  "#Y": "Year",
  "#AT": "AlbumTitle"
}

```

Inhalt von expression-attribute-values.json:

```

{
  ":y": {"N": "2015"},
  ":t": {"S": "Louder Than Ever"}
}

```

Wenn das Element bereits über ein Year Attribut verfügt, gibt DynamoDB die folgende Ausgabe zurück.

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem operation: The conditional request failed
```

Weitere Informationen finden Sie unter [Artikel schreiben](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [UpdateItem AWS CLI Befehlsreferenz](#).

update-table-replica-auto-scaling

Das folgende Codebeispiel zeigt die Verwendung `update-table-replica-auto-scaling`.

AWS CLI

So aktualisieren Sie die Auto Scaling-Einstellungen für Replikate einer globalen Tabelle

Im folgenden `update-table-replica-auto-scaling` Beispiel werden die Einstellungen für die auto Skalierung der Schreibkapazität für alle Replikate der angegebenen globalen Tabelle aktualisiert.

```
aws dynamodb update-table-replica-auto-scaling \  
  --table-name MusicCollection \  
  --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

Inhalt von `auto-scaling-policy.json`:

```
{  
  "MinimumUnits": 10,  
  "MaximumUnits": 100,  
  "AutoScalingDisabled": false,  
  "ScalingPolicyUpdate": {  
    "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",  
    "TargetTrackingScalingPolicyConfiguration": {  
      "TargetValue": 80  
    }  
  }  
}
```


Ausgabe:

```
{
  "TableAutoScalingDescription": {
    "TableName": "MusicCollection",
    "TableStatus": "ACTIVE",
    "Replicas": [
      {
        "RegionName": "eu-central-1",
        "GlobalSecondaryIndexes": [],
        "ReplicaProvisionedReadCapacityAutoScalingSettings": {
          "MinimumUnits": 5,
          "MaximumUnits": 40000,
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
          "ScalingPolicies": [
            {
              "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
              "TargetTrackingScalingPolicyConfiguration": {
                "TargetValue": 70.0
              }
            }
          ]
        },
        "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
          "MinimumUnits": 10,
          "MaximumUnits": 100,
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
          "ScalingPolicies": [
            {
              "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
              "TargetTrackingScalingPolicyConfiguration": {
                "TargetValue": 80.0
              }
            }
          ]
        },
        "ReplicaStatus": "ACTIVE"
      }
    ],
  },
}
```

```
{
  "RegionName": "us-east-1",
  "GlobalSecondaryIndexes": [],
  "ReplicaProvisionedReadCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
    "MinimumUnits": 10,
    "MaximumUnits": 100,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 80.0
        }
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
},
{
  "RegionName": "us-east-2",
  "GlobalSecondaryIndexes": [],
  "ReplicaProvisionedReadCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
```

```

        "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
        "ScalingPolicies": [
            {
                "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
                "TargetTrackingScalingPolicyConfiguration": {
                    "TargetValue": 70.0
                }
            }
        ],
    },
    "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
        "MinimumUnits": 10,
        "MaximumUnits": 100,
        "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
        "ScalingPolicies": [
            {
                "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
                "TargetTrackingScalingPolicyConfiguration": {
                    "TargetValue": 80.0
                }
            }
        ],
    },
    "ReplicaStatus": "ACTIVE"
}
]
}
}

```

Weitere Informationen finden Sie unter [DynamoDB Global Tables](#) im Amazon DynamoDB Developer Guide.

- API-Einheiten finden Sie unter [UpdateTableReplicaAutoScaling](#) Befehlsreferenz.AWS CLI

update-table

Das folgende Codebeispiel zeigt die Verwendung `update-table`.

AWS CLI

Beispiel 1: Um den Abrechnungsmodus einer Tabelle zu ändern

Das folgende `update-table` Beispiel erhöht die bereitgestellte Lese- und Schreibkapazität für die `MusicCollection` Tabelle.

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --billing-mode PROVISIONED \  
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10
```

Ausgabe:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "AlbumTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "UPDATING",  
    "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
```

```

    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2020-07-28T13:18:18.921000-07:00",
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 15,
      "WriteCapacityUnits": 10
    },
    "TableSizeBytes": 182,
    "ItemCount": 2,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED",
      "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
    }
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer Tabelle](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 2: So erstellen Sie einen globalen sekundären Index

Das folgende Beispiel fügt der MusicCollection Tabelle einen globalen sekundären Index hinzu.

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \
  --global-secondary-index-updates file://gsi-updates.json

```

Inhalt von `gsi-updates.json`:

```

[
  {
    "Create": {
      "IndexName": "AlbumTitle-index",
      "KeySchema": [
        {
          "AttributeName": "AlbumTitle",
          "KeyType": "HASH"
        }
      ]
    }
  }
]

```

```

    "ProvisionedThroughput": {
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 10
    },
    "Projection": {
      "ProjectionType": "ALL"
    }
  }
}
]

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "AlbumTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "MusicCollection",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "UPDATING",
    "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
    "ProvisionedThroughput": {

```

```

        "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 15,
        "WriteCapacityUnits": 10
    },
    "TableSizeBytes": 182,
    "ItemCount": 2,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
    "BillingModeSummary": {
        "BillingMode": "PROVISIONED",
        "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
    },
    "GlobalSecondaryIndexes": [
        {
            "IndexName": "AlbumTitle-index",
            "KeySchema": [
                {
                    "AttributeName": "AlbumTitle",
                    "KeyType": "HASH"
                }
            ],
            "Projection": {
                "ProjectionType": "ALL"
            },
            "IndexStatus": "CREATING",
            "Backfilling": false,
            "ProvisionedThroughput": {
                "NumberOfDecreasesToday": 0,
                "ReadCapacityUnits": 10,
                "WriteCapacityUnits": 10
            },
            "IndexSizeBytes": 0,
            "ItemCount": 0,
            "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
        }
    ]
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer Tabelle](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 3: So aktivieren Sie DynamoDB Streams für eine Tabelle

Der folgende Befehl aktiviert DynamoDB Streams für die MusicCollection Tabelle.

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE
```

Ausgabe:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "AlbumTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "UPDATING",  
    "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",  
    "ProvisionedThroughput": {  
      "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",  
      "NumberOfDecreasesToday": 0,  
      "ReadCapacityUnits": 15,  
    }  
  }  
}
```



```
    "WriteCapacityUnits": 10
  },
  "TableSizeBytes": 182,
  "ItemCount": 2,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
  "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
  "BillingModeSummary": {
    "BillingMode": "PROVISIONED",
    "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
  },
  "LocalSecondaryIndexes": [
    {
      "IndexName": "AlbumTitleIndex",
      "KeySchema": [
        {
          "AttributeName": "Artist",
          "KeyType": "HASH"
        },
        {
          "AttributeName": "AlbumTitle",
          "KeyType": "RANGE"
        }
      ],
      "Projection": {
        "ProjectionType": "INCLUDE",
        "NonKeyAttributes": [
          "Year",
          "Genre"
        ]
      },
      "IndexSizeBytes": 139,
      "ItemCount": 2,
      "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
    }
  ],
  "GlobalSecondaryIndexes": [
    {
      "IndexName": "AlbumTitle-index",
      "KeySchema": [
        {
          "AttributeName": "AlbumTitle",
          "KeyType": "HASH"
        }
      ]
    }
  ]
}
```

```

    ],
    "Projection": {
      "ProjectionType": "ALL"
    },
    "IndexStatus": "ACTIVE",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 10
    },
    "IndexSizeBytes": 0,
    "ItemCount": 0,
    "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
  }
],
"StreamSpecification": {
  "StreamEnabled": true,
  "StreamViewType": "NEW_IMAGE"
},
"LatestStreamLabel": "2020-07-28T21:53:39.112",
"LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112"
}
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer Tabelle](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

Beispiel 4: Um die serverseitige Verschlüsselung zu aktivieren

Das folgende Beispiel aktiviert die serverseitige Verschlüsselung für die MusicCollection Tabelle.

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --sse-specification Enabled=true,SSEType=KMS

```

Ausgabe:

```

{
  "TableDescription": {
    "AttributeDefinitions": [

```

```
{
  "AttributeName": "AlbumTitle",
  "AttributeType": "S"
},
{
  "AttributeName": "Artist",
  "AttributeType": "S"
},
{
  "AttributeName": "SongTitle",
  "AttributeType": "S"
}
],
"TableName": "MusicCollection",
"KeySchema": [
  {
    "AttributeName": "Artist",
    "KeyType": "HASH"
  },
  {
    "AttributeName": "SongTitle",
    "KeyType": "RANGE"
  }
],
"TableStatus": "ACTIVE",
"CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
"ProvisionedThroughput": {
  "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
  "NumberOfDecreasesToday": 0,
  "ReadCapacityUnits": 15,
  "WriteCapacityUnits": 10
},
"TableSizeBytes": 182,
"ItemCount": 2,
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
"TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
"BillingModeSummary": {
  "BillingMode": "PROVISIONED",
  "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
},
"LocalSecondaryIndexes": [
  {
    "IndexName": "AlbumTitleIndex",
    "KeySchema": [
```

```
        {
            "AttributeName": "Artist",
            "KeyType": "HASH"
        },
        {
            "AttributeName": "AlbumTitle",
            "KeyType": "RANGE"
        }
    ],
    "Projection": {
        "ProjectionType": "INCLUDE",
        "NonKeyAttributes": [
            "Year",
            "Genre"
        ]
    },
    "IndexSizeBytes": 139,
    "ItemCount": 2,
    "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
    }
],
"GlobalSecondaryIndexes": [
    {
        "IndexName": "AlbumTitle-index",
        "KeySchema": [
            {
                "AttributeName": "AlbumTitle",
                "KeyType": "HASH"
            }
        ],
        "Projection": {
            "ProjectionType": "ALL"
        },
        "IndexStatus": "ACTIVE",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 10
        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
```

```

    }
  ],
  "StreamSpecification": {
    "StreamEnabled": true,
    "StreamViewType": "NEW_IMAGE"
  },
  "LatestStreamLabel": "2020-07-28T21:53:39.112",
  "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112",
  "SSEDescription": {
    "Status": "UPDATING"
  }
}
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer Tabelle](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [UpdateTable AWS CLI Befehlsreferenz](#).

update-time-to-live

Das folgende Codebeispiel zeigt die Verwendung `update-time-to-live`.

AWS CLI

Um die Time-to-Live-Einstellungen in einer Tabelle zu aktualisieren

Im folgenden `update-time-to-live` Beispiel wird Time to Live für die angegebene Tabelle aktiviert.

```

aws dynamodb update-time-to-live \
  --table-name MusicCollection \
  --time-to-live-specification Enabled=true,AttributeName=ttl

```

Ausgabe:

```

{
  "TimeToLiveSpecification": {
    "Enabled": true,
    "AttributeName": "ttl"
  }
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Time to Live](#) im Amazon DynamoDB DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [UpdateTimeToLive AWS CLI](#) Befehlsreferenz.

Beispiele für DynamoDB Streams mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Streams AWS Command Line Interface mit DynamoDB Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-stream

Das folgende Codebeispiel zeigt die Verwendung `describe-stream`.

AWS CLI

So rufen Sie Informationen über einen DynamoDB-Stream ab

Der folgende `describe-stream` Befehl zeigt Informationen über den spezifischen DynamoDB-Stream an.

```
aws dynamodbstreams describe-stream \  
  --stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/  
stream/2019-10-22T18:02:01.576
```

Ausgabe:

```
{
  "StreamDescription": {
    "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576",
    "StreamLabel": "2019-10-22T18:02:01.576",
    "StreamStatus": "ENABLED",
    "StreamViewType": "NEW_AND_OLD_IMAGES",
    "CreationRequestDateTime": 1571767321.571,
    "TableName": "Music",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "Shards": [
      {
        "ShardId": "shardId-00000001571767321804-697ce3d2",
        "SequenceNumberRange": {
          "StartingSequenceNumber": "4000000000000642977831",
          "EndingSequenceNumber": "4000000000000642977831"
        }
      },
      {
        "ShardId": "shardId-00000001571780995058-40810d86",
        "SequenceNumberRange": {
          "StartingSequenceNumber": "75740000000005655171150"
        },
        "ParentShardId": "shardId-00000001571767321804-697ce3d2"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Erfassen von Tabellenaktivitäten mit DynamoDB Streams im Amazon DynamoDB](#) DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DescribeStream](#) Befehlsreferenz.AWS CLI

get-records

Das folgende Codebeispiel zeigt die Verwendung `get-records`.

AWS CLI

Um Datensätze aus einem Dynamodb-Stream abzurufen

Der folgende `get-records` Befehl ruft Datensätze mit dem angegebenen Amazon DynamoDB-Shard-Iterator ab.

```
aws dynamodbstreams get-records \
  --shard-iterator "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmi0Nmy6xVW4IRcIIbs1z07MNI
+CjNPLqQjnyRSAnf0wWmKhL1/KNParWSfz2odf780o00bIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QI10XPc8mRBfNIazfrVCjJK8/jsjCzsqMyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPI dmTRG
+w/LEGS05ha1qNP+VL4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+DtcM5BHayBs
+PVYHITaTLiInFLT
+LCwvaz1QH3MY3b8A05Z800wjpkM60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
```

Ausgabe:

```
{
  "Records": [
    {
      "eventID": "c3b5d798eef6215d42f8137b19a88e50",
      "eventName": "INSERT",
      "eventVersion": "1.1",
      "eventSource": "aws:dynamodb",
      "awsRegion": "us-west-1",
      "dynamodb": {
        "ApproximateCreationDateTime": 1571849028.0,
        "Keys": {
          "Artist": {
            "S": "No One You Know"
          },
          "SongTitle": {
            "S": "Call Me Today"
          }
        }
      },
      "NewImage": {
```



```
        "AlbumTitle": {
            "S": "Somewhat Famous"
        },
        "Artist": {
            "S": "No One You Know"
        },
        "Awards": {
            "N": "1"
        },
        "SongTitle": {
            "S": "Call Me Today"
        }
    },
    "SequenceNumber": "700000000013256296913",
    "SizeBytes": 119,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
},
{
    "eventID": "878960a6967867e2da16b27380a27328",
    "eventName": "INSERT",
    "eventVersion": "1.1",
    "eventSource": "aws:dynamodb",
    "awsRegion": "us-west-1",
    "dynamodb": {
        "ApproximateCreationDateTime": 1571849029.0,
        "Keys": {
            "Artist": {
                "S": "Acme Band"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        },
        "NewImage": {
            "AlbumTitle": {
                "S": "Songs About Life"
            },
            "Artist": {
                "S": "Acme Band"
            },
            "Awards": {
                "N": "10"
            }
        },
    },
}
```

```
        "SongTitle": {
            "S": "Happy Day"
        }
    },
    "SequenceNumber": "800000000013256297217",
    "SizeBytes": 100,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
}
},
{
    "eventID": "520fabde080e159fc3710b15ee1d4daa",
    "eventName": "MODIFY",
    "eventVersion": "1.1",
    "eventSource": "aws:dynamodb",
    "awsRegion": "us-west-1",
    "dynamodb": {
        "ApproximateCreationDateTime": 1571849734.0,
        "Keys": {
            "Artist": {
                "S": "Acme Band"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        },
        "NewImage": {
            "AlbumTitle": {
                "S": "Updated Album Title"
            },
            "Artist": {
                "S": "Acme Band"
            },
            "Awards": {
                "N": "10"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        },
        "OldImage": {
            "AlbumTitle": {
                "S": "Songs About Life"
            },
            "Artist": {
```

```

        "S": "Acme Band"
      },
      "Awards": {
        "N": "10"
      },
      "SongTitle": {
        "S": "Happy Day"
      }
    },
    "SequenceNumber": "900000000013256687845",
    "SizeBytes": 170,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
  }
},
  "NextShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/
Music/stream/2019-10-23T16:41:08.740|1|AAAAAAAAAAAEhEI04jkFLW
+LK0wivjT8d/IHEh3iExV2xK00aTxEzVy1C1C7Kbb5+Z0W6bT9VQ2n1/
mrs7+PRia0ZCHJu7JHJVW7zlsq0i/ges3fw8GYEymyL+piEk35cx67rQqwKKyq
+Q6w9JyjreIOj4F2lWLV26lBwRTrIYC4IB7C3BZZK4715QwYdDxNdVHiSBRZX8UqoS6W0t0F87xZLNB9F/
NhYBLXi/wcGvAcBcC0TNI0H+N0Nqwt0B/
FGcKNrf8YZ0xRoNN6RgGuVWHF3px0hxEJeFZoSoJTlKeG9YcYxzi5Ci/
mhdtm7tBXnbw5c6xmsGsBqTirNj1DyJLcWl8C10UOLX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0fF/
ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWf2wvwT6Iy2xGrR2r2HzYps9dwG0arVdEITaJfWzNoL4HajMhmREZ
+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PvV6iknY6d19MrdbSTMKoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da
}
}

```

Weitere Informationen finden Sie unter [Erfassen von Tabellenaktivitäten mit DynamoDB Streams im Amazon DynamoDB](#) DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [GetRecords](#) Befehlsreferenz. AWS CLI

get-shard-iterator

Das folgende Codebeispiel zeigt die Verwendung von `get-shard-iterator`.

AWS CLI

Um einen Shard-Iterator zu bekommen

Der folgende `get-shard-iterator` Befehl ruft einen Shard-Iterator für den angegebenen Shard ab.

```
aws dynamodbstreams get-shard-iterator \
  --stream-arn arn:aws:dynamodb:us-west-1:12356789012:table/Music/
stream/2019-10-22T18:02:01.576 \
  --shard-id shardId-00000001571780995058-40810d86 \
  --shard-iterator-type LATEST
```

Ausgabe:

```
{
  "ShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmi0Nmy6xVW4IRcIIbs1z07MNI
+CjNPlqQjnyRSAnf0wWmKhL1/KNParWSfz2odf780o00bIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QI10XPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPI dmTRG
+w/1EGS05ha1qNP+V14+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+DtcM5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
}
```

Weitere Informationen finden Sie unter [Erfassen von Tabellenaktivitäten mit DynamoDB Streams im Amazon DynamoDB](#) DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [GetShardIterator](#) Befehlsreferenz. AWS CLI

list-streams

Das folgende Codebeispiel zeigt die Verwendung `list-streams`.

AWS CLI

Um DynamoDB-Streams aufzulisten

Der folgende `list-streams` Befehl listet alle vorhandenen Amazon DynamoDB DynamoDB-Streams in der AWS Standardregion auf.

```
aws dynamodbstreams list-streams
```

Ausgabe:

```
{
  "Streams": [
```

```
{
  "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576",
  "TableName": "Music",
  "StreamLabel": "2019-10-22T18:02:01.576"
}
]
```

Weitere Informationen finden Sie unter [Erfassen von Tabellenaktivitäten mit DynamoDB Streams im Amazon DynamoDB](#) DynamoDB-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [ListStreams](#) Befehlsreferenz.AWS CLI

EC2 Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden EC2.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-address-transfer

Das folgende Codebeispiel zeigt die Verwendung `accept-address-transfer`.

AWS CLI

Um eine auf Ihr Konto übertragene Elastic IP-Adresse zu akzeptieren

Das folgende `accept-address-transfer` Beispiel akzeptiert die Übertragung der angegebenen Elastic IP-Adresse auf Ihr Konto.

```
aws ec2 accept-address-transfer \
  --address 100.21.184.216
```

Ausgabe:

```
{
  "AddressTransfer": {
    "PublicIp": "100.21.184.216",
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",
    "TransferAccountId": "123456789012",
    "TransferOfferExpirationTimestamp": "2023-02-22T20:51:10.000Z",
    "TransferOfferAcceptedTimestamp": "2023-02-22T22:52:54.000Z",
    "AddressTransferStatus": "accepted"
  }
}
```

Weitere Informationen finden Sie unter [Transfer Elastic IP-Adressen](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AcceptAddressTransfer](#) in der AWS CLI Befehlsreferenz.

accept-reserved-instances-exchange-quote

Das folgende Codebeispiel zeigt die Verwendung `accept-reserved-instances-exchange-quote`.

AWS CLI

Um einen Convertible Reserved Instance-Austausch durchzuführen

In diesem Beispiel wird ein Austausch der angegebenen Convertible Reserved Instances durchgeführt.

Befehl:

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

Ausgabe:

```
{
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"
}
```

- API-Einheiten finden Sie [AcceptReservedInstancesExchangeQuote](#) unter AWS CLI Befehlsreferenz.

accept-transit-gateway-peering-attachment

Das folgende Codebeispiel zeigt die Verwendung `accept-transit-gateway-peering-attachment`.

AWS CLI

Um einen Transit-Gateway-Peering-Anhang zu akzeptieren

Im folgenden `accept-transit-gateway-peering-attachment` Beispiel wird der angegebene Transit-Gateway-Peering-Anhang akzeptiert. Der `--region` Parameter gibt die Region an, in der sich das Transit-Gateway für den Akzeptierer befindet.

```
aws ec2 accept-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
  --region us-east-2
```

Ausgabe:

```
{
  "TransitGatewayPeeringAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
    "RequesterTgwInfo": {
      "TransitGatewayId": "tgw-123abc05e04123abc",
      "OwnerId": "123456789012",
      "Region": "us-west-2"
    },
    "AcceptorTgwInfo": {
      "TransitGatewayId": "tgw-11223344aabbcc112",
      "OwnerId": "123456789012",
      "Region": "us-east-2"
    },
    "State": "pending",
  }
}
```

```

    "CreationTime": "2019-12-09T11:38:31.000Z"
  }
}

```

Weitere Informationen finden Sie unter [Transit Gateway Peering Attachments](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [AcceptTransitGatewayPeeringAttachment AWS CLI-Befehlsreferenz](#).

accept-transit-gateway-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `accept-transit-gateway-vpc-attachment`.

AWS CLI

Um eine Anfrage zum Anhängen eines VPC an ein Transit-Gateway anzunehmen.

Das folgende `accept-transit-gateway-vpc-attachment` Beispiel akzeptiert die Anfrage für den angegebenen Anhang.

```

aws ec2 accept-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE

```

Ausgabe:

```

{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "VpcId": "vpc-07e8ffd50fEXAMPLE",
    "VpcOwnerId": "123456789012",
    "State": "pending",
    "SubnetIds": [
      "subnet-0752213d59EXAMPLE"
    ],
    "CreationTime": "2019-07-10T17:33:46.000Z",
    "Options": {
      "DnsSupport": "enable",
      "Ipv6Support": "disable"
    }
  }
}

```



```
}
```

Weitere Informationen finden Sie unter [Transit Gateway Attachments to a VPC](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [AcceptTransitGatewayVpcAttachment AWS CLI](#) Befehlsreferenz.

accept-vpc-endpoint-connections

Das folgende Codebeispiel zeigt die Verwendung `accept-vpc-endpoint-connections`.

AWS CLI

Um eine Verbindungsanforderung für einen Schnittstellenendpunkt anzunehmen

In diesem Beispiel wird die angegebene Endpunkt-Verbindungsanforderung für den angegebenen Endpunktdienst akzeptiert.

Befehl:

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --  
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Ausgabe:

```
{  
  "Unsuccessful": []  
}
```

- API-Einheiten finden Sie [AcceptVpcEndpointConnections](#) in der AWS CLI Befehlsreferenz.

accept-vpc-peering-connection

Das folgende Codebeispiel zeigt die Verwendung `accept-vpc-peering-connection`.

AWS CLI

Um eine VPC Peering-Verbindung zu akzeptieren

In diesem Beispiel wird die angegebene VPC Peering-Verbindungsanforderung akzeptiert.

Befehl:

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Ausgabe:

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-44455566",
      "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-1a2b3c4d",
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-111abc45",
      "CidrBlock": "10.0.0.0/28"
    }
  }
}
```

- API-Einheiten finden Sie [AcceptVpcPeeringConnection](#) in der AWS CLI Befehlsreferenz.

advertise-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `advertise-byoip-cidr`.

AWS CLI

Um für einen Adressbereich zu werben

Im folgenden `advertise-byoip-cidr` Beispiel wird für den angegebenen öffentlichen IPv4 Adressbereich geworben.

```
aws ec2 advertise-byoip-cidr \
```

```
--cidr 203.0.113.25/24
```

Ausgabe:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    "State": "provisioned"
  }
}
```

- API-Einheiten finden Sie unter [AdvertiseByoipCidr AWS CLI-Befehlsreferenz](#).

allocate-address

Das folgende Codebeispiel zeigt die Verwendung `allocate-address`.

AWS CLI

Beispiel 1: So weisen Sie eine Elastic-IP-Adresse aus dem Adress-Pool von Amazon zu

Im folgenden `allocate-address`-Beispiel wird eine Elastic-IP-Adresse zugewiesen. Amazon EC2 wählt die Adresse aus dem Adresspool von Amazon aus.

```
aws ec2 allocate-address
```

Ausgabe:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-01435ba59eEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2",
  "Domain": "vpc"
}
```

Weitere Informationen finden Sie unter [Elastic IP-Adressen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: So weisen Sie eine Elastic-IP-Adresse zu und verknüpfen sie mit einer Netzwerkgranzgruppe

Im folgenden `allocate-address`-Beispiel wird eine Elastic-IP-Adresse zugewiesen und sie der angegebenen Netzwerkrenzgruppe zugeordnet.

```
aws ec2 allocate-address \  
  --network-border-group us-west-2-lax-1
```

Ausgabe:

```
{  
  "PublicIp": "70.224.234.241",  
  "AllocationId": "eipalloc-e03dd489ceEXAMPLE",  
  "PublicIpv4Pool": "amazon",  
  "NetworkBorderGroup": "us-west-2-lax-1",  
  "Domain": "vpc"  
}
```

Weitere Informationen finden Sie unter [Elastic IP-Adressen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 3: So weisen Sie eine Elastic-IP-Adresse aus einem Adress-Pool zu, der Ihnen gehört

Im folgenden `allocate-address`-Beispiel wird eine Elastic-IP-Adresse aus einem Adress-Pool zugewiesen, den Sie in Ihr Amazon-Web-Services-Konto eingebunden haben. Amazon EC2 wählt die Adresse aus dem Adresspool aus.

```
aws ec2 allocate-address \  
  --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

Ausgabe:

```
{  
  "AllocationId": "eipalloc-02463d08ceEXAMPLE",  
  "NetworkBorderGroup": "us-west-2",  
  "CustomerOwnedIp": "18.218.95.81",  
  "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",  
  "Domain": "vpc"  
  "NetworkBorderGroup": "us-west-2",  
}
```

Weitere Informationen finden Sie unter [Elastic IP-Adressen](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AllocateAddress](#) in der AWS CLI Befehlsreferenz.

allocate-hosts

Das folgende Codebeispiel zeigt die Verwendung `allocate-hosts`.

AWS CLI

Beispiel 1: Um einen Dedicated Host zuzuweisen

Im folgenden `allocate-hosts` Beispiel wird ein einzelner Dedicated Host in der `eu-west-1a` Availability Zone zugewiesen, auf dem Sie Instances starten können. `m5.large` Standardmäßig akzeptiert der Dedicated Host nur Starts von Ziel-Instances und unterstützt keine Host-Wiederherstellung.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --quantity 1
```

Ausgabe:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Beispiel 2: So weisen Sie einen Dedicated Host mit aktivierter Auto-Platzierung und Host-Wiederherstellung zu

Im folgenden `allocate-hosts` Beispiel wird ein einzelner Dedicated Host in der `eu-west-1a` Availability Zone zugewiesen, wobei Auto-Platzierung und Host-Wiederherstellung aktiviert sind.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --auto-placement on \  
  --host-recovery on \  
  --quantity 1
```

Ausgabe:

```
{
  "HostIds": [
    "h-07879acf49EXAMPLE"
  ]
}
```

Beispiel 3: Um einen Dedicated Host mit Tags zuzuweisen

Das folgende `allocate-hosts` Beispiel weist einen einzelnen Dedicated Host zu und wendet ein Tag mit einem Schlüssel mit dem Namen `purpose` und dem Wert `an. production` an.

```
aws ec2 allocate-hosts \
  --instance-type m5.large \
  --availability-zone eu-west-1a \
  --quantity 1 \
  --tag-specifications 'ResourceType=dedicated-
host,Tags={Key=purpose,Value=production}'
```

Ausgabe:

```
{
  "HostIds": [
    "h-07879acf49EXAMPLE"
  ]
}
```

Weitere Informationen finden Sie unter [Allocating Dedicated Hosts](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [AllocateHosts](#) in der AWS CLI Befehlsreferenz.

allocate-ipam-pool-cidr

Das folgende Codebeispiel zeigt die Verwendung `allocate-ipam-pool-cidr`.

AWS CLI

Um eine CIDR aus einem IPAM Pool zuzuweisen

Im folgenden `allocate-ipam-pool-cidr` Beispiel wird eine CIDR aus einem Pool zugewiesen. IPAM

(Linux):

```
aws ec2 allocate-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --netmask-length 24
```

(Windows):

```
aws ec2 allocate-ipam-pool-cidr ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --netmask-length 24
```

Ausgabe:

```
{  
  "IpamPoolAllocation": {  
    "Cidr": "10.0.0.0/24",  
    "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfb",  
    "ResourceType": "custom",  
    "ResourceOwner": "123456789012"  
  }  
}
```

Weitere Informationen finden Sie unter [Manuelles Zuweisen von a CIDR zu einem Pool, um IP-Adressraum zu reservieren](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AllocateIpamPoolCidr](#) in der AWS CLI Befehlsreferenz.

apply-security-groups-to-client-vpn-target-network

Das folgende Codebeispiel zeigt die Verwendung `apply-security-groups-to-client-vpn-target-network`.

AWS CLI

Um Sicherheitsgruppen auf ein Zielnetzwerk für einen VPN Client-Endpunkt anzuwenden

Im folgenden `apply-security-groups-to-client-vpn-target-network` Beispiel wird die Sicherheitsgruppe `sg-01f6e627a89f4db32` auf die Verbindung zwischen dem angegebenen Zielnetzwerk und dem VPN Client-Endpunkt angewendet.

```
aws ec2 apply-security-groups-to-client-vpn-target-network \
  --security-group-ids sg-01f6e627a89f4db32 \
  --vpc-id vpc-0e2110c2f324332e0 \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Ausgabe:

```
{
  "SecurityGroupIds": [
    "sg-01f6e627a89f4db32"
  ]
}
```

Weitere Informationen finden Sie unter [Zielnetzwerke](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [ApplySecurityGroupsToClientVpnTargetNetwork](#) in der AWS CLI Befehlsreferenz.

assign-ipv6-addresses

Das folgende Codebeispiel zeigt die Verwendung `assign-ipv6-addresses`.

AWS CLI

Um einer Netzwerkschnittstelle bestimmte IPv6-Adressen zuzuweisen

In diesem Beispiel werden die angegebenen IPv6-Adressen der angegebenen Netzwerkschnittstelle zugewiesen.

Befehl:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 2001:db8:1234:1a00:9691:9503:25ad:1761
```

Ausgabe:

```
{
  "AssignedIpv6Addresses": [
    "2001:db8:1234:1a00:3304:8879:34cf:4071",
    "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ]
}
```



```
],  
  "NetworkInterfaceId": "eni-38664473"  
}
```

Um IPv6 Adressen, die Amazon auswählt, einer Netzwerkschnittstelle zuzuweisen

In diesem Beispiel werden der angegebenen Netzwerkschnittstelle zwei IPv6 Adressen zugewiesen. Amazon weist diese IPv6 Adressen automatisch aus den verfügbaren IPv6 Adressen im IPv6 CIDR Blockbereich des Subnetzes zu.

Befehl:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-  
count 2
```

Ausgabe:

```
{  
  "AssignedIpv6Addresses": [  
    "2001:db8:1234:1a00:3304:8879:34cf:4071",  
    "2001:db8:1234:1a00:9691:9503:25ad:1761"  
  ],  
  "NetworkInterfaceId": "eni-38664473"  
}
```

- API-Einheiten finden Sie unter [AssignIpv6Addresses](#) in AWS CLI der Befehlsreferenz.

assign-private-ip-addresses

Das folgende Codebeispiel zeigt die Verwendung `assign-private-ip-addresses`.

AWS CLI

Um einer bestimmten sekundären privaten IP-Adresse eine Netzwerkschnittstelle zuzuweisen

In diesem Beispiel wird die angegebene sekundäre private IP-Adresse der angegebenen Netzwerkschnittstelle zugewiesen. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-  
ip-addresses 10.0.0.82
```

Um einer Netzwerkschnittstelle sekundäre private IP-Adressen zuzuweisen, die Amazon EC2 auswählt

In diesem Beispiel werden der angegebenen Netzwerkschnittstelle zwei sekundäre private IP-Adressen zugewiesen. Amazon weist diese IP-Adressen EC2 automatisch aus den verfügbaren IP-Adressen im CIDR Blockbereich des Subnetzes zu, mit dem die Netzwerkschnittstelle verknüpft ist. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-  
private-ip-address-count 2
```

- API-Einheiten finden Sie [AssignPrivateIpAddresses](#) in der AWS CLI Befehlsreferenz.

assign-private-nat-gateway-address

Das folgende Codebeispiel zeigt die Verwendung `assign-private-nat-gateway-address`.

AWS CLI

Um Ihrem privaten NAT Gateway private IP-Adressen zuzuweisen

Im folgenden `assign-private-nat-gateway-address` Beispiel werden dem angegebenen privaten NAT Gateway zwei private IP-Adressen zugewiesen.

```
aws ec2 assign-private-nat-gateway-address \  
--nat-gateway-id nat-1234567890abcdef0 \  
--private-ip-address-count 2
```

Ausgabe:

```
{  
  "NatGatewayId": "nat-1234567890abcdef0",  
  "NatGatewayAddresses": [  
    {
```

```

        "NetworkInterfaceId": "eni-0065a61b324d1897a",
        "IsPrimary": false,
        "Status": "assigning"
    },
    {
        "NetworkInterfaceId": "eni-0065a61b324d1897a",
        "IsPrimary": false,
        "Status": "assigning"
    }
]
}

```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AssignPrivateNatGatewayAddress](#) in der AWS CLI Befehlsreferenz.

associate-address

Das folgende Codebeispiel zeigt die Verwendung `associate-address`.

AWS CLI

Um Elastic IP-Adressen in EC2 -Classic zuzuordnen

In diesem Beispiel wird eine Elastic IP-Adresse einer Instanz in EC2 -Classic zugeordnet. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 associate-address --instance-id i-07ffe74c7330ebf53 --public-ip 198.51.100.0
```

Um eine Elastic IP-Adresse in EC2 - zuzuordnen VPC

In diesem Beispiel wird eine Elastic IP-Adresse einer Instance in a zugeordnetVPC.

Befehl:

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-id eipalloc-64d5890a
```

Ausgabe:

```
{
  "AssociationId": "eipassoc-2bebb745"
}
```

In diesem Beispiel wird eine Elastic-IP-Adresse mit einer Netzwerkschnittstelle verknüpft.

Befehl:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d
```

In diesem Beispiel wird eine Elastic IP mit einer privaten IP-Adresse verknüpft, die mit einer Netzwerkschnittstelle verbunden ist.

Befehl:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d --private-ip-address 10.0.0.85
```

- API-Einheiten finden Sie [AssociateAddress](#) in der AWS CLI Befehlsreferenz.

associate-client-vpn-target-network

Das folgende Codebeispiel zeigt die Verwendung `associate-client-vpn-target-network`.

AWS CLI

Um ein Zielnetzwerk einem VPN Client-Endpunkt zuzuordnen

Im folgenden `associate-client-vpn-target-network` Beispiel wird dem angegebenen VPN Client-Endpunkt ein Subnetz zugeordnet.

```
aws ec2 associate-client-vpn-target-network \
  --subnet-id subnet-0123456789abcabca \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Ausgabe:

```
{
```

```
"AssociationId": "cvpn-assoc-12312312312312312",
"Status": {
  "Code": "associating"
}
}
```

Weitere Informationen finden Sie unter [Zielnetzwerke](#) im AWS VPNClient-Administratorhandbuch.

- APIEinzelheiten finden Sie [AssociateClientVpnTargetNetwork](#) in der AWS CLI Befehlsreferenz.

associate-dhcp-options

Das folgende Codebeispiel zeigt die Verwendung `associate-dhcp-options`.

AWS CLI

Um einen DHCP Optionssatz mit Ihrem zu verknüpfen VPC

In diesem Beispiel wird der angegebene DHCP Optionssatz dem angegebenen zugewiesen VPC. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

Um den DHCP Standardoptionssatz mit Ihrem zu verknüpfen VPC

In diesem Beispiel wird der DHCP Standardoptionssatz den angegebenen Optionen zugeordnet VPC. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- APIEinzelheiten finden Sie [AssociateDhcpOptions](#) unter AWS CLI Befehlsreferenz.

associate-iam-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `associate-iam-instance-profile`.

AWS CLI

Um ein IAM Instanzprofil einer Instanz zuzuordnen

In diesem Beispiel wird ein IAM Instanzprofil `admin-role` mit dem Namen Instanz `i-123456789abcde123` verknüpft.

Befehl:

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-  
instance-profile Name=admin-role
```

Ausgabe:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-123456789abcde123",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0e7736511a163c209",  
    "IamInstanceProfile": {  
      "Id": "AIPAJBLK7RKJKWDXVHIEC",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"  
    }  
  }  
}
```

- API-Einheiten finden Sie [AssociateIamInstanceProfile](#) in der AWS CLI Befehlsreferenz.

associate-instance-event-window

Das folgende Codebeispiel zeigt die Verwendung `associate-instance-event-window`.

AWS CLI

Beispiel 1: Um eine oder mehrere Instanzen einem Ereignisfenster zuzuordnen

Im folgenden `associate-instance-event-window` Beispiel werden eine oder mehrere Instanzen einem Ereignisfenster zugeordnet.

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  \
```

```
--instance-event-window-id iew-0abcdef1234567890 \  
--association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Ausgabe:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-1234567890abcdef0",  
        "i-0598c7d356eba48d7"  
      ],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 2: So verknüpfen Sie Instance-Tags mit einem Ereignisfenster

Im folgenden `associate-instance-event-window` Beispiel werden Instanz-Tags einem Ereignisfenster zugeordnet. Geben Sie einen `instance-event-window-id` Parameter ein, um das Ereignisfenster zu spezifizieren. Um Instanzkennzeichnungen zuzuordnen, geben Sie den `association-target` Parameter und für den Parameterwert ein oder mehrere Tags an.

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

Ausgabe:

```
{  
  "InstanceEventWindow": {
```

```

    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [
        {
          "Key": "k2",
          "Value": "v2"
        },
        {
          "Key": "k1",
          "Value": "v1"
        }
      ],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 3: So verknüpfen Sie einen Dedicated Host mit einem Eventfenster

Im folgenden `associate-instance-event-window` Beispiel wird ein Dedicated Host einem Ereignisfenster zugeordnet. Geben Sie einen `instance-event-window-id` Parameter ein, um das Ereignisfenster zu spezifizieren. Um einen Dedicated Host zuzuordnen, geben Sie den `--association-target` Parameter und für die Parameterwerte einen oder mehrere Dedicated Host anIDs.

```

aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "DedicatedHostIds=h-029fa35a02b99801d"

```

Ausgabe:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",

```



```

    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": [
        "h-029fa35a02b99801d"
      ]
    },
    "State": "creating"
  }
}

```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

- API-Einheiten finden Sie [AssociateInstanceEventWindow](#) in der AWS CLI Befehlsreferenz.

associate-ipam-resource-discovery

Das folgende Codebeispiel zeigt die Verwendung `associate-ipam-resource-discovery`.

AWS CLI

Um eine Ressourcenerkennung mit einem zu verknüpfen IPAM

In diesem Beispiel sind Sie ein IPAM delegierter Administrator und ein Ressourcensuchvorgang wurde von einem anderen AWS Konto erstellt und für Sie freigegeben, sodass Sie die Ressource verwalten und überwachen können IPAM, die dem anderen Konto CIDRs gehört.

Hinweis

Um diese Anfrage abzuschließen, benötigen Sie die Resource Discovery-ID, die Sie erhalten können, [describe-ipam-resource-discoveries](#) und die IPAM ID, die Sie mit [describe-ipams](#) erhalten können. Die Resource Discovery, die Sie verknüpfen, muss zuerst mit Ihrem Konto über AWS RAM .The, die `--region` Sie eingeben, geteilt worden sein. Die von Ihnen eingegebene Heimatregion muss mit der Heimatregion der Region übereinstimmen, der IPAM Sie sie zuordnen.

Das folgende `associate-ipam-resource-discovery` Beispiel verknüpft eine Ressourcensuche mit einem IPAM

```
aws ec2 associate-ipam-resource-discovery \
```

```
--ipam-id ipam-005f921c17ebd5107 \
--ipam-resource-discovery-id ipam-res-disco-03e0406de76a044ee \
--tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-center,Value=cc123}]' \
--region us-east-1
```

Ausgabe:

```
{
  {
    "IpamResourceDiscoveryAssociation": {
      "OwnerId": "320805250157",
      "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
      "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-04382a6346357cf82",
      "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
      "IpamId": "ipam-005f921c17ebd5107",
      "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
      "IpamRegion": "us-east-1",
      "IsDefault": false,
      "ResourceDiscoveryStatus": "active",
      "State": "associate-in-progress",
      "Tags": []
    }
  }
}
```

Sobald Sie eine Ressourcenerkennung verknüpft haben, können Sie die IP-Adressen der Ressourcen überwachen und/oder verwalten, die von den anderen Konten erstellt wurden. Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

- API-Einheiten finden Sie [AssociateIpamResourceDiscovery](#) in der AWS CLI Befehlsreferenz.

associate-nat-gateway-address

Das folgende Codebeispiel zeigt die Verwendung `associate-nat-gateway-address`.

AWS CLI

Um eine Elastic IP-Adresse einem öffentlichen NAT Gateway zuzuordnen

Im folgenden `associate-nat-gateway-address` Beispiel wird die angegebene Elastic IP-Adresse dem angegebenen öffentlichen NAT Gateway zugeordnet. AWS weist automatisch eine sekundäre private IPv4 Adresse zu.

```
aws ec2 associate-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --allocation-ids eipalloc-0be6ecac95EXAMPLE
```

Ausgabe:

```
{  
  "NatGatewayId": "nat-1234567890abcdef0",  
  "NatGatewayAddresses": [  
    {  
      "AllocationId": "eipalloc-0be6ecac95EXAMPLE",  
      "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
      "IsPrimary": false,  
      "Status": "associating"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AssociateNatGatewayAddress](#) in der AWS CLI Befehlsreferenz.

associate-route-table

Das folgende Codebeispiel zeigt die Verwendung `associate-route-table`.

AWS CLI

Um eine Routing-Tabelle einem Subnetz zuzuordnen

In diesem Beispiel wird die angegebene Routing-Tabelle dem angegebenen Subnetz zugeordnet.

Befehl:

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-  
id subnet-9d4a7b6c
```

Ausgabe:

```
{
  "AssociationId": "rtbassoc-781d0d1a"
}
```

- API-Einheiten finden Sie unter [AssociateRouteTable AWS CLI Befehlsreferenz](#).

associate-subnet-cidr-block

Das folgende Codebeispiel zeigt die Verwendung `associate-subnet-cidr-block`.

AWS CLI

Um einen IPv6 CIDR Block einem Subnetz zuzuordnen

In diesem Beispiel wird ein IPv6 CIDR Block dem angegebenen Subnetz zugeordnet.

Befehl:

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-  
block 2001:db8:1234:1a00::/64
```

Ausgabe:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
    "AssociationId": "subnet-cidr-assoc-3aa54053",
    "Ipv6CidrBlockState": {
      "State": "associating"
    }
  }
}
```

- API-Einheiten finden Sie unter [AssociateSubnetCidrBlock AWS CLI Befehlsreferenz](#).

associate-transit-gateway-multicast-domain

Das folgende Codebeispiel zeigt die Verwendung `associate-transit-gateway-multicast-domain`.

AWS CLI

Um ein Transit-Gateway einer Multicast-Domäne zuzuordnen

Im folgenden `associate-transit-gateway-multicast-domain` Beispiel werden das angegebene Subnetz und die Anlage der angegebenen Multicast-Domäne zugeordnet.

```
aws ec2 associate-transit-gateway-multicast-domain \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
  --transit-gateway-attachment-id tgw-attach-028c1dd0f8f5cbe8e \
  --subnet-ids subnet-000de86e3b49c932a \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Ausgabe:

```
{
  "Associations": [
    {
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
      "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",
      "ResourceId": "vpc-01128d2c240c09bd5",
      "ResourceType": "vpc",
      "Subnets": [
        {
          "SubnetId": "subnet-000de86e3b49c932a",
          "State": "associating"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie im Transit [Gateways Guide unter Managing Multicast-Domains](#).

- API-Einheiten finden Sie [AssociateTransitGatewayMulticastDomain](#) in der AWS CLI Befehlsreferenz.

associate-transit-gateway-route-table

Das folgende Codebeispiel zeigt die Verwendung `associate-transit-gateway-route-table`.

AWS CLI

Um eine Transit-Gateway-Routentabelle einem Transit-Gateway-Anhang zuzuordnen

Im folgenden Beispiel wird die angegebene Transit-Gateway-Routentabelle der angegebenen VPC Anlage zugeordnet.

```
aws ec2 associate-transit-gateway-route-table \  
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Ausgabe:

```
{  
  "Association": {  
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",  
    "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
    "ResourceId": "vpc-0065acced4EXAMPLE",  
    "ResourceType": "vpc",  
    "State": "associating"  
  }  
}
```

Weitere Informationen finden Sie unter [Zuordnen einer Transit-Gateway-Routentabelle](#) im AWS Transit Gateways Guide.

- API-Einheiten finden Sie unter [AssociateTransitGatewayRouteTable AWS CLIBefehlsreferenz](#).

associate-vpc-cidr-block

Das folgende Codebeispiel zeigt die Verwendung `associate-vpc-cidr-block`.

AWS CLI

Beispiel 1: Um einen von Amazon bereitgestellten IPv6 CIDR Block mit einem zu verknüpfen VPC

Das folgende `associate-vpc-cidr-block` Beispiel verknüpft einen IPv6 CIDR Block mit dem angegebenen VPC :

```
aws ec2 associate-vpc-cidr-block \  
  --amazon-provided-ipv6-cidr-block \  
  --ipv6-cidr-block-network-border-group us-west-2-lax-1 \  
  --vpc-id vpc-8EXAMPLE
```

Ausgabe:

```
{
  "Ipv6CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",
    "Ipv6CidrBlockState": {
      "State": "associating"
    },
    "NetworkBorderGroup": "us-west-2-lax-1"
  },
  "VpcId": "vpc-8EXAMPLE"
}
```

Beispiel 2: Um einen zusätzlichen IPv4 CIDR Block mit einem zu verknüpfen VPC

Das folgende `associate-vpc-cidr-block` Beispiel verknüpft den IPv4 CIDR Block `10.2.0.0/16` mit dem angegebenen VPC.

```
aws ec2 associate-vpc-cidr-block \
  --vpc-id vpc-1EXAMPLE \
  --cidr-block 10.2.0.0/16
```

Ausgabe:

```
{
  "CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-2EXAMPLE",
    "CidrBlock": "10.2.0.0/16",
    "CidrBlockState": {
      "State": "associating"
    }
  },
  "VpcId": "vpc-1EXAMPLE"
}
```

- API Einheiten finden Sie [AssociateVpcCidrBlock](#) unter AWS CLI Befehlsreferenz.

attach-classic-link-vpc

Das folgende Codebeispiel zeigt die Verwendung `attach-classic-link-vpc`.

AWS CLI

Um eine EC2 -Classic-Instanz mit einem zu verknüpfen (anzuhängen) VPC

In diesem Beispiel wird die Instanz `i-1234567890abcdef0` über die Sicherheitsgruppe `sg-12312312` mit `vpc-88888888` verknüpft. VPC VPC

Befehl:

```
aws ec2 attach-classic-link-vpc --instance-id i-1234567890abcdef0 --vpc-id vpc-88888888 --groups sg-12312312
```

Ausgabe:

```
{  
  "Return": true  
}
```

- APIEinzelheiten [AttachClassicLinkVpc](#) finden Sie unter AWS CLI Befehlsreferenz.

attach-internet-gateway

Das folgende Codebeispiel zeigt die Verwendung `attach-internet-gateway`.

AWS CLI

Um ein Internet-Gateway an Ihr anzuschließen VPC

Im folgenden `attach-internet-gateway` Beispiel wird das angegebene Internet-Gateway an das spezifische VPC angehängt.

```
aws ec2 attach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Internet-Gateways](#) im VPC Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AttachInternetGateway](#) in der AWS CLI Befehlsreferenz.

attach-network-interface

Das folgende Codebeispiel zeigt die Verwendung `attach-network-interface`.

AWS CLI

Beispiel 1: Um eine Netzwerkschnittstelle an eine Instanz anzuhängen

Im folgenden `attach-network-interface` Beispiel wird die angegebene Netzwerkschnittstelle an die angegebene Instanz angehängt.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-0dc56a8d4640ad10a \  
  --instance-id i-1234567890abcdef0 \  
  --device-index 1
```

Ausgabe:

```
{  
  "AttachmentId": "eni-attach-01a8fc87363f07cf9"  
}
```

Weitere Informationen finden Sie unter [Elastic Network Interfaces](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Um eine Netzwerkschnittstelle an eine Instance mit mehreren Netzwerkkarten anzuschließen

Im folgenden `attach-network-interface` Beispiel wird die angegebene Netzwerkschnittstelle an die angegebene Instanz und Netzwerkkarte angehängt.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-07483b1897541ad83 \  
  --instance-id i-01234567890abcdef \  
  --network-card-index 1 \  
  --device-index 1
```

Ausgabe:

```
{
```

```
"AttachmentId": "eni-attach-0fbd7ee87a88cd06c"
}
```

Weitere Informationen finden Sie unter [Elastic Network Interfaces](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AttachNetworkInterface](#) in der AWS CLI Befehlsreferenz.

attach-verified-access-trust-provider

Das folgende Codebeispiel zeigt die Verwendung `attach-verified-access-trust-provider`.

AWS CLI

Um einen Vertrauensanbieter an eine Instanz anzuhängen

Im folgenden `attach-verified-access-trust-provider` Beispiel wird der angegebene Verified Access-Vertrauensanbieter an die angegebene Verified Access-Instanz angehängt.

```
aws ec2 attach-verified-access-trust-provider \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Ausgabe:

```
{
  "VerifiedAccessTrustProvider": {
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
    "Description": "",
    "TrustProviderType": "user",
    "UserTrustProviderType": "iam-identity-center",
    "PolicyReferenceName": "idc",
    "CreationTime": "2023-08-25T19:00:38",
    "LastUpdatedTime": "2023-08-25T19:00:38"
  },
  "VerifiedAccessInstance": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "",
    "VerifiedAccessTrustProviders": [
      {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
```

```
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center"
    }
],
"CreationTime": "2023-08-25T18:27:56",
"LastUpdatedTime": "2023-08-25T18:27:56"
}
}
```

Weitere Informationen finden Sie unter [Verified Access-Instanzen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [AttachVerifiedAccessTrustProvider](#) in der AWS CLI Befehlsreferenz.

attach-volume

Das folgende Codebeispiel zeigt die Verwendung `attach-volume`.

AWS CLI

Um ein Volume an eine Instance anzuhängen

Mit diesem Beispielbefehl wird ein Volume (`vol-1234567890abcdef0`) an eine Instanz (`i-01474ef662b89480`) angehängt als `/dev/sdf`.

Befehl:

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-id i-01474ef662b89480 --device /dev/sdf
```

Ausgabe:

```
{
  "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
  "InstanceId": "i-01474ef662b89480",
  "VolumeId": "vol-1234567890abcdef0",
  "State": "attaching",
  "Device": "/dev/sdf"
}
```

- API-Einheiten finden Sie [AttachVolume](#) in der AWS CLI Befehlsreferenz.

attach-vpn-gateway

Das folgende Codebeispiel zeigt die Verwendung `attach-vpn-gateway`.

AWS CLI

Um ein virtuelles privates Gateway an Ihr anzuschließen VPC

Im folgenden `attach-vpn-gateway` Beispiel wird das angegebene virtuelle private Gateway an das angegebene VPC angehängt.

```
aws ec2 attach-vpn-gateway \  
  --vpn-gateway-id vgw-9a4cacf3 \  
  --vpc-id vpc-a01106c2
```

Ausgabe:

```
{  
  "VpcAttachment": {  
    "State": "attaching",  
    "VpcId": "vpc-a01106c2"  
  }  
}
```

- API-Einheiten finden Sie unter [AttachVpnGateway AWS CLI Befehlsreferenz](#).

authorize-client-vpn-ingress

Das folgende Codebeispiel zeigt die Verwendung `authorize-client-vpn-ingress`.

AWS CLI

Um eine Autorisierungsregel für einen VPN Client-Endpunkt hinzuzufügen

Im folgenden `authorize-client-vpn-ingress` Beispiel wird eine Autorisierungsregel für eingehende Zugriffe hinzugefügt, die allen Clients den Zugriff auf das Internet ermöglicht (`0.0.0.0/0`).

```
aws ec2 authorize-client-vpn-ingress \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --source-cidr-blocks 0.0.0.0/0
```

```
--target-network-cidr 0.0.0.0/0 \  
--authorize-all-groups
```

Ausgabe:

```
{  
  "Status": {  
    "Code": "authorizing"  
  }  
}
```

Weitere Informationen finden Sie unter [Autorisierungsregeln](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [AuthorizeClientVpnIngress](#) in der AWS CLI Befehlsreferenz.

authorize-security-group-egress

Das folgende Codebeispiel zeigt die Verwendung `authorize-security-group-egress`.

AWS CLI

Um eine Regel hinzuzufügen, die ausgehenden Verkehr in einen bestimmten Adressbereich zulässt

Dieser Beispielbefehl fügt eine Regel hinzu, die Zugriff auf die angegebenen Adressbereiche an TCP Port 80 gewährt.

Befehl (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions  
IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges='[{"CidrIp=10.0.0.0/16}]'
```

Befehl (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-  
permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{"CidrIp=10.0.0.0/16}]
```

Um eine Regel hinzuzufügen, die ausgehenden Datenverkehr zu einer bestimmten Sicherheitsgruppe zulässt

Dieser Beispielbefehl fügt eine Regel hinzu, die Zugriff auf die angegebene Sicherheitsgruppe an TCP Port 80 gewährt.

Befehl (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions
IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=' [{GroupId=sg-4b51a32f}]'
```

Befehl (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-
permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-4b51a32f}]
```

- API-Einheiten finden Sie [AuthorizeSecurityGroupEgress](#) in der AWS CLI Befehlsreferenz.

authorize-security-group-ingress

Das folgende Codebeispiel zeigt die Verwendung `authorize-security-group-ingress`.

AWS CLI

Beispiel 1: Um eine Regel hinzuzufügen, die eingehenden Datenverkehr SSH zulässt

Das folgende `authorize-security-group-ingress` Beispiel fügt eine Regel hinzu, die eingehenden Verkehr auf TCP Port 22 () SSH zulässt.

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-1234567890abcdef0 \
  --protocol tcp \
  --port 22 \
  --cidr 203.0.113.0/24
```

Ausgabe:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-01afa97ef3e1bedfc",
      "GroupId": "sg-1234567890abcdef0",
```

```

        "GroupOwnerId": "123456789012",
        "IsEgress": false,
        "IpProtocol": "tcp",
        "FromPort": 22,
        "ToPort": 22,
        "CidrIpv4": "203.0.113.0/24"
    }
]
}

```

Beispiel 2: Um eine Regel hinzuzufügen, die eingehenden HTTP Verkehr von einer anderen Sicherheitsgruppe zulässt

Im folgenden `authorize-security-group-ingress` Beispiel wird eine Regel hinzugefügt, die eingehenden Zugriff auf TCP Port 80 von der Quellsicherheitsgruppe aus ermöglicht. `sg-1a2b3c4d` Die Quellgruppe muss sich in derselben Gruppe VPC oder in einem Peer befinden VPC (erfordert eine VPC Peering-Verbindung). Eingehender Datenverkehr ist basierend auf den privaten IP-Adressen der Instances erlaubt, die der Quellsicherheitsgruppe zugeordnet sind (nicht die öffentliche IP-Adresse oder die Elastic-IP-Adresse).

```

aws ec2 authorize-security-group-ingress \
  --group-id sg-1234567890abcdef0 \
  --protocol tcp \
  --port 80 \
  --source-group sg-1a2b3c4d

```

Ausgabe:

```

{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-01f4be99110f638a7",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "ReferencedGroupInfo": {
        "GroupId": "sg-1a2b3c4d",
        "UserId": "123456789012"
      }
    }
  ]
}

```

```

    }
  }
]
}

```

Beispiel 3: So fügen Sie mehrere Regeln im selben Aufruf hinzu

Im folgenden `authorize-security-group-ingress` Beispiel werden mithilfe des `ip-permissions` Parameters zwei Regeln für eingehenden Datenverkehr hinzugefügt, von denen eine den eingehenden Zugriff auf TCP Port 3389 (RDP) und die andere, die ping/ aktiviert. ICMP

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol =tcp, =3389, FromPort ToPort =3389, IpRanges = "[{CidrIp=172.31.0.0/16}]" IpProtocol
=icmp, =-1, FromPort ToPort =-1, IpRanges = "[{CidrIp=172.31.0.0/16}]"
```

Ausgabe:

```

{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupId": "sgr-00e06e5d3690f29f3",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3389,
      "ToPort": 3389,
      "CidrIpv4": "172.31.0.0/16"
    },
    {
      "SecurityGroupId": "sgr-0a133dd4493944b87",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv4": "172.31.0.0/16"
    }
  ]
}

```


Beispiel 4: Um eine Regel ICMP für den Datenverkehr hinzuzufügen

Im folgenden `authorize-security-group-ingress` Beispiel wird der `ip-permissions` Parameter verwendet, um eine Regel für eingehende Nachrichten hinzuzufügen, die die ICMP Nachricht `Destination Unreachable: Fragmentation Needed and Don't Fragment` was Set (Typ 3, Code 4) von überall aus zulässt.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=icmp, =3, =4, = [{"=0.0.0.0/0}] IpProtocol" FromPort ToPort IpRanges CidrIp
```

Ausgabe:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0de3811019069b787",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmp",
      "FromPort": 3,
      "ToPort": 4,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

Beispiel 5: Um eine Regel für den Verkehr IPv6 hinzuzufügen

Im folgenden `authorize-security-group-ingress` Beispiel wird der `ip-permissions` Parameter verwendet, um eine Regel für eingehenden Datenverkehr hinzuzufügen, die SSH den Zugriff (Port 22) aus dem IPv6 Bereich `2001:db8:1234:1a00::/64` ermöglicht.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol =tcp, =22, =22, Ipv6Ranges= [{"=2001:db 8:1234:1 a00: :/64}]" FromPort ToPort
CidrIpv6
```

Ausgabe:

```
{
  "Return": true,
```

```

"SecurityGroupRules": [
  {
    "SecurityGroupRuleId": "sgr-0455bc68b60805563",
    "GroupId": "sg-1234567890abcdef0",
    "GroupOwnerId": "123456789012",
    "IsEgress": false,
    "IpProtocol": "tcp",
    "FromPort": 22,
    "ToPort": 22,
    "CidrIpv6": "2001:db8:1234:1a00::/64"
  }
]
}

```

Beispiel 6: Um eine Regel für den Verkehr hinzuzufügen ICMPv6

Im folgenden `authorize-security-group-ingress` Beispiel wird der `ip-permissions` Parameter verwendet, um eine Regel für eingehenden Datenverkehr hinzuzufügen, die ICMPv6 Datenverkehr von überall her zulässt.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=icmpv6, Ipv6Ranges=["[{6=: :/0}]" IpProtocol CidrIpv
```

Ausgabe:

```

{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmpv6",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv6": "::<0"
    }
  ]
}

```

Beispiel 7: Eine Regel mit einer Beschreibung hinzuzufügen

Im folgenden `authorize-security-group-ingress` Beispiel wird der Parameter verwendet, um eine Regel für `ip-permissions` eingehenden Datenverkehr hinzuzufügen, die Datenverkehr aus dem angegebenen Adressbereich zulässt RDP. IPv4 Die Regel enthält eine Beschreibung, die Ihnen später hilft, sie zu identifizieren.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol=tcp,FromPort=3389,ToPort=3389,CidrIpRanges=["203.0.113.0/24",
Description='Zugriff von einer Niederlassung in New York aus']
```

Ausgabe:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0397bbcc01e974db3",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3389,
      "ToPort": 3389,
      "CidrIpv4": "203.0.113.0/24",
      "Description": "RDP access from NY office"
    }
  ]
}
```

Beispiel 8: So fügen Sie eine eingehende Regel hinzu, die eine Präfixliste verwendet

Im folgenden `authorize-security-group-ingress` Beispiel wird der Parameter verwendet, um eine Regel für `ip-permissions` eingehenden Datenverkehr hinzuzufügen, die den gesamten Verkehr für die Bereiche in der angegebenen CIDR Präfixliste zulässt.

```
aws ec2 authorize-security-group-ingress --group-id sg-04a351bfe432d4e71 --ip-permissions=all,
PrefixListIds=["pl-002dc3ec097de1514"]
```

Ausgabe:

```
{
  "Return": true,
```

```
"SecurityGroupRules": [  
  {  
    "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",  
    "GroupId": "sg-1234567890abcdef0",  
    "GroupOwnerId": "123456789012",  
    "IsEgress": false,  
    "IpProtocol": "-1",  
    "FromPort": -1,  
    "ToPort": -1,  
    "PrefixListId": "pl-0721453c7ac4ec009"  
  }  
]  
}
```

Weitere Informationen finden Sie unter [Sicherheitsgruppen](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AuthorizeSecurityGroupIngress](#) in der AWS CLI Befehlsreferenz.

bundle-instance

Das folgende Codebeispiel zeigt die Verwendung `bundle-instance`.

AWS CLI

Um eine Instanz zu bündeln

In diesem Beispiel wird die Instanz `i-1234567890abcdef0` zu einem Bucket namens `bundletasks` gebündelt. Bevor Sie Werte für Ihren Zugriffsschlüssel angeben IDs, lesen und befolgen Sie die Anleitungen unter [Bewährte Methoden für die Verwaltung von AWS Zugriffsschlüsseln](#).

Befehl:

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --  
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

Ausgabe:

```
{  
  "BundleTask": {  
    "UpdateTime": "2015-09-15T13:30:35.000Z",  
    "InstanceId": "i-1234567890abcdef0",  
  }  
}
```

```
"Storage": {
  "S3": {
    "Prefix": "winami",
    "Bucket": "bundletasks"
  }
},
"State": "pending",
"StartTime": "2015-09-15T13:30:35.000Z",
"BundleId": "bun-294e041f"
}
}
```

- API-Einheiten finden Sie [BundleInstance](#) in der AWS CLI Befehlsreferenz.

cancel-bundle-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-bundle-task`.

AWS CLI

Um eine Bundle-Aufgabe abubrechen

In diesem Beispiel wird die Bundle-Aufgabe `bun-2a4e041c` storniert.

Befehl:

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

Ausgabe:

```
{
  "BundleTask": {
    "UpdateTime": "2015-09-15T13:27:40.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "Storage": {
      "S3": {
        "Prefix": "winami",
        "Bucket": "bundletasks"
      }
    },
    "State": "cancelling",
    "StartTime": "2015-09-15T13:24:35.000Z",
```

```
"BundleId": "bun-2a4e041c"
}
}
```

- API-Einheiten finden Sie [CancelBundleTask](#) in der AWS CLI Befehlsreferenz.

cancel-capacity-reservation-fleets

Das folgende Codebeispiel zeigt die Verwendung `cancel-capacity-reservation-fleets`.

AWS CLI

Um eine Flotte mit Kapazitätsreservierungen zu stornieren

Im folgenden `cancel-capacity-reservation-fleets` Beispiel werden die angegebene Kapazitätsreservierungsflotte und die von ihr reservierte Kapazität storniert. Wenn Sie eine Flotte stornieren, ändert sich ihr Status in `cancelled` und es können keine neuen Kapazitätsreservierungen mehr erstellt werden. Darüber hinaus werden alle einzelnen Kapazitätsreservierungen in der Flotte storniert, und die Instances, die zuvor in der reservierten Kapazität ausgeführt wurden, werden weiterhin normal in gemeinsam genutzter Kapazität ausgeführt.

```
aws ec2 cancel-capacity-reservation-fleets \
  --capacity-reservation-fleet-ids crf-abcdef01234567890
```

Ausgabe:

```
{
  "SuccessfulFleetCancellations": [
    {
      "CurrentFleetState": "cancelling",
      "PreviousFleetState": "active",
      "CapacityReservationFleetId": "crf-abcdef01234567890"
    }
  ],
  "FailedFleetCancellations": []
}
```

Weitere Informationen zu Kapazitätsreservierungsflotten finden Sie unter [Kapazitätsreservierungsflotten](#) im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CancelCapacityReservationFleets AWS CLI Befehlsreferenz](#).

cancel-capacity-reservation

Das folgende Codebeispiel zeigt die Verwendung `cancel-capacity-reservation`.

AWS CLI

Um eine Kapazitätsreservierung zu stornieren

Im folgenden `cancel-capacity-reservation` Beispiel wird die angegebene Kapazitätsreservierung storniert.

```
aws ec2 cancel-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie unter [Stornieren einer Kapazitätsreservierung](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie unter [CancelCapacityReservation AWS CLI Befehlsreferenz](#).

cancel-conversion-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-conversion-task`.

AWS CLI

Um eine aktive Konvertierung einer Instance oder eines Volumes abubrechen

In diesem Beispiel wird der mit der Task-ID `import-i-fh95npoc` verknüpfte Upload abgebrochen. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- API-Einheiten finden Sie unter [CancelConversionTask](#) Befehlsreferenz. AWS CLI

cancel-export-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-export-task`.

AWS CLI

Um eine aktive Exportaufgabe abubrechen

In diesem Beispiel wird eine aktive Exportaufgabe mit der Task-ID `export-i-fgelt0i7` storniert. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- API-Einheiten finden Sie unter Befehlsreferenz [CancelExportTask](#). AWS CLI

cancel-image-launch-permission

Das folgende Codebeispiel zeigt die Verwendung `cancel-image-launch-permission`.

AWS CLI

Um zu kündigen, dass ein Konto mit Ihrem Amazon Web Services AMI geteilt wurde

Im folgenden `cancel-image-launch-permission` Beispiel werden Ihrem Konto die Startberechtigungen für AMI das angegebene Konto entzogen.

```
aws ec2 cancel-image-launch-permission \  
  --image-id ami-0123456789example \  
  --region us-east-1
```

Ausgabe:

```
{  
  "Return": true  
}
```


Weitere Informationen finden Sie unter [Kündigen eines mit Ihrem Amazon Web Services Services-Konto AMI geteilten Kontos](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CancelImageLaunchPermission](#) unter AWS CLI Befehlsreferenz.

cancel-import-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-import-task`.

AWS CLI

Um eine Importaufgabe abubrechen

Im folgenden `cancel-import-task` Beispiel wird die angegebene Aufgabe zum Importieren eines Bilds abgebrochen.

```
aws ec2 cancel-import-task \  
  --import-task-id import-ami-1234567890abcdef0
```

Ausgabe:

```
{  
  "ImportTaskId": "import-ami-1234567890abcdef0",  
  "PreviousState": "active",  
  "State": "deleting"  
}
```

- APIEinzelheiten finden Sie [CancelImportTask](#) in der AWS CLI Befehlsreferenz.

cancel-reserved-instances-listing

Das folgende Codebeispiel zeigt die Verwendung `cancel-reserved-instances-listing`.

AWS CLI

Um ein Reserved Instance-Angebot zu stornieren

Im folgenden `cancel-reserved-instances-listing` Beispiel wird die angegebene Reserved Instance-Liste storniert.

```
aws ec2 cancel-reserved-instances-listing \  
  --reserved-instances-listing-id ri-1234567890abcdef0
```

```
--reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- API-Einheiten finden Sie unter [CancelReservedInstancesListing AWS CLI Befehlsreferenz](#).

cancel-spot-fleet-requests

Das folgende Codebeispiel zeigt die Verwendung `cancel-spot-fleet-requests`.

AWS CLI

Beispiel 1: Um eine Spot-Flottenanfrage zu stornieren und die zugehörigen Instances zu beenden

Im folgenden `cancel-spot-fleet-requests` Beispiel wird eine Spot-Flottenanfrage storniert und die zugehörigen On-Demand-Instances und Spot-Instances beendet.

```
aws ec2 cancel-spot-fleet-requests \
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --terminate-instances
```

Ausgabe:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_terminating",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

Weitere Informationen finden Sie unter [Stornieren einer Spot-Flottenanfrage](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: Um eine Spot-Flottenanfrage zu stornieren, ohne die zugehörigen Instances zu beenden

Im folgenden `cancel-spot-fleet-requests` Beispiel wird eine Spot-Flottenanforderung storniert, ohne die zugehörigen On-Demand-Instances und Spot-Instances zu beenden.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --no-terminate-instances
```

Ausgabe:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_running",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

Weitere Informationen finden Sie unter [Stornieren einer Spot-Flottenanfrage](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [CancelSpotFleetRequests](#) in der AWS CLI Befehlsreferenz.

cancel-spot-instance-requests

Das folgende Codebeispiel zeigt die Verwendung `cancel-spot-instance-requests`.

AWS CLI

Um Spot-Instance-Anfragen zu stornieren

Dieser Beispielbefehl storniert eine Spot-Instance-Anfrage.

Befehl:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Ausgabe:

```
{  
  "CancelledSpotInstanceRequests": [  
    {  
      "SpotInstanceRequestId": "sir-08b93456",  
      "CurrentSpotInstanceRequestState": "cancelled_running",  
      "PreviousSpotInstanceRequestState": "active"  
    }  
  ]  
}
```

```
{
  "State": "cancelled",
  "SpotInstanceRequestId": "sir-08b93456"
}
]
```

- API-Einheiten finden Sie [CancelSpotInstanceRequests](#) in der AWS CLI Befehlsreferenz.

confirm-product-instance

Das folgende Codebeispiel zeigt die Verwendung `confirm-product-instance`.

AWS CLI

Um die Produktinstanz zu bestätigen

In diesem Beispiel wird ermittelt, ob der angegebene Produktcode der angegebenen Instanz zugeordnet ist.

Befehl:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-id i-1234567890abcdef0
```

Ausgabe:

```
{
  "OwnerId": "123456789012"
}
```

- API-Einheiten finden Sie [ConfirmProductInstance](#) in der AWS CLI Befehlsreferenz.

copy-fpga-image

Das folgende Codebeispiel zeigt die Verwendung `copy-fpga-image`.

AWS CLI

Um ein FPGA Amazon-Bild zu kopieren

In diesem Beispiel wird der angegebene AFI Wert aus der `us-east-1` Region in die aktuelle Region (`eu-west-1`) kopiert.

Befehl:

```
aws ec2 copy-fpga-image --name copy-afi --source-fpga-image-id afi-0d123e123bfc85abc
--source-region us-east-1 --region eu-west-1
```

Ausgabe:

```
{
  "FpgaImageId": "afi-06b12350a123fbabc"
}
```

- API-Einheiten finden Sie [CopyFpgaImage](#) unter AWS CLI Befehlsreferenz.

copy-image

Das folgende Codebeispiel zeigt die Verwendung `copy-image`.

AWS CLI

Beispiel 1: Um eine in eine andere Region AMI zu kopieren

Der folgende `copy-image` Beispielbefehl kopiert die angegebene Datei AMI aus der `us-west-2` Region in die `us-east-1` Region und fügt eine kurze Beschreibung hinzu.

```
aws ec2 copy-image \
  --region us-east-1 \
  --name ami-name \
  --source-region us-west-2 \
  --source-image-id ami-066877671789bd71b \
  --description "This is my copied image."
```

Ausgabe:

```
{
  "ImageId": "ami-0123456789abcdefg"
}
```

Weitere Informationen finden Sie unter [Kopieren einer AMI](#) im EC2 Amazon-Benutzerhandbuch.

Beispiel 2: Um einen Snapshot in eine andere Region AMI zu kopieren und den Backing-Snapshot zu verschlüsseln

Der folgende `copy-image` Befehl kopiert den angegebenen Wert AMI aus der `us-west-2` Region in die aktuelle Region und verschlüsselt den Backing-Snapshot mit dem angegebenen KMS Schlüssel.

```
aws ec2 copy-image \  
  --source-region us-west-2 \  
  --name ami-name \  
  --source-image-id ami-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

Ausgabe:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Weitere Informationen finden Sie unter [Kopieren einer AMI](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 3: Um Ihre benutzerdefinierten AMI Tags beim Kopieren eines einzubeziehen AMI

Der folgende `copy-image` Befehl verwendet den `--copy-image-tags` Parameter, um Ihre benutzerdefinierten AMI Tags beim Kopieren von zu kopieren. AMI

```
aws ec2 copy-image \  
  --region us-east-1 \  
  --name ami-name \  
  --source-region us-west-2 \  
  --source-image-id ami-066877671789bd71b \  
  --description "This is my copied image." \  
  --copy-image-tags
```

Ausgabe:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Weitere Informationen finden Sie unter [Kopieren einer AMI](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CopyImage](#) in der AWS CLI Befehlsreferenz.

copy-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-snapshot`.

AWS CLI

Beispiel 1: Um einen Snapshot in eine andere Region zu kopieren

Der folgende `copy-snapshot` Beispielbefehl kopiert den angegebenen Snapshot von der `us-west-2` Region in die `us-east-1` Region und fügt eine kurze Beschreibung hinzu.

```
aws ec2 copy-snapshot \  
  --region us-east-1 \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --description "This is my copied snapshot."
```

Ausgabe:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Weitere Informationen finden Sie unter [Kopieren eines EBS Amazon-Snapshots](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Um einen unverschlüsselten Snapshot zu kopieren und den neuen Snapshot zu verschlüsseln

Der folgende `copy-snapshot` Befehl kopiert den angegebenen unverschlüsselten Snapshot aus der `us-west-2` Region in die aktuelle Region und verschlüsselt den neuen Snapshot mit dem angegebenen Schlüssel. KMS

```
aws ec2 copy-snapshot \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --encrypted \  
  --kms-key-id arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
```

```
--kms-key-id alias/my-kms-key
```

Ausgabe:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Weitere Informationen finden Sie unter [Kopieren eines EBS Amazon-Snapshots](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CopySnapshot](#) in der AWS CLI Befehlsreferenz.

create-capacity-reservation-fleet

Das folgende Codebeispiel zeigt die Verwendung `create-capacity-reservation-fleet`.

AWS CLI

Um eine Flotte für Kapazitätsreservierungen zu erstellen

Im folgenden `create-capacity-reservation-fleet` Beispiel wird eine Flotte für Kapazitätsreservierungen für den in der Anfrage angegebenen Instance-Typ bis zur angegebenen Gesamtzielkapazität erstellt. Die Anzahl der Instances, für die die Kapazitätsreservierungsflotte Kapazität reserviert, hängt von der Gesamtzielkapazität und den Instance-Typ-Gewichtungen ab, die Sie in der Anforderung angeben. Geben Sie die zu verwendenden Instance-Typen und eine Priorität für jeden der angegebenen Instance-Typen an.

```
aws ec2 create-capacity-reservation-fleet \  
--total-target-capacity 24 \  
--allocation-strategy prioritized \  
--instance-match-criteria open \  
--tenancy default \  
--end-date 2022-12-31T23:59:59.000Z \  
--instance-type-specifications file://instanceTypeSpecification.json
```

Inhalt von `instanceTypeSpecification.json`:

```
[  
  {  
    "InstanceType": "m5.xlarge",
```



```
    "InstancePlatform": "Linux/UNIX",
    "Weight": 3.0,
    "AvailabilityZone": "us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

Ausgabe:

```
{
  "Status": "submitted",
  "TotalFulfilledCapacity": 0.0,
  "CapacityReservationFleetId": "crf-abcdef01234567890",
  "TotalTargetCapacity": 24
}
```

Weitere Informationen zu Kapazitätsreservierungsflotten finden Sie unter [Kapazitätsreservierungsflotten](#) im EC2Amazon-Benutzerhandbuch.

Weitere Informationen zum Gewicht des Instance-Typs und zur Gesamtzielkapazität finden Sie unter [Instance-Typgewicht](#) und [Gesamtzielkapazität](#) im EC2Amazon-Benutzerhandbuch.

Weitere Informationen zur Festlegung von Prioritäten für bestimmte Instance-Typen finden Sie unter [Zuweisungsstrategie](#) und [Instance-Typpriorität](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCapacityReservationFleet](#) in der AWS CLI Befehlsreferenz.

create-capacity-reservation

Das folgende Codebeispiel zeigt die Verwendung `create-capacity-reservation`.

AWS CLI

Beispiel 1: Um eine Kapazitätsreservierung zu erstellen

Im folgenden `create-capacity-reservation` Beispiel wird eine Kapazitätsreservierung in der `eu-west-1a` Availability Zone erstellt, in der Sie drei `t2.medium` Instances starten können, auf denen ein Linux/Unix-Betriebssystem ausgeführt wird. Standardmäßig wird die Kapazitätsreservierung mit offenen Instance-Übereinstimmungskriterien und ohne Unterstützung für kurzlebigen Speicher erstellt. Sie bleibt aktiv, bis Sie sie manuell stornieren.

```
aws ec2 create-capacity-reservation \  
  --availability-zone eu-west-1a \  
  --instance-type t2.medium \  
  --instance-platform Linux/UNIX \  
  --instance-count 3
```

Ausgabe:

```
{  
  "CapacityReservation": {  
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",  
    "EndDateType": "unlimited",  
    "AvailabilityZone": "eu-west-1a",  
    "InstanceMatchCriteria": "open",  
    "EphemeralStorage": false,  
    "CreateDate": "2019-08-16T09:27:35.000Z",  
    "AvailableInstanceCount": 3,  
    "InstancePlatform": "Linux/UNIX",  
    "TotalInstanceCount": 3,  
    "State": "active",  
    "Tenancy": "default",  
    "EbsOptimized": false,  
    "InstanceType": "t2.medium"  
  }  
}
```

Beispiel 2: Um eine Kapazitätsreservierung zu erstellen, die automatisch an einem bestimmten Datum/einer bestimmten Uhrzeit endet

Im folgenden `create-capacity-reservation` Beispiel wird eine Kapazitätsreservierung in der `eu-west-1a` Availability Zone erstellt, in der Sie drei `m5.large` Instances starten können, auf denen ein Linux/Unix-Betriebssystem ausgeführt wird. Diese Kapazitätsreservierung endet automatisch am 31.08.2019 um 23:59:59 Uhr.

```
aws ec2 create-capacity-reservation \  
  --availability-zone eu-west-1a \  
  --instance-type m5.large \  
  --instance-platform Linux/UNIX \  
  --instance-count 3 \  
  --end-date-type Limited \  
  --end-date 2019-08-31T23:59:59Z
```

Ausgabe:

```
{
  "CapacityReservation": {
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "EndDateType": "limited",
    "AvailabilityZone": "eu-west-1a",
    "EndDate": "2019-08-31T23:59:59.000Z",
    "InstanceMatchCriteria": "open",
    "EphemeralStorage": false,
    "CreateDate": "2019-08-16T10:15:53.000Z",
    "AvailableInstanceCount": 3,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 3,
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": false,
    "InstanceType": "m5.large"
  }
}
```

Beispiel 3: So erstellen Sie eine Kapazitätsreservierung, die nur gezielte Instance-Starts akzeptiert

Im folgenden `create-capacity-reservation` Beispiel wird eine Kapazitätsreservierung erstellt, die nur gezielte Instance-Starts akzeptiert.

```
aws ec2 create-capacity-reservation \
  --availability-zone eu-west-1a \
  --instance-type m5.large \
  --instance-platform Linux/UNIX \
  --instance-count 3 \
  --instance-match-criteria targeted
```

Ausgabe:

```
{
  "CapacityReservation": {
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "EndDateType": "unlimited",
    "AvailabilityZone": "eu-west-1a",
    "InstanceMatchCriteria": "targeted",
    "EphemeralStorage": false,
  }
}
```

```
    "CreateDate": "2019-08-16T10:21:57.000Z",
    "AvailableInstanceCount": 3,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 3,
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": false,
    "InstanceType": "m5.large"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen einer Kapazitätsreservierung](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [CreateCapacityReservation](#) unter AWS CLI Befehlsreferenz.

create-carrier-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-carrier-gateway`.

AWS CLI

Um ein Carrier-Gateway zu erstellen

Im folgenden `create-carrier-gateway` Beispiel wird ein Carrier-Gateway für den angegebenen Server erstellt VPC.

```
aws ec2 create-carrier-gateway \
  --vpc-id vpc-0c529aEXAMPLE1111
```

Ausgabe:

```
{
  "CarrierGateway": {
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
    "VpcId": "vpc-0c529aEXAMPLE1111",
    "State": "pending",
    "OwnerId": "123456789012"
  }
}
```

Weitere Informationen finden Sie unter [Carrier Gateways im AWS Wavelength User Guide](#).

- API-Einheiten finden Sie [CreateCarrierGateway](#) in der AWS CLI Befehlsreferenz.

create-client-vpn-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-client-vpn-endpoint`.

AWS CLI

Um einen VPN Client-Endpunkt zu erstellen

Im folgenden `create-client-vpn-endpoint` Beispiel wird ein VPN Client-Endpunkt erstellt, der die gegenseitige Authentifizierung verwendet und einen Wert für den CIDR Client-Block angibt.

```
aws ec2 create-client-vpn-endpoint \
  --client-cidr-block "172.31.0.0/16" \
  --server-certificate-arn arn:aws:acm:ap-south-1:123456789012:certificate/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --authentication-options Type=certificate-
authentication,MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE} \
  --connection-log-options Enabled=false
```

Ausgabe:

```
{
  "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
  "Status": {
    "Code": "pending-associate"
  },
  "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com"
}
```

Weitere Informationen finden Sie unter [VPNClient-Endpunkte](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [CreateClientVpnEndpoint](#) in der AWS CLI Befehlsreferenz.

create-client-vpn-route

Das folgende Codebeispiel zeigt die Verwendung `create-client-vpn-route`.

AWS CLI

Um eine Route für einen VPN Client-Endpunkt zu erstellen

Das folgende `create-client-vpn-route` Beispiel fügt eine Route zum Internet (`0.0.0.0/0`) für das angegebene Subnetz des VPN Client-Endpunkts hinzu.

```
aws ec2 create-client-vpn-route \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --destination-cidr-block 0.0.0.0/0 \  
  --target-vpc-subnet-id subnet-0123456789abcabca
```

Ausgabe:

```
{  
  "Status": {  
    "Code": "creating"  
  }  
}
```

Weitere Informationen finden Sie unter [Routes](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [CreateClientVpnRoute](#) in der AWS CLI Befehlsreferenz.

`create-coip-cidr`

Das folgende Codebeispiel zeigt die Verwendung `create-coip-cidr`.

AWS CLI

Um einen Bereich von kundeneigenen IP-Adressen (CoIP) zu erstellen

Im folgenden `create-coip-cidr` Beispiel wird der angegebene CoIP-Adressbereich im angegebenen CoIP-Pool erstellt.

```
aws ec2 create-coip-cidr \  
  --cidr 15.0.0.0/24 \  
  --coip-pool-id ipvp4pool-coip-1234567890abcdefg
```

Ausgabe:

```
{
  "CoipCidr": {
    "Cidr": "15.0.0.0/24",
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
  }
}
```

Weitere Informationen finden Sie unter [Kundeneigene IP-Adressen](#) im AWS -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateCoipCidr AWS CLI](#) Befehlsreferenz.

create-coip-pool

Das folgende Codebeispiel zeigt die Verwendung `create-coip-pool`.

AWS CLI

Um einen Pool von kundeneigenen IP-Adressen (CoIP) zu erstellen

Im folgenden `create-coip-pool` Beispiel wird ein CoIP-Pool für CoIP-Adressen in der angegebenen Routentabelle des lokalen Gateways erstellt.

```
aws ec2 create-coip-pool \
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Ausgabe:

```
{
  "CoipPool": {
    "PoolId": "ipv4pool-coip-1234567890abcdefg",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-coip-1234567890abcdefg"
  }
}
```

Weitere Informationen finden Sie unter [Kundeneigene IP-Adressen](#) im AWS -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateCoipPool AWS CLI](#) Befehlsreferenz.

create-customer-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-customer-gateway`.

AWS CLI

Um ein Kunden-Gateway zu erstellen

In diesem Beispiel wird ein Kunden-Gateway mit der angegebenen IP-Adresse für die externe Schnittstelle erstellt.

Befehl:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

Ausgabe:

```
{
  "CustomerGateway": {
    "CustomerGatewayId": "cgw-0e11f167",
    "IpAddress": "12.1.2.3",
    "State": "available",
    "Type": "ipsec.1",
    "BgpAsn": "65534"
  }
}
```

- API-Einheiten finden Sie [CreateCustomerGateway](#) unter AWS CLI Befehlsreferenz.

create-default-subnet

Das folgende Codebeispiel zeigt die Verwendung `create-default-subnet`.

AWS CLI

Um ein Standardsubnetz zu erstellen

In diesem Beispiel wird ein Standardsubnetz in der Availability Zone erstellt. `us-east-2a`

Befehl:

```
aws ec2 create-default-subnet --availability-zone us-east-2a
```



```
{
  "Subnet": {
    "AvailabilityZone": "us-east-2a",
    "Tags": [],
    "AvailableIpAddressCount": 4091,
    "DefaultForAz": true,
    "Ipv6CidrBlockAssociationSet": [],
    "VpcId": "vpc-1a2b3c4d",
    "State": "available",
    "MapPublicIpOnLaunch": true,
    "SubnetId": "subnet-1122aabb",
    "CidrBlock": "172.31.32.0/20",
    "AssignIpv6AddressOnCreation": false
  }
}
```

- API-Einheiten finden Sie unter [CreateDefaultSubnet AWS CLI](#) Befehlsreferenz.

create-default-vpc

Das folgende Codebeispiel zeigt die Verwendung `create-default-vpc`.

AWS CLI

Um einen Standard zu erstellen VPC

In diesem Beispiel wird eine Standardeinstellung erstellt VPC.

Befehl:

```
aws ec2 create-default-vpc
```

Ausgabe:

```
{
  "Vpc": {
    "VpcId": "vpc-8eaae5ea",
    "InstanceTenancy": "default",
    "Tags": [],
    "Ipv6CidrBlockAssociationSet": [],
    "State": "pending",
    "DhcpOptionsId": "dopt-af0c32c6",
    "CidrBlock": "172.31.0.0/16",
  }
}
```

```

    "IsDefault": true
  }
}

```

- API-Einheiten finden Sie [CreateDefaultVpc](#) in der AWS CLI Befehlsreferenz.

create-dhcp-options

Das folgende Codebeispiel zeigt die Verwendung `create-dhcp-options`.

AWS CLI

Um eine Reihe von DHCP-Optionen zu erstellen

Im folgenden `create-dhcp-options` Beispiel wird eine Reihe von DHCP-Optionen erstellt, die den Domännennamen, die Domännennamenserver und den BIOS-Netzknottentyp angeben.

```

aws ec2 create-dhcp-options \
  --dhcp-configuration \
    "Key=domain-name-servers,Values=10.2.5.1,10.2.5.2" \
    "Key=domain-name,Values=example.com" \
    "Key=netbios-node-type,Values=2"

```

Ausgabe:

```

{
  "DhcpOptions": {
    "DhcpConfigurations": [
      {
        "Key": "domain-name",
        "Values": [
          {
            "Value": "example.com"
          }
        ]
      },
      {
        "Key": "domain-name-servers",
        "Values": [
          {
            "Value": "10.2.5.1"
          }
        ],

```

```

        {
            "Value": "10.2.5.2"
        }
    ],
    {
        "Key": "netbios-node-type",
        "Values": [
            {
                "Value": "2"
            }
        ]
    }
],
"DhcpOptionsId": "dopt-06d52773eff4c55f3"
}
}

```

- API-Einheiten finden Sie [CreateDhcpOptions](#) unter AWS CLI Befehlsreferenz.

create-egress-only-internet-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-egress-only-internet-gateway`.

AWS CLI

Um ein Internet-Gateway nur für ausgehenden Datenverkehr zu erstellen

In diesem Beispiel wird für den angegebenen Server ein Internet-Gateway nur für ausgehenden Datenverkehr erstellt. VPC

Befehl:

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

Ausgabe:

```

{
  "EgressOnlyInternetGateway": {
    "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
    "Attachments": [
      {

```

```

        "State": "attached",
        "VpcId": "vpc-0c62a468"
    }
]
}
}

```

- API-Einheiten finden Sie [CreateEgressOnlyInternetGateway](#) in AWS CLI der Befehlsreferenz.

create-fleet

Das folgende Codebeispiel zeigt die Verwendung `create-fleet`.

AWS CLI

Beispiel 1: Um eine EC2 Flotte zu erstellen, die Spot-Instances als Standardkaufmodell einführt

Im folgenden `create-fleet` Beispiel wird eine EC2 Flotte mit den Mindestparametern erstellt, die für den Start einer Flotte erforderlich sind: eine Startvorlage, eine Zielkapazität und ein Standard-Einkaufsmodell. Die Startvorlage wird durch ihre Startvorlagen-ID und Versionsnummer identifiziert. Die Zielkapazität für die Flotte beträgt 2 Instances, und das Standard-Kaufmodell ist `spot`, was dazu führt, dass die Flotte 2 Spot-Instances startet.

Wenn Sie eine EC2 Flotte erstellen, verwenden Sie eine JSON Datei, um Informationen zu den Instances anzugeben, die gestartet werden sollen.

```

aws ec2 create-fleet \
  --cli-input-json file://file_name.json

```

Inhalt von `file_name.json`:

```

{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,

```

```
    "DefaultTargetCapacityType": "spot"
  }
}
```

Ausgabe:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Beispiel 2: Um eine EC2 Flotte zu erstellen, die On-Demand-Instances als Standard-Kaufmodell startet

Im folgenden `create-fleet` Beispiel wird eine EC2 Flotte mit den Mindestparametern erstellt, die für den Start einer Flotte erforderlich sind: eine Startvorlage, eine Zielkapazität und ein Standard-Einkaufsmodell. Die Startvorlage wird durch ihre Startvorlagen-ID und Versionsnummer identifiziert. Die Zielkapazität für die Flotte beträgt 2 Instances, und das Standard-Kaufmodell ist `on-demand`, was dazu führt, dass die Flotte 2 On-Demand-Instances startet.

Wenn Sie eine EC2 Flotte erstellen, verwenden Sie eine JSON Datei, um Informationen zu den Instances anzugeben, die gestartet werden sollen.

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.json
```

Inhalt von `file_name.json`:

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
  }
}
```

Ausgabe:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Beispiel 3: Um eine EC2 Flotte zu erstellen, die On-Demand-Instances als primäre Kapazität startet

Im folgenden `create-fleet` Beispiel wird eine EC2 Flotte erstellt, die die Gesamtzielkapazität von 2 Instances für die Flotte und eine Zielkapazität von 1 On-Demand-Instance angibt. Das Standard-Einkaufsmodell ist `spot`. Die Flotte startet wie angegeben eine On-Demand-Instance, muss aber eine weitere Instance starten, um die gesamte Zielkapazität zu erreichen. Das Kaufmodell für die Differenz wird wie folgt berechnet: `TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType`, was dazu führt, dass die Flotte 1 Spot-Instance startet.

Wenn Sie eine EC2 Flotte erstellen, verwenden Sie eine JSON Datei, um Informationen zu den Instances anzugeben, die gestartet werden sollen.

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.json
```

Inhalt von `file_name.json`:

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "OnDemandTargetCapacity": 1,
    "DefaultTargetCapacityType": "spot"
  }
}
```

Ausgabe:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Beispiel 4: Um eine EC2 Flotte zu erstellen, die Spot-Instances mit der Zuweisungsstrategie zum niedrigsten Preis startet

Wenn die Zuweisungsstrategie für Spot-Instances nicht angegeben ist, wird die Standard-Zuweisungsstrategie, d. h. `lowest-price` verwendet. Im folgenden `create-fleet` Beispiel wird mithilfe der EC2 `lowest-price` Zuweisungsstrategie eine Flotte erstellt. Die drei Startspezifikationen, die die Startvorlage überschreiben, haben unterschiedliche Instance-Typen, aber die gleiche gewichtete Kapazität und das gleiche Subnetz. Die Gesamtzielkapazität beträgt 2 Instances und das Standard-Einkaufsmodell ist `spot`. Die EC2 Flotte startet 2 Spot-Instances unter Verwendung des Instance-Typs, der in der Startspezifikation mit dem niedrigsten Preis angegeben ist.

Wenn Sie eine EC2 Flotte erstellen, verwenden Sie eine JSON Datei, um Informationen über die zu startenden Instances anzugeben.

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.jsonContents of file_name.json::
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      },
      "Overrides": [
        {
          "InstanceType": "c4.large",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-a4f6c5d3"
        },
        {
          "InstanceType": "c3.large",
          "WeightedCapacity": 1,
          "SubnetId": "subnet-a4f6c5d3"
        }
      ]
    }
  ]
}
```

```
    },  
    {  
      "InstanceType": "c5.large",  
      "WeightedCapacity": 1,  
      "SubnetId": "subnet-a4f6c5d3"  
    }  
  ],  
  "TargetCapacitySpecification": {  
    "TotalTargetCapacity": 2,  
    "DefaultTargetCapacityType": "spot"  
  }  
}
```

Ausgabe:

```
{  
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

- API-Einheiten finden Sie [CreateFleet](#) unter AWS CLI Befehlsreferenz.

create-flow-logs

Das folgende Codebeispiel zeigt die Verwendung `create-flow-logs`.

AWS CLI

Beispiel 1: Um ein Flow-Protokoll zu erstellen

Im folgenden `create-flow-logs` Beispiel wird ein Flow-Protokoll erstellt, das den gesamten abgelehnten Datenverkehr für die angegebene Netzwerkschnittstelle aufzeichnet. Die Flow-Protokolle werden mithilfe der Berechtigungen in der angegebenen IAM Rolle an eine Protokollgruppe in CloudWatch Logs übermittelt.

```
aws ec2 create-flow-logs \  
  --resource-type NetworkInterface \  
  --resource-ids eni-11223344556677889 \  
  --traffic-type REJECT \  
  --log-group-name my-flow-logs \  
  --log-stream-name my-flow-logs-stream
```



```
--deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
```

Ausgabe:

```
{
  "ClientToken": "so0eNA2uSHUN1HI0S2cJ305GuIX1CezaRdGtexample",
  "FlowLogIds": [
    "fl-12345678901234567"
  ],
  "Unsuccessful": []
}
```

Weitere Informationen finden Sie unter [VPCFlow Logs](#) im VPCAmazon-Benutzerhandbuch.

Beispiel 2: Um ein Flow-Protokoll mit einem benutzerdefinierten Format zu erstellen

Das folgende `create-flow-logs` Beispiel erstellt ein Flow-Protokoll, das den gesamten Datenverkehr für die angegebenen Daten erfasst VPC und die Flow-Logs an einen Amazon S3 S3-Bucket übermittelt. Der Parameter `--log-format` legt ein benutzerdefiniertes Format für die Flow-Protokolldatensätze fest. Um diesen Befehl unter Windows auszuführen, ändern Sie die einfachen Anführungszeichen (') in doppelte Anführungszeichen (,).

```
aws ec2 create-flow-logs \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --traffic-type ALL \
  --log-destination-type s3 \
  --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
  --log-format '${version} ${vpc-id} ${subnet-id} ${instance-id} ${srcaddr}
  ${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-srcaddr}
  ${pkt-dstaddr}'
```

Weitere Informationen finden Sie unter [VPCFlow Logs](#) im VPCAmazon-Benutzerhandbuch.

Beispiel 3: Um ein Flow-Protokoll mit einem maximalen Aggregationsintervall von einer Minute zu erstellen

Das folgende `create-flow-logs` Beispiel erstellt ein Flow-Protokoll, das den gesamten Datenverkehr für die angegebenen Daten erfasst VPC und die Flow-Logs an einen Amazon S3 S3-Bucket übermittelt. Der `--max-aggregation-interval` Parameter gibt ein maximales Aggregationsintervall von 60 Sekunden (1 Minute) an.

```
aws ec2 create-flow-logs \  
  --resource-type VPC \  
  --resource-ids vpc-00112233344556677 \  
  --traffic-type ALL \  
  --log-destination-type s3 \  
  --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \  
  --max-aggregation-interval 60
```

Weitere Informationen finden Sie unter [VPCFlow Logs](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CreateFlowLogs](#) in der AWS CLI Befehlsreferenz.

create-fpga-image

Das folgende Codebeispiel zeigt die Verwendung `create-fpga-image`.

AWS CLI

Um ein FPGA Amazon-Image zu erstellen

In diesem Beispiel wird ein AFI aus dem angegebenen Tarball im angegebenen Bucket erstellt.

Befehl:

```
aws ec2 create-fpga-image --name my-afi --description test-afi --input-storage-  
location Bucket=my-fpga-bucket,Key=dcp/17_12_22-103226.Developer_CL.tar --logs-  
storage-location Bucket=my-fpga-bucket,Key=logs
```

Ausgabe:

```
{  
  "FpgaImageId": "afi-0d123e123bfc85abc",  
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"  
}
```

- APIEinzelheiten finden Sie [CreateFpgaImage](#) in der AWS CLI Befehlsreferenz.

create-image

Das folgende Codebeispiel zeigt die Verwendung `create-image`.

AWS CLI

Beispiel 1: So erstellen Sie eine AMI von einer EBS Amazon-gestützten Instance

Im folgenden `create-image` Beispiel wird AMI aus der angegebenen Instance eine erstellt.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --description "An AMI for my server"
```

Ausgabe:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Weitere Informationen zur Angabe einer Blockgerätezuweisung für Sie AMI finden Sie unter [Angeben einer Blockgerätezuweisung für eine AMI](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: So erstellen Sie eine AMI aus einer EBS Amazon-gestützten Instance ohne Neustart

Das folgende `create-image` Beispiel erstellt eine AMI und legt den Parameter `--no-reboot` fest, sodass die Instance nicht neu gestartet wird, bevor das Image erstellt wurde.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --no-reboot
```

Ausgabe:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Weitere Informationen zur Angabe einer Blockgerätezuweisung für Sie AMI finden Sie unter [Angeben einer Blockgerätezuweisung für eine AMI](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 3: Um ein AMI und Schnappschüsse bei der Erstellung zu taggen

Im folgenden `create-image` Beispiel wird ein AMI, AMI und die Schnappschüsse werden mit demselben Tag versehen `cost-center=cc123`

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --tag-specifications "ResourceType=image,Tags=[{Key=cost-center,Value=cc123}]" "ResourceType=snapshot,Tags=[{Key=cost-center,Value=cc123}]"
```

Ausgabe:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Weitere Informationen zum Taggen Ihrer Ressourcen bei der Erstellung finden [Sie unter Hinzufügen von Tags bei der Erstellung von Ressourcen](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateImage](#) in der AWS CLI Befehlsreferenz.

create-instance-connect-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-instance-connect-endpoint`.

AWS CLI

So erstellen Sie einen EC2 Instance Connect-Endpoint

Im folgenden `create-instance-connect-endpoint` Beispiel wird ein EC2 Instance Connect-Endpoint im angegebenen Subnetz erstellt.

```
aws ec2 create-instance-connect-endpoint \  
  --region us-east-1 \  
  --subnet-id subnet-0123456789example
```

Ausgabe:

```
{  
  "VpcId": "vpc-0123abcd",  
  "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
```

```

    "AvailabilityZone": "us-east-1a",
    "NetworkInterfaceIds": [
        "eni-0123abcd"
    ],
    "PreserveClientIp": true,
    "Tags": [],
    "FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
    "StateMessage": "",
    "State": "create-complete",
    "DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-endpoint.us-
east-1.amazonaws.com",
    "SubnetId": "subnet-0123abcd",
    "OwnerId": "111111111111",
    "SecurityGroupIds": [
        "sg-0123abcd"
    ],
    "InstanceConnectEndpointId": "eice-0123456789example",
    "CreatedAt": "2023-04-07T15:43:53.000Z"
}

```

Weitere Informationen finden Sie unter [Erstellen eines EC2 Instance Connect-Endpunkts](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateInstanceConnectEndpoint](#) in der AWS CLI Befehlsreferenz.

create-instance-event-window

Das folgende Codebeispiel zeigt die Verwendung `create-instance-event-window`.

AWS CLI

Beispiel 1: Um ein Ereignisfenster mit einem Zeitraum zu erstellen

Im folgenden `create-instance-event-window` Beispiel wird ein Ereignisfenster mit einem Zeitraum erstellt. Sie können außerdem den Parameter `cron-expression` nicht angeben.

```

aws ec2 create-instance-event-window \
  --region us-east-1 \
  --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
  --tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName

```

Ausgabe:

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "TimeRanges": [
      {
        "StartWeekDay": "monday",
        "StartHour": 2,
        "EndWeekDay": "wednesday",
        "EndHour": 8
      }
    ],
    "Name": "myEventWindowName",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 2: Um ein Ereignisfenster mit einem Cron-Ausdruck zu erstellen

Das folgende `create-instance-event-window` Beispiel erstellt ein Ereignisfenster mit einem Cron-Ausdruck. Sie können außerdem den Parameter `time-range` nicht angeben.

```
aws ec2 create-instance-event-window \
  --region us-east-1 \
  --cron-expression "* 21-23 * * 2,3" \
  --tag-specifications "ResourceType=instance-event-window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName
```

Ausgabe:

```
{
```

```

    "InstanceEventWindow": {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "State": "creating",
      "Tags": [
        {
          "Key": "K1",
          "Value": "V1"
        }
      ]
    }
  }
}

```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

- API-Einheiten finden Sie [CreateInstanceEventWindow](#) in der AWS CLI Befehlsreferenz.

create-instance-export-task

Das folgende Codebeispiel zeigt die Verwendung `create-instance-export-task`.

AWS CLI

Um eine Instanz zu exportieren

Dieser Beispielbefehl erstellt eine Aufgabe zum Exportieren der Instance `i-1234567890abcdef0` in den Amazon S3 S3-Bucket `myexportbucket`.

Befehl:

```

aws ec2 create-instance-export-task --description "RHEL5 instance" --
instance-id i-1234567890abcdef0 --target-environment vmware --export-to-s3-
task DiskImageFormat=vmdk,ContainerFormat=ova,S3Bucket=myexportbucket,S3Prefix=RHEL5

```

Ausgabe:

```

{
  "ExportTask": {
    "State": "active",
    "InstanceExportDetails": {

```

```

        "InstanceId": "i-1234567890abcdef0",
        "TargetEnvironment": "vmware"
    },
    "ExportToS3Task": {
        "S3Bucket": "myexportbucket",
        "S3Key": "RHEL5export-i-fh8sjjsq.ova",
        "DiskImageFormat": "vmdk",
        "ContainerFormat": "ova"
    },
    "Description": "RHEL5 instance",
    "ExportTaskId": "export-i-fh8sjjsq"
}
}

```

- API-Einheiten finden Sie unter Befehlsreferenz. [CreateInstanceExportTask](#) AWS CLI

create-internet-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-internet-gateway`.

AWS CLI

Um ein Internet-Gateway zu erstellen

Im folgenden `create-internet-gateway` Beispiel wird ein Internet-Gateway mit dem Tag `erstelltName=my-igw`.

```

aws ec2 create-internet-gateway \
  --tag-specifications ResourceType=internet-gateway, Tags=[{Key=Name, Value=my-igw}]

```

Ausgabe:

```

{
  "InternetGateway": {
    "Attachments": [],
    "InternetGatewayId": "igw-0d0fb496b3994d755",
    "OwnerId": "123456789012",
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-igw"
      }
    ]
  }
}

```



```

    }
  ]
}

```

Weitere Informationen finden Sie unter [Internet Gateways](#) im VPC Amazon-Benutzerhandbuch.

- API Einzelheiten finden Sie [CreateInternetGateway](#) in der AWS CLI Befehlsreferenz.

create-ipam-pool

Das folgende Codebeispiel zeigt die Verwendung `create-ipam-pool`.

AWS CLI

Um einen IPAM Pool zu erstellen

Das folgende `create-ipam-pool` Beispiel erstellt einen IPAM Pool.

(Linux):

```

aws ec2 create-ipam-pool \
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
  --address-family ipv4 \
  --auto-import \
  --allocation-min-netmask-length 16 \
  --allocation-max-netmask-length 26 \
  --allocation-default-netmask-length 24 \
  --allocation-resource-tags "Key=Environment,Value=Preprod" \
  --tag-specifications 'ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod
pool"}]'

```

(Windows):

```

aws ec2 create-ipam-pool ^
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^
  --address-family ipv4 ^
  --auto-import ^
  --allocation-min-netmask-length 16 ^
  --allocation-max-netmask-length 26 ^
  --allocation-default-netmask-length 24 ^
  --allocation-resource-tags "Key=Environment,Value=Preprod" ^

```

```
--tag-specifications ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod pool"}]
```

Ausgabe:

```
{
  "IpamPool": {
    "OwnerId": "123456789012",
    "IpamPoolId": "ipam-pool-0533048da7d823723",
    "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-0533048da7d823723",
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
    "IpamScopeType": "private",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
    "IpamRegion": "us-east-1",
    "Locale": "None",
    "PoolDepth": 1,
    "State": "create-in-progress",
    "AutoImport": true,
    "AddressFamily": "ipv4",
    "AllocationMinNetmaskLength": 16,
    "AllocationMaxNetmaskLength": 26,
    "AllocationDefaultNetmaskLength": 24,
    "AllocationResourceTags": [
      {
        "Key": "Environment",
        "Value": "Preprod"
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "Preprod pool"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Plan für die Bereitstellung von IP-Adressen](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateIpamPool](#) in der AWS CLI Befehlsreferenz.

create-ipam-resource-discovery

Das folgende Codebeispiel zeigt die Verwendung `create-ipam-resource-discovery`.

AWS CLI

Um eine Ressourcenerkennung zu erstellen

In diesem Beispiel sind Sie ein delegierter IPAM Administrator, der eine Ressourcenerkennung erstellen und mit dem IPAM Administrator in einer anderen AWS Organisation teilen möchte, sodass der Administrator in der anderen Organisation die IP-Adressen der Ressourcen in Ihrer Organisation verwalten und überwachen kann.

Wichtig

Dieses Beispiel umfasst `--region` sowohl die `--operating-regions` Optionen als auch, da sie zwar optional sind, aber auf eine bestimmte Weise konfiguriert werden müssen, um eine Ressourcenerkennung erfolgreich mit einem* zu integrieren IPAM. * `--operating-regions` muss den Regionen entsprechen, in denen Sie Ressourcen haben, die Sie ermitteln IPAM möchten. Wenn es Regionen gibt, in denen Sie die IP-Adressen nicht verwalten IPAM möchten (z. B. aus Compliance-Gründen), schließen Sie diese nicht ein. * `--region` muss mit der Heimatregion der Region übereinstimmen, der IPAM Sie sie zuordnen möchten. Sie müssen die Ressourcensuche in derselben Region erstellen, in der die erstellt IPAM wurde. Wenn die, mit der IPAM Sie verknüpfen, beispielsweise in `us-east-1` erstellt wurde, fügen Sie sie `--region us-east-1` in die Anfrage ein. Beide `--operating-regions` Optionen `--region` und sind standardmäßig auf die Region eingestellt, in der Sie den Befehl ausführen, wenn Sie sie nicht angeben.

In diesem Beispiel beinhalten die Betriebsregionen von, mit denen IPAM wir die Integration durchführen,, und. `us-west-1 us-west-2 ap-south-1` Wenn wir die Ressourcenerkennung erstellen, IPAM möchten wir die Ressourcen-IP-Adressen in `us-west-1` und `us-west-2` aber nicht ermitteln `ap-south-1`. Wir beziehen sie also nur `--operating-regions` `RegionName='us-west-1' RegionName='us-west-2'` in die Anfrage mit ein.

Das folgende `create-ipam-resource-discovery` Beispiel erstellt eine IPAM Ressourcenerkennung.

```
aws ec2 create-ipam-resource-discovery \
  --description 'Example-resource-discovery' \
  --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-
center,Value=cc123}]' \
```

```
--operating-regions RegionName='us-west-1' RegionName='us-west-2' \
--region us-east-1
```

Ausgabe:

```
{
  "IpamResourceDiscovery":{
    "OwnerId": "149977607591",
    "IpamResourceDiscoveryId": "ipam-res-disco-0257046d8aa78b8bc",
    "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0257046d8aa78b8bc",
    "IpamResourceDiscoveryRegion": "us-east-1",
    "Description": "'Example-resource-discovery'",
    "OperatingRegions":[
      {"RegionName": "us-west-1"},
      {"RegionName": "us-west-2"},
      {"RegionName": "us-east-1"}
    ],
    "IsDefault": false,
    "State": "create-in-progress",
    "Tags": [
      {
        "Key": "cost-center",
        "Value": "cc123"
      }
    ]
  }
}
```

Sobald Sie eine Ressourcensuche erstellt haben, möchten Sie sie möglicherweise mit einem anderen IPAM delegierten Administrator teilen, was Sie tun können. [create-resource-share](#) Weitere Informationen finden Sie im VPC/IPAM Amazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

- API-Einheiten finden Sie [CreateIpamResourceDiscovery](#) in der AWS CLI Befehlsreferenz.

create-ipam-scope

Das folgende Codebeispiel zeigt die Verwendung `create-ipam-scope`.

AWS CLI

Um einen IPAM-Bereich zu erstellen

Im folgenden `create-ipam-scope` Beispiel wird ein IPAM Bereich erstellt.

(Linux):

```
aws ec2 create-ipam-scope \  
  --ipam-id ipam-08440e7a3acde3908 \  
  --description "Example description" \  
  --tag-specifications 'ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example  
name value"}]'
```

(Windows):

```
aws ec2 create-ipam-scope ^  
  --ipam-id ipam-08440e7a3acde3908 ^  
  --description "Example description" ^  
  --tag-specifications ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example name  
value"}]
```

Ausgabe:

```
{  
  "IpamScope": {  
    "OwnerId": "123456789012",  
    "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",  
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-01c1ebab2b63bd7e4",  
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
    "IpamRegion": "us-east-1",  
    "IpamScopeType": "private",  
    "IsDefault": false,  
    "Description": "Example description",  
    "PoolCount": 0,  
    "State": "create-in-progress",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "Example name value"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Zusätzliche Bereiche erstellen](#).

- APIEinzelheiten finden Sie [CreateIpamScope](#) in der AWS CLI Befehlsreferenz.

create-ipam

Das folgende Codebeispiel zeigt die Verwendung `create-ipam`.

AWS CLI

Um ein zu erstellen IPAM

Das folgende `create-ipam` Beispiel erstellt eine IPAM.

(Linux):

```
aws ec2 create-ipam \
  --description "Example description" \
  --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" \
  --tag-specifications 'ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]'
```

(Windows):

```
aws ec2 create-ipam ^
  --description "Example description" ^
  --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" ^
  --tag-specifications ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]
```

Ausgabe:

```
{
  "Ipam": {
    "OwnerId": "123456789012",
    "IpamId": "ipam-036486dfa6af58ee0",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
    "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
    "ScopeCount": 2,
    "OperatingRegions": [
      {
        "RegionName": "us-east-2"
```

```
    },
    {
      "RegionName": "us-west-1"
    },
    {
      "RegionName": "us-east-1"
    }
  ],
  "State": "create-in-progress",
  "Tags": [
    {
      "Key": "Name",
      "Value": "ExampleIPAM"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Create an IPAM](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [Createlpam](#) in der AWS CLI Befehlsreferenz.

create-key-pair

Das folgende Codebeispiel zeigt die Verwendung `create-key-pair`.

AWS CLI

So erstellen Sie ein Schlüsselpaar

In diesem Beispiel wird eine Schlüsselrolle mit dem Namen `MyKeyPair` erstellt.

Befehl:

```
aws ec2 create-key-pair --key-name MyKeyPair
```

Die Ausgabe ist eine ASCII Version des privaten Schlüssels und des Schlüsselfingerabdrucks. Sie müssen den Schlüssel in einer Datei speichern.

Weitere Informationen finden Sie unter Verwenden von Schlüsselpaaren im Benutzerhandbuch für die AWS -Befehlszeilenschnittstelle.

- APIEinzelheiten finden Sie [CreateKeyPair](#) in der AWS CLI Befehlsreferenz.

create-launch-template-version

Das folgende Codebeispiel zeigt die Verwendung `create-launch-template-version`.

AWS CLI

Um eine Startvorlagenversion zu erstellen

In diesem Beispiel wird eine neue Version der Startvorlage erstellt, die auf Version 1 der Startvorlage basiert, und es wird eine andere AMI ID angegeben.

Befehl:

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
--version-description WebVersion2 --source-version 1 --launch-template-data
'{"ImageId": "ami-c998b6b2"}'
```

Ausgabe:

```
{
  "LaunchTemplateVersion": {
    "VersionDescription": "WebVersion2",
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "WebServers",
    "VersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "LaunchTemplateData": {
      "ImageId": "ami-c998b6b2",
      "InstanceType": "t2.micro",
      "NetworkInterfaces": [
        {
          "Ipv6Addresses": [
            {
              "Ipv6Address": "2001:db8:1234:1a00::123"
            }
          ],
          "DeviceIndex": 0,
          "SubnetId": "subnet-7b16de0c",
          "AssociatePublicIpAddress": true
        }
      ]
    },
    "DefaultVersion": false,
  }
}
```



```

    "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}

```

- API-Einheiten finden Sie [CreateLaunchTemplateVersion](#) unter AWS CLI Befehlsreferenz.

create-launch-template

Das folgende Codebeispiel zeigt die Verwendung `create-launch-template`.

AWS CLI

Beispiel 1: So erstellen Sie eine Startvorlage

Im folgenden `create-launch-template` Beispiel wird eine Startvorlage erstellt, die das Subnetz angibt, in dem die Instance gestartet werden soll, der Instance eine öffentliche IP-Adresse und eine IPv6 Adresse zuweist und ein Tag für die Instance erstellt.

```

aws ec2 create-launch-template \
  --launch-template-name TemplateForWebServer \
  --version-description WebVersion1 \
  --launch-template-data '{"NetworkInterfaces":
  [{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1,"SubnetId":"subnet-7b
  [{"ResourceType":"instance","Tags":[{"Key":"purpose","Value":"webserver"}]}]}'

```

Ausgabe:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-01238c059e3466abc",
    "LaunchTemplateName": "TemplateForWebServer",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-01-27T09:13:24.000Z"
  }
}

```

Weitere Informationen finden Sie unter [Starten einer Instance von einer Startvorlage](#) im Benutzerhandbuch für Amazon Elastic Compute Cloud. Informationen zum Angeben von

Parametern im JSON Format „Quoting“ finden Sie unter Quoting Strings im Command Line Interface User Guide.AWS

Beispiel 2: So erstellen Sie eine Startvorlage für Amazon EC2 Auto Scaling

Im folgenden `create-launch-template` Beispiel wird eine Startvorlage mit mehreren Tags und einer Blockgerätezuordnung erstellt, um beim Start einer Instance ein zusätzliches EBS Volume anzugeben. Geben Sie einen Wert an `Groups`, der den Sicherheitsgruppen für die entspricht, in VPC denen Ihre Auto Scaling Scaling-Gruppe Instances starten wird. Geben Sie die Subnetze VPC und als Eigenschaften der Auto Scaling Scaling-Gruppe an.

```
aws ec2 create-launch-template \
  --launch-template-name TemplateForAutoScaling \
  --version-description AutoScalingVersion1 \
  --launch-template-data '{"NetworkInterfaces":
  [{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
  [{"sg-7c227019,sg-903004f8"},"DeleteOnTermination":true}],"ImageId":"ami-
  b42209de","InstanceType":"m4.large","TagSpecifications":
  [{"ResourceType":"instance","Tags":[{"Key":"environment","Value":"production"},
  {"Key":"purpose","Value":"webserver"}]},{"ResourceType":"volume","Tags":
  [{"Key":"environment","Value":"production"}, {"Key":"cost-
  center","Value":"cc123"}]}],"BlockDeviceMappings":[{"DeviceName":"/dev/sda1","Ebs":
  {"VolumeSize":100}}]}' --region us-east-1
```

Ausgabe:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0123c79c33a54e0abc",
    "LaunchTemplateName": "TemplateForAutoScaling",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-04-30T18:16:06.000Z"
  }
}
```

Weitere Informationen finden Sie unter Erstellen einer Startvorlage für eine Auto Scaling Scaling-Gruppe im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch. Informationen zum Angeben von Parametern im JSON Format „Quoting Strings“ finden Sie unter Quoting Strings im AWS Command Line Interface User Guide.

Beispiel 3: So erstellen Sie eine Startvorlage, die die Verschlüsselung von Volumes spezifiziert EBS

Im folgenden `create-launch-template` Beispiel wird eine Startvorlage erstellt, die verschlüsselte EBS Volumes enthält, die aus einem unverschlüsselten Snapshot erstellt wurden. Außerdem werden die Volumes bei der Erstellung mit Tags versehen. Wenn die standardmäßige Verschlüsselung deaktiviert ist, müssen Sie die `"Encrypted"`-Option wie im folgenden Beispiel gezeigt angeben. Wenn Sie die `"KmsKeyId"` Option verwenden, um ein vom Kunden verwaltetes Objekt anzugeben CMK, müssen Sie die `"Encrypted"` Option auch dann angeben, wenn die Verschlüsselung standardmäßig aktiviert ist.

```
aws ec2 create-launch-template \  
  --launch-template-name TemplateForEncryption \  
  --launch-template-data file://config.json
```

Inhalt von `config.json`:

```
{  
  "BlockDeviceMappings":[  
    {  
      "DeviceName":"/dev/sda1",  
      "Ebs":{  
        "VolumeType":"gp2",  
        "DeleteOnTermination":true,  
        "SnapshotId":"snap-066877671789bd71b",  
        "Encrypted":true,  
        "KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/abcd1234-  
a123-456a-a12b-a123b4cd56ef"  
      }  
    }  
  ],  
  "ImageId":"ami-00068cd7555f543d5",  
  "InstanceType":"c5.large",  
  "TagSpecifications":[  
    {  
      "ResourceType":"volume",  
      "Tags":[  
        {  
          "Key":"encrypted",  
          "Value":"yes"  
        }  
      ]  
    }  
  ]  
}
```

```

    }
  ]
}

```

Ausgabe:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
    "LaunchTemplateName": "TemplateForEncryption",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2020-01-07T19:08:36.000Z"
  }
}

```

Weitere Informationen finden Sie unter Wiederherstellung eines EBS Amazon-Volumes aus einem Snapshot und Standardverschlüsselung im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateLaunchTemplate](#) in der AWS CLI Befehlsreferenz.

create-local-gateway-route-table-virtual-interface-group-association

Das folgende Codebeispiel zeigt die Verwendung `create-local-gateway-route-table-virtual-interface-group-association`.

AWS CLI

Um eine lokale Gateway-Routentabelle einer virtuellen Interfaces (VIFs) -Gruppe zuzuordnen

Im folgenden `create-local-gateway-route-table-virtual-interface-group-association` Beispiel wird eine Zuordnung zwischen der angegebenen lokalen Gateway-Routentabelle und der angegebenen VIF Gruppe erstellt.

```

aws ec2 create-local-gateway-route-table-virtual-interface-group-association \
  --local-gateway-route-table-id lgw-rtb-exampleidabcd1234 \
  --local-gateway-virtual-interface-group-id lgw-vif-grp-exampleid0123abcd

```

Ausgabe:

```
{
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-
assoc-exampleid12345678",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",
    "LocalGatewayId": "lgw-exampleid11223344",
    "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-exampleidabcd1234",
    "OwnerId": "111122223333",
    "State": "pending",
    "Tags": []
  }
}
```

Weitere Informationen finden Sie unter [VIFGruppenzuordnungen](#) im AWS Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#) in der AWS CLI Befehlsreferenz.

create-local-gateway-route-table-vpc-association

Das folgende Codebeispiel zeigt die Verwendung `create-local-gateway-route-table-vpc-association`.

AWS CLI

Um eine VPC mit einer Routentabelle zu verknüpfen

Im folgenden `create-local-gateway-route-table-vpc-association` Beispiel wird die angegebene VPC Routentabelle der angegebenen lokalen Gateway-Routentabelle zugeordnet.

```
aws ec2 create-local-gateway-route-table-vpc-association \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
  --vpc-id vpc-07ef66ac71EXAMPLE
```

Ausgabe:

```
{
```

```

    "LocalGatewayRouteTableVpcAssociation": {
      "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
      "VpcId": "vpc-07ef66ac71EXAMPLE",
      "State": "associated"
    }
  }
}

```

- API-Einheiten finden Sie [CreateLocalGatewayRouteTableVpcAssociation](#) unter AWS CLI Befehlsreferenz.

create-local-gateway-route-table

Das folgende Codebeispiel zeigt die Verwendung `create-local-gateway-route-table`.

AWS CLI

Um eine lokale Gateway-Routentabelle zu erstellen

Im folgenden `create-local-gateway-route-table` Beispiel wird eine lokale Gateway-Routentabelle mit dem direkten VPC Routing-Modus erstellt.

```

aws ec2 create-local-gateway-route-table \
  --local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \
  --mode direct-vpc-routing

```

Ausgabe:

```

{
  "LocalGatewayRouteTable": {
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-gateway-route-table/lgw-rtb-abcdefg1234567890",
    "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
    "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/op-021345abcdef67890",
    "OwnerId": "111122223333",
    "State": "pending",
    "Tags": [],
    "Mode": "direct-vpc-routing"
  }
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Routing-Tabellen für lokale Gateways](#) im AWS -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateLocalGatewayRouteTable](#) unter AWS CLI Befehlsreferenz.

create-local-gateway-route

Das folgende Codebeispiel zeigt die Verwendung `create-local-gateway-route`.

AWS CLI

Um eine statische Route für eine lokale Gateway-Routentabelle zu erstellen

Im folgenden `create-local-gateway-route` Beispiel wird die angegebene Route in der angegebenen Routentabelle für das lokale Gateway erstellt.

```
aws ec2 create-local-gateway-route \  
  --destination-cidr-block 0.0.0.0/0 \  
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

Ausgabe:

```
{  
  "Route": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",  
    "Type": "static",  
    "State": "deleted",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"  
  }  
}
```

- API-Einheiten finden Sie [CreateLocalGatewayRoute](#) unter AWS CLI Befehlsreferenz.

create-managed-prefix-list

Das folgende Codebeispiel zeigt die Verwendung `create-managed-prefix-list`.

AWS CLI

Um eine Präfixliste zu erstellen

Im folgenden `create-managed-prefix-list` Beispiel wird eine IPv4 Präfixliste mit maximal 10 Einträgen und 2 Einträgen in der Präfixliste erstellt.

```
aws ec2 create-managed-prefix-list \  
  --address-family IPv4 \  
  --max-entries 10 \  
  --entries Cidr=10.0.0.0/16,Description=vpc-a Cidr=10.2.0.0/16,Description=vpc-b \  
 \  
  --prefix-list-name vpc-cidrs
```

Ausgabe:

```
{  
  "PrefixList": {  
    "PrefixListId": "pl-0123456abcabcabc1",  
    "AddressFamily": "IPv4",  
    "State": "create-in-progress",  
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabcabc1",  
    "PrefixListName": "vpc-cidrs",  
    "MaxEntries": 10,  
    "Version": 1,  
    "Tags": [],  
    "OwnerId": "123456789012"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateManagedPrefixList](#) in der AWS CLI Befehlsreferenz.

create-nat-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-nat-gateway`.

AWS CLI

Beispiel 1: Um ein öffentliches NAT Gateway zu erstellen

Das folgende `create-nat-gateway` Beispiel erstellt ein öffentliches NAT Gateway im angegebenen Subnetz und ordnet die Elastic IP-Adresse der angegebenen Zuweisungs-ID zu. Wenn Sie ein öffentliches NAT Gateway erstellen, müssen Sie eine Elastic IP-Adresse zuordnen.

```
aws ec2 create-nat-gateway \  
  --subnet-id subnet-0250c25a1fEXAMPLE \  
  --allocation-id eipalloc-09ad461b0dEXAMPLE
```

Ausgabe:

```
{  
  "NatGateway": {  
    "CreateTime": "2021-12-01T22:22:38.000Z",  
    "NatGatewayAddresses": [  
      {  
        "AllocationId": "eipalloc-09ad461b0dEXAMPLE"  
      }  
    ],  
    "NatGatewayId": "nat-0c61bf8a12EXAMPLE",  
    "State": "pending",  
    "SubnetId": "subnet-0250c25a1fEXAMPLE",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "ConnectivityType": "public"  
  }  
}
```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

Beispiel 2: Um ein privates NAT Gateway zu erstellen

Im folgenden `create-nat-gateway` Beispiel wird ein privates NAT Gateway im angegebenen Subnetz erstellt. Einem privaten NAT Gateway ist keine Elastic IP-Adresse zugeordnet.

```
aws ec2 create-nat-gateway \  
  --subnet-id subnet-0250c25a1fEXAMPLE \  
  --connectivity-type private
```

Ausgabe:

```
{  
  "NatGateway": {  
    "CreateTime": "2021-12-01T22:26:00.000Z",
```

```

    "NatGatewayAddresses": [
      {}
    ],
    "NatGatewayId": "nat-011b568379EXAMPLE",
    "State": "pending",
    "SubnetId": "subnet-0250c25a1fEXAMPLE",
    "VpcId": "vpc-0a60eb65b4EXAMPLE",
    "ConnectivityType": "private"
  }
}

```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateNatGateway](#) in der AWS CLI Befehlsreferenz.

create-network-acl-entry

Das folgende Codebeispiel zeigt die Verwendung `create-network-acl-entry`.

AWS CLI

Um einen ACL Netzwerkeintrag zu erstellen

In diesem Beispiel wird ein Eintrag für das angegebene Netzwerk erstellt. Die Regel erlaubt eingehenden Datenverkehr von einer beliebigen IPv4 Adresse (0.0.0.0/0) an UDP Port 53 (DNS) in jedes zugehörige Subnetz. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 0.0.0.0/0 --rule-action allow
```

In diesem Beispiel wird eine Regel für das angegebene Netzwerk erstellt, die eingehenden Datenverkehr von einer beliebigen IPv6 Adresse (::/0) an Port 80 () zulässt. TCP HTTP

Befehl:

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 120 --protocol tcp --port-range From=80,To=80 --ipv6-cidr-block ::/0 --rule-action allow
```

- API-Einheiten finden Sie unter [CreateNetworkAclEntry AWS CLI-Befehlsreferenz](#).

create-network-acl

Das folgende Codebeispiel zeigt die Verwendung `create-network-acl`.

AWS CLI

Um ein Netzwerk zu erstellen ACL

In diesem Beispiel wird ein Netzwerk ACL für das angegebene Objekt erstellt VPC.

Befehl:

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

Ausgabe:

```
{
  "NetworkAcl": {
    "Associations": [],
    "NetworkAclId": "acl-5fb85d36",
    "VpcId": "vpc-a01106c2",
    "Tags": [],
    "Entries": [
      {
        "CidrBlock": "0.0.0.0/0",
        "RuleNumber": 32767,
        "Protocol": "-1",
        "Egress": true,
        "RuleAction": "deny"
      },
      {
        "CidrBlock": "0.0.0.0/0",
        "RuleNumber": 32767,
        "Protocol": "-1",
        "Egress": false,
        "RuleAction": "deny"
      }
    ],
    "IsDefault": false
  }
}
```

```
}
```

- API-Einheiten finden Sie [CreateNetworkACL](#) in der AWS CLI Befehlsreferenz.

create-network-insights-access-scope

Das folgende Codebeispiel zeigt die Verwendung `create-network-insights-access-scope`.

AWS CLI

Um einen Netzwerkzugriffsbereich zu erstellen

Im folgenden `create-network-insights-access-scope` Beispiel wird ein Netzwerkzugriffsbereich erstellt.

```
aws ec2 create-network-insights-access-scope \  
  --cli-input-json file://access-scope-file.json
```

Inhalt von `access-scope-file.json`:

```
{  
  "MatchPaths": [  
    {  
      "Source": {  
        "ResourceStatement": {  
          "Resources": [  
            "vpc-abcd12e3"  
          ]  
        }  
      }  
    }  
  ],  
  "ExcludePaths": [  
    {  
      "Source": {  
        "ResourceStatement": {  
          "ResourceTypes": [  
            "AWS::EC2::InternetGateway"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
]
}
```

Ausgabe:

```
{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789abc01234",
    "CreateDate": "2022-01-25T19:20:28.796000+00:00",
    "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
  },
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "Resources": [
              "vpc-abcd12e3"
            ]
          }
        }
      }
    ],
    "ExcludePaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer unter Verwendung des AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [CreateNetworkInsightsAccessScope](#) unter AWS CLI Befehlsreferenz.

create-network-insights-path

Das folgende Codebeispiel zeigt die Verwendung `create-network-insights-path`.

AWS CLI

Um einen Pfad zu erstellen

Im folgenden `create-network-insights-path` Beispiel wird ein Pfad erstellt. Die Quelle ist das angegebene Internet-Gateway und das Ziel ist die angegebene EC2 Instanz. Um festzustellen, ob das Ziel mit dem angegebenen Protokoll und Port erreichbar ist, analysieren Sie den Pfad mithilfe des `start-network-insights-analysis` Befehls.

```
aws ec2 create-network-insights-path \  
  --source igw-0797cccdc9d73b0e5 \  
  --destination i-0495d385ad28331c7 \  
  --destination-port 22 \  
  --protocol TCP
```

Ausgabe:

```
{  
  "NetworkInsightsPaths": {  
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",  
    "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-  
insights-path/nip-0b26f224f1d131fa8",  
    "CreateDate": "2021-01-20T22:43:46.933Z",  
    "Source": "igw-0797cccdc9d73b0e5",  
    "Destination": "i-0495d385ad28331c7",  
    "Protocol": "tcp"  
  }  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS CLI](#) im Reachability Analyzer Guide.

- API-Einheiten finden Sie [CreateNetworkInsightsPath](#) in der AWS CLI Befehlsreferenz.

create-network-interface-permission

Das folgende Codebeispiel zeigt die Verwendung `create-network-interface-permission`.

AWS CLI

Um eine Netzwerkschnittstellenberechtigung zu erstellen

In diesem Beispiel wird einem Konto die Berechtigung erteilt `123456789012`, eine Netzwerkschnittstelle `eni-1a2b3c4d` an eine Instanz anzuhängen.

Befehl:

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --  
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

Ausgabe:

```
{  
  "InterfacePermission": {  
    "PermissionState": {  
      "State": "GRANTED"  
    },  
    "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",  
    "NetworkInterfaceId": "eni-1a2b3c4d",  
    "Permission": "INSTANCE-ATTACH",  
    "AwsAccountId": "123456789012"  
  }  
}
```

- API-Einheiten finden Sie [CreateNetworkInterfacePermission](#) in der AWS CLI Befehlsreferenz.

create-network-interface

Das folgende Codebeispiel zeigt die Verwendung `create-network-interface`.

AWS CLI

Beispiel 1: Um eine IPv4 Adresse für eine Netzwerkschnittstelle anzugeben

Im folgenden `create-network-interface` Beispiel wird eine Netzwerkschnittstelle für das angegebene Subnetz mit der angegebenen primären IPv4 Adresse erstellt.

```
aws ec2 create-network-interface \  
  --subnet-id subnet-00a24d0d67acf6333 \  
  --description "my network interface" \  
  --groups sg-09dfba7ed20cda78b \  
  --private-ip-address 10.0.8.17
```

Ausgabe:

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "my network interface",  
    "Groups": [  
      {  
        "GroupName": "my-security-group",  
        "GroupId": "sg-09dfba7ed20cda78b"  
      }  
    ],  
    "InterfaceType": "interface",  
    "Ipv6Addresses": [],  
    "MacAddress": "06:6a:0f:9a:49:37",  
    "NetworkInterfaceId": "eni-0492b355f0cf3b3f8",  
    "OwnerId": "123456789012",  
    "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",  
    "PrivateIpAddress": "10.0.8.17",  
    "PrivateIpAddresses": [  
      {  
        "Primary": true,  
        "PrivateDnsName": "ip-10-0-8-17.us-west-2.compute.internal",  
        "PrivateIpAddress": "10.0.8.17"  
      }  
    ],  
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",  
    "RequesterManaged": false,  
    "SourceDestCheck": true,  
    "Status": "pending",  
    "SubnetId": "subnet-00a24d0d67acf6333",  
    "TagSet": [],  
    "VpcId": "vpc-02723a0feeeb9d57b"  
  }  
}
```


Beispiel 2: Um eine Netzwerkschnittstelle mit einer IPv4 Adresse und einer IPv6 Adresse zu erstellen

Das folgende `create-network-interface` Beispiel erstellt eine Netzwerkschnittstelle für das angegebene Subnetz mit einer IPv4 Adresse und einer IPv6 Adresse, die von Amazon EC2 ausgewählt wurden.

```
aws ec2 create-network-interface \  
  --subnet-id subnet-00a24d0d67acf6333 \  
  --description "my dual stack network interface" \  
  --ipv6-address-count 1 \  
  --groups sg-09dfba7ed20cda78b
```

Ausgabe:

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "my dual stack network interface",  
    "Groups": [  
      {  
        "GroupName": "my-security-group",  
        "GroupId": "sg-09dfba7ed20cda78b"  
      }  
    ],  
    "InterfaceType": "interface",  
    "Ipv6Addresses": [  
      {  
        "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7",  
        "IsPrimaryIpv6": false  
      }  
    ],  
    "MacAddress": "06:b8:68:d2:b2:2d",  
    "NetworkInterfaceId": "eni-05da417453f9a84bf",  
    "OwnerId": "123456789012",  
    "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",  
    "PrivateIpAddress": "10.0.8.18",  
    "PrivateIpAddresses": [  
      {  
        "Primary": true,  
        "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",  
        "PrivateIpAddress": "10.0.8.18"  
      }  
    ]  
  }  
}
```

```

    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeeb9d57b",
    "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7"
  }
}

```

Beispiel 3: So erstellen Sie eine Netzwerkschnittstelle mit Konfigurationsoptionen für die Verbindungsverfolgung

Im folgenden `create-network-interface` Beispiel wird eine Netzwerkschnittstelle erstellt und die Timeouts für die Verbindungsverfolgung im Leerlauf konfiguriert.

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --groups sg-02e57dbcfe0331c1b \
  --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60

```

Ausgabe:

```

{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "ConnectionTrackingConfiguration": {
      "TcpEstablishedTimeout": 86400,
      "UdpTimeout": 60
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "my-security-group",
        "GroupId": "sg-02e57dbcfe0331c1b"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "06:4c:53:de:6d:91",

```

```

    "NetworkInterfaceId": "eni-0c133586e08903d0b",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
    "PrivateIpAddress": "10.0.8.94",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.94"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeb9d57b"
  }
}

```

Beispiel 4: So erstellen Sie einen Elastic Fabric-Adapter

Das folgende `create-network-interface` Beispiel erstellt eine EFA.

```

aws ec2 create-network-interface \
  --interface-type efa \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my efa" \
  --groups sg-02e57dbcf0331c1b

```

Ausgabe:

```

{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "Description": "my efa",
    "Groups": [
      {
        "GroupName": "my-efa-sg",
        "GroupId": "sg-02e57dbcf0331c1b"
      }
    ]
  },

```

```
"InterfaceType": "efa",
  "Ipv6Addresses": [],
  "MacAddress": "06:d7:a4:f7:4d:57",
  "NetworkInterfaceId": "eni-034acc2885e862b65",
  "OwnerId": "123456789012",
  "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
  "PrivateIpAddress": "10.0.8.180",
  "PrivateIpAddresses": [
    {
      "Primary": true,
      "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
      "PrivateIpAddress": "10.0.8.180"
    }
  ],
  "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
  "RequesterManaged": false,
  "SourceDestCheck": true,
  "Status": "pending",
  "SubnetId": "subnet-00a24d0d67acf6333",
  "TagSet": [],
  "VpcId": "vpc-02723a0feeeb9d57b"
}
```

Weitere Informationen finden Sie unter [Elastic Network Interfaces](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateNetworkInterface](#) in der AWS CLI Befehlsreferenz.

create-placement-group

Das folgende Codebeispiel zeigt die Verwendung `create-placement-group`.

AWS CLI

Um eine Platzierungsgruppe zu erstellen

Dieser Beispielbefehl erstellt eine Platzierungsgruppe mit dem angegebenen Namen.

Befehl:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

Um eine Platzierungsgruppe für Partitionen zu erstellen

Dieser Beispielbefehl erstellt eine Partitionsplatzierungsgruppe HDFS-Group-A mit dem Namen fünf Partitionen.

Befehl:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

- API-Einheiten finden Sie [CreatePlacementGroup](#) in der AWS CLI Befehlsreferenz.

create-replace-root-volume-task

Das folgende Codebeispiel zeigt die Verwendung `create-replace-root-volume-task`.

AWS CLI

Beispiel 1: Um ein Root-Volume in seinen ursprünglichen Startzustand zurückzusetzen

Im folgenden `create-replace-root-volume-task` Beispiel wird das Root-Volume der Instanz `i-0123456789abcdefa` in den ursprünglichen Startzustand zurückversetzt.

```
aws ec2 create-replace-root-volume-task \
  --instance-id i-0123456789abcdefa
```

Ausgabe:

```
{
  "ReplaceRootVolumeTask":
  {
    "InstanceId": "i-0123456789abcdefa",
    "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
    "TaskState": "pending",
    "StartTime": "2022-03-14T15:06:38Z",
    "Tags": []
  }
}
```

Weitere Informationen finden Sie unter [Ersetzen eines Root-Volumes](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.


```
--price-schedules CurrencyCode=USD,Price=25.50 \  
--client-token 550e8400-e29b-41d4-a716-446655440000
```

- API-Einheiten finden Sie [CreateReservedInstancesListing](#) unter AWS CLI Befehlsreferenz.

create-restore-image-task

Das folgende Codebeispiel zeigt die Verwendung `create-restore-image-task`.

AWS CLI

Um einen AMI aus einem S3-Bucket wiederherzustellen

Das folgende `create-restore-image-task` Beispiel stellt eine AMI aus einem S3-Bucket wieder her. Verwenden Sie die Werte für `S3ObjectKey` und `Bucket` aus der `describe-store-image-tasks` Ausgabe, geben Sie den Objektschlüssel AMI und den Namen des S3-Buckets an, in den der kopierte AMI wurde, und geben Sie den Namen für den wiederhergestellten Bucket an. Der Name muss AMIs in der Region für dieses Konto eindeutig sein. Das wiederhergestellte Objekt AMI erhält eine neue AMI ID.

```
aws ec2 create-restore-image-task \  
  --object-key ami-1234567890abcdef0.bin \  
  --bucket my-ami-bucket \  
  --name "New AMI Name"
```

Ausgabe:

```
{  
  "ImageId": "ami-0eab20fe36f83e1a8"  
}
```

Weitere Informationen zum Speichern und Wiederherstellen eines AMI mit S3 finden Sie unter [Speichern und Wiederherstellen eines AMI mit S3](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/ami-store-restore.html) <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/ami-store-restore.html> im Amazon-Benutzerhandbuch. EC2

- Einzelheiten API finden Sie in der Befehlsreferenz. [CreateRestoreImageTask](#) AWS CLI

create-route-table

Das folgende Codebeispiel zeigt die Verwendung `create-route-table`.

AWS CLI

So erstellen Sie eine Routing-Tabelle

In diesem Beispiel wird eine Routentabelle für das angegebene Objekt erstelltVPC.

Befehl:

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

Ausgabe:

```
{
  "RouteTable": {
    "Associations": [],
    "RouteTableId": "rtb-22574640",
    "VpcId": "vpc-a01106c2",
    "PropagatingVgws": [],
    "Tags": [],
    "Routes": [
      {
        "GatewayId": "local",
        "DestinationCidrBlock": "10.0.0.0/16",
        "State": "active"
      }
    ]
  }
}
```

- APIEinzelheiten finden Sie [CreateRouteTable](#)unter AWS CLI Befehlsreferenz.

create-route

Das folgende Codebeispiel zeigt die Verwendungcreate-route.

AWS CLI

Um eine Route zu erstellen

In diesem Beispiel wird eine Route für die angegebene Routentabelle erstellt. Die Route entspricht dem gesamten IPv4 Verkehr (0.0.0.0/0) und leitet ihn an das angegebene Internet-Gateway weiter. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

Dieser Beispielbefehl erstellt eine Route in der Routentabelle *rtb-g8ff4ea2*. Die Route entspricht dem Verkehr für den IPv4 CIDR Block *10.0.0.0/16* und leitet ihn an die Peering-Verbindung *pcx-111aaa22* weiter. VPC Diese Route ermöglicht die Weiterleitung des Datenverkehrs an den Peer in der Peering-Verbindung. VPC VPC Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

In diesem Beispiel wird in der angegebenen Routentabelle eine Route erstellt, die dem gesamten IPv6 Datenverkehr (*:::/0*) entspricht, und ihn an das angegebene Internet-Gateway weiterleitet, das nur für ausgehenden Datenverkehr bestimmt ist.

Befehl:

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-block :::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- APIEinzelheiten finden Sie unter Befehlsreferenz [CreateRoute](#).AWS CLI

create-security-group

Das folgende Codebeispiel zeigt die Verwendung `create-security-group`.

AWS CLI

Um eine Sicherheitsgruppe für EC2 -Classic zu erstellen

In diesem Beispiel wird eine Sicherheitsgruppe mit dem Namen `MySecurityGroup` erstellt.

Befehl:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

Ausgabe:

```
{
  "GroupId": "sg-903004f8"
}
```

Um eine Sicherheitsgruppe für EC2 - zu erstellen VPC

In diesem Beispiel wird eine Sicherheitsgruppe erstellt, die MySecurityGroup nach der angegebenen benannt istVPC.

Befehl:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group" --vpc-id vpc-1a2b3c4d
```

Ausgabe:

```
{
  "GroupId": "sg-903004f8"
}
```

Weitere Informationen finden Sie unter Verwenden von Sicherheitsgruppen im Benutzerhandbuch für die AWS -Befehlszeilenschnittstelle.

- APIEinzelheiten finden Sie [CreateSecurityGroup](#)unter AWS CLI Befehlsreferenz.

create-snapshot

Das folgende Codebeispiel zeigt die Verwendungcreate-snapshot.

AWS CLI

Um einen Snapshot zu erstellen

Dieser Beispielbefehl erstellt einen Snapshot des Volumes mit der Volume-ID vol-1234567890abcdef0 und einer kurzen Beschreibung zur Identifizierung des Snapshots.

Befehl:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot"
```

Ausgabe:

```
{
  "Description": "This is my root volume snapshot",
  "Tags": [],
  "Encrypted": false,
  "VolumeId": "vol-1234567890abcdef0",
  "State": "pending",
  "VolumeSize": 8,
  "StartTime": "2018-02-28T21:06:01.000Z",
  "Progress": "",
  "OwnerId": "012345678910",
  "SnapshotId": "snap-066877671789bd71b"
}
```

Um einen Snapshot mit Tags zu erstellen

Dieser Beispielbefehl erstellt einen Snapshot und wendet zwei Tags an: purpose=prod und costcenter=123.

Befehl:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description 'Prod backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},{Key=costcenter,Value=123}]'
```

Ausgabe:

```
{
  "Description": "Prod backup",
  "Tags": [
    {
      "Value": "prod",
      "Key": "purpose"
    },
    {
      "Value": "123",
      "Key": "costcenter"
    }
  ],
  "Encrypted": false,
  "VolumeId": "vol-1234567890abcdef0",
  "State": "pending",
}
```

```

    "VolumeSize": 8,
    "StartTime": "2018-02-28T21:06:06.000Z",
    "Progress": "",
    "OwnerId": "012345678910",
    "SnapshotId": "snap-09ed24a70bc19bbe4"
  }

```

- Einzelheiten API finden Sie in der Befehlsreferenz. [CreateSnapshot](#) AWS CLI

create-snapshots

Das folgende Codebeispiel zeigt die Verwendung `create-snapshots`.

AWS CLI

Beispiel 1: Um einen Snapshot mit mehreren Volumes zu erstellen

Im folgenden `create-snapshots` Beispiel werden Snapshots aller Volumes erstellt, die an die angegebene Instanz angehängt sind.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --description "This is snapshot of a volume from my-instance"

```

Ausgabe:

```

{
  "Snapshots": [
    {
      "Description": "This is a snapshot of a volume from my-instance",
      "Tags": [],
      "Encrypted": false,
      "VolumeId": "vol-0a01d2d5a34697479",
      "State": "pending",
      "VolumeSize": 16,
      "StartTime": "2019-08-05T16:58:19.000Z",
      "Progress": "",
      "OwnerId": "123456789012",
      "SnapshotId": "snap-07f30e3909aa0045e"
    },
    {
      "Description": "This is a snapshot of a volume from my-instance",

```

```

    "Tags": [],
    "Encrypted": false,
    "VolumeId": "vol-02d0d4947008cb1a2",
    "State": "pending",
    "VolumeSize": 20,
    "StartTime": "2019-08-05T16:58:19.000Z",
    "Progress": "",
    "OwnerId": "123456789012",
    "SnapshotId": "snap-0ec20b602264aad48"
  },
  ...
]
}

```

Beispiel 2: Um einen Snapshot mit mehreren Volumes mit Tags aus dem Quellvolume zu erstellen

Das folgende `create-snapshots` Beispiel erstellt Snapshots aller Volumes, die an die angegebene Instance angehängt sind, und kopiert die Tags von jedem Volume in den entsprechenden Snapshot.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --copy-tags-from-source volume \
  --description "This is snapshot of a volume from my-instance"

```

Ausgabe:

```

{
  "Snapshots": [
    {
      "Description": "This is a snapshot of a volume from my-instance",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-volume"
        }
      ],
      "Encrypted": false,
      "VolumeId": "vol-02d0d4947008cb1a2",
      "State": "pending",
      "VolumeSize": 20,
      "StartTime": "2019-08-05T16:53:04.000Z",

```

```

        "Progress": "",
        "OwnerId": "123456789012",
        "SnapshotId": "snap-053bfaeb821a458dd"
    }
    ...
]
}

```

Beispiel 3: Um einen Snapshot mit mehreren Volumes ohne das Root-Volume zu erstellen

Im folgenden `create-snapshots` Beispiel wird ein Snapshot aller Volumes erstellt, die an die angegebene Instance angehängt sind, mit Ausnahme des Root-Volumes.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true

```

Eine Beispielausgabe finden Sie in Beispiel 1.

Beispiel 4: Um einen Snapshot mit mehreren Volumes zu erstellen und Tags hinzuzufügen

Das folgende `create-snapshots` Beispiel erstellt Snapshots aller Volumes, die an die angegebene Instance angehängt sind, und fügt jedem Snapshot zwei Tags hinzu.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},
{Key=costcenter,Value=123}]'

```

Eine Beispielausgabe finden Sie in Beispiel 1.

- API-Einheiten finden Sie [CreateSnapshots](#) in der AWS CLI Befehlsreferenz.

create-spot-datafeed-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-spot-datafeed-subscription`.

AWS CLI

Um einen Spot-Instance-Datenfeed zu erstellen

Im folgenden `create-spot-datafeed-subscription` Beispiel wird ein Spot-Instance-Datenfeed erstellt.

```
aws ec2 create-spot-datafeed-subscription \  
  --bucket my-bucket \  
  --prefix spot-data-feed
```

Ausgabe:

```
{  
  "SpotDatafeedSubscription": {  
    "Bucket": "my-bucket",  
    "OwnerId": "123456789012",  
    "Prefix": "spot-data-feed",  
    "State": "Active"  
  }  
}
```

Der Datenfeed wird in dem Amazon S3 S3-Bucket gespeichert, den Sie angegeben haben. Die Dateinamen für diesen Datenfeed haben das folgende Format.

```
my-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-HH.n.abcd1234.gz
```

Weitere Informationen finden Sie unter [Spot-Instance-Datenfeed](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [CreateSpotDatafeedSubscription](#) in der AWS CLI Befehlsreferenz.

create-store-image-task

Das folgende Codebeispiel zeigt die Verwendung `create-store-image-task`.

AWS CLI

Um einen AMI in einem S3-Bucket zu speichern

Im folgenden `create-store-image-task` Beispiel wird ein AMI in einem S3-Bucket gespeichert. Geben Sie die ID AMI und den Namen des S3-Buckets an, in dem der gespeichert werden soll AMI.

```
aws ec2 create-store-image-task \  
  --image-id ami-1234567890abcdef0 \  
  --bucket my-ami-bucket
```

Ausgabe:

```
{
  "ObjectKey": "ami-1234567890abcdef0.bin"
}
```

Weitere Informationen finden Sie unter [Speichern und Wiederherstellen AMI mit S3](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CreateStoreImageTask](#) in der AWS CLI Befehlsreferenz.

create-subnet-cidr-reservation

Das folgende Codebeispiel zeigt die Verwendung `create-subnet-cidr-reservation`.

AWS CLI

Um eine CIDR Subnetzreservierung zu erstellen

Im folgenden `create-subnet-cidr-reservation` Beispiel wird eine CIDR Subnetzreservierung für das angegebene Subnetz und den angegebenen Bereich erstellt. CIDR

```
aws ec2 create-subnet-cidr-reservation \
  --subnet-id subnet-03c51e2eEXAMPLE \
  --reservation-type prefix \
  --cidr 10.1.0.20/26
```

Ausgabe:

```
{
  "SubnetCidrReservation": {
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",
    "Cidr": "10.1.0.16/28",
    "ReservationType": "prefix",
    "OwnerId": "123456789012"
  }
}
```

Weitere Informationen finden Sie unter [CIDRSubnetzreservierungen](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSubnetCidrReservation](#) in der AWS CLI Befehlsreferenz.

create-subnet

Das folgende Codebeispiel zeigt die Verwendung `create-subnet`.

AWS CLI

Beispiel 1: Um ein Subnetz nur mit einem IPv4 CIDR Block zu erstellen

Das folgende `create-subnet` Beispiel erstellt ein Subnetz im angegebenen Bereich VPC mit dem angegebenen IPv4 CIDR Block.

```
aws ec2 create-subnet \  
  --vpc-id vpc-081ec835f3EXAMPLE \  
  --cidr-block 10.0.0.0/24 \  
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

Ausgabe:

```
{  
  "Subnet": {  
    "AvailabilityZone": "us-west-2a",  
    "AvailabilityZoneId": "usw2-az2",  
    "AvailableIpAddressCount": 251,  
    "CidrBlock": "10.0.0.0/24",  
    "DefaultForAz": false,  
    "MapPublicIpOnLaunch": false,  
    "State": "available",  
    "SubnetId": "subnet-0e99b93155EXAMPLE",  
    "VpcId": "vpc-081ec835f3EXAMPLE",  
    "OwnerId": "123456789012",  
    "AssignIpv6AddressOnCreation": false,  
    "Ipv6CidrBlockAssociationSet": [],  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-ipv4-only-subnet"  
      }  
    ],  
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0e99b93155EXAMPLE"
```

```
}
}
```

Beispiel 2: Um ein Subnetz mit Blöcken sowohl als auch IPv4 zu erstellen IPv6 CIDR

Im folgenden `create-subnet` Beispiel wird ein Subnetz im angegebenen Bereich VPC mit den angegebenen Blöcken IPv4 und IPv6 CIDR erstellt.

```
aws ec2 create-subnet \
  --vpc-id vpc-081ec835f3EXAMPLE \
  --cidr-block 10.0.0.0/24 \
  --ipv6-cidr-block 2600:1f16:cfe:3660::/64 \
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-ipv6-  
subnet}]
```

Ausgabe:

```
{
  "Subnet": {
    "AvailabilityZone": "us-west-2a",
    "AvailabilityZoneId": "usw2-az2",
    "AvailableIpAddressCount": 251,
    "CidrBlock": "10.0.0.0/24",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-0736441d38EXAMPLE",
    "VpcId": "vpc-081ec835f3EXAMPLE",
    "OwnerId": "123456789012",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "subnet-cidr-assoc-06c5f904499fcc623",
        "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
        "Ipv6CidrBlockState": {
          "State": "associating"
        }
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-ipv4-ipv6-subnet"
      }
    ]
  }
}
```

```

    }
  ],
  "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0736441d38EXAMPLE"
}
}

```

Beispiel 3: Um ein Subnetz nur mit einem IPv6 CIDR Block zu erstellen

Das folgende `create-subnet` Beispiel erstellt ein Subnetz im angegebenen Bereich VPC mit dem angegebenen IPv6 CIDR Block.

```

aws ec2 create-subnet \
  --vpc-id vpc-081ec835f3EXAMPLE \
  --ipv6-native \
  --ipv6-cidr-block 2600:1f16:115:200::/64 \
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv6-only-
subnet}]

```

Ausgabe:

```

{
  "Subnet": {
    "AvailabilityZone": "us-west-2a",
    "AvailabilityZoneId": "usw2-az2",
    "AvailableIpAddressCount": 0,
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-03f720e7deEXAMPLE",
    "VpcId": "vpc-081ec835f3EXAMPLE",
    "OwnerId": "123456789012",
    "AssignIpv6AddressOnCreation": true,
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "subnet-cidr-assoc-01ef639edde556709",
        "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
        "Ipv6CidrBlockState": {
          "State": "associating"
        }
      }
    ]
  },
  "Tags": [

```

```

        {
            "Key": "Name",
            "Value": "my-ipv6-only-subnet"
        }
    ],
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-03f720e7deEXAMPLE"
}
}

```

Weitere Informationen finden Sie unter [VPCs und Subnetze](#) im VPC Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSubnet](#) in der AWS CLI Befehlsreferenz.

create-tags

Das folgende Codebeispiel zeigt die Verwendung `create-tags`.

AWS CLI

Beispiel 1: Um einer Ressource ein Tag hinzuzufügen

Das folgende `create-tags` Beispiel fügt das Tag dem angegebenen Bild `Stack=production` hinzu oder überschreibt ein vorhandenes Tag für den AMI Ort, an dem sich der Tag-Schlüssel befindet `Stack`.

```

aws ec2 create-tags \
  --resources ami-1234567890abcdef0 \
  --tags Key=Stack,Value=production

```

Weitere Informationen finden Sie unter [Dies ist der Thementitel](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: Um mehreren Ressourcen Tags hinzuzufügen

Im folgenden `create-tags` Beispiel werden zwei Tags für eine Instanz AMI und eine Instanz hinzugefügt (oder überschrieben). Eines der Tags hat einen Schlüssel (`webserver`), aber keinen Wert (Wert ist auf eine leere Zeichenfolge festgelegt). Das andere Tag hat einen Schlüssel (`stack`) und einen Wert (`Production`).

```

aws ec2 create-tags \
  --resources ami-1a2b3c4d i-1234567890abcdef0 \

```

```
--tags Key=webserver,Value= Key=stack,Value=Production
```

Weitere Informationen finden Sie unter [Dies ist der Thementitel](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 3: Um Tags hinzuzufügen, die Sonderzeichen enthalten

Das folgende `create-tags`-Beispiel fügt das Tag `[Group]=test` für eine Instance hinzu. Die eckigen Klammern (`[` und `]`) sind Sonderzeichen und müssen mit Escape-Zeichen versehen werden. In den folgenden Beispielen wird auch das Zeilenfortsetzungszeichen verwendet, das für jede Umgebung geeignet ist.

Wenn Sie Windows verwenden, schließen Sie das Element, das Sonderzeichen enthält, in doppelte Anführungszeichen („“) ein und stellen Sie jedem doppelten Anführungszeichen wie folgt einen umgekehrten Schrägstrich (`\`) voran:

```
aws ec2 create-tags ^  
  --resources i-1234567890abcdef0 ^  
  --tags Key=\ "[Group]" ,Value=test
```

Wenn Sie Windows verwenden PowerShell, setzen Sie für das Element den Wert, der Sonderzeichen enthält, doppelte Anführungszeichen („“), stellen Sie jedem doppelten Anführungszeichen einen umgekehrten Schrägstrich (`\`) voran und setzen Sie dann die gesamte Schlüssel- und Wertstruktur wie folgt in einfache Anführungszeichen (‘) :

```
aws ec2 create-tags `   
  --resources i-1234567890abcdef0 `   
  --tags 'Key=\ "[Group]" ,Value=test'
```

Wenn Sie Linux oder OS X verwenden, schließen Sie das Element mit den Sonderzeichen mit doppelten Anführungszeichen („“) ein und dann die gesamte Schlüssel- und Wertstruktur mit einfachen Anführungszeichen (‘) wie folgt:

```
aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags 'Key="[Group]" ,Value=test'
```

Weitere Informationen finden Sie unter [Dies ist der Thementitel](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [CreateTags](#) in der AWS CLI Befehlsreferenz.

create-traffic-mirror-filter-rule

Das folgende Codebeispiel zeigt die Verwendung `create-traffic-mirror-filter-rule`.

AWS CLI

Um eine Filterregel für eingehenden TCP Datenverkehr zu erstellen

Im folgenden `create-traffic-mirror-filter-rule` Beispiel wird eine Regel erstellt, mit der Sie den gesamten eingehenden TCP Datenverkehr spiegeln können. Verwenden Sie, um den Traffic Mirror-Filter `create-traffic-mirror-filter` zu erstellen, bevor Sie diesen Befehl ausführen.

```
aws ec2 create-traffic-mirror-filter-rule \  
  --description "TCP Rule" \  
  --destination-cidr-block 0.0.0.0/0 \  
  --protocol 6 \  
  --rule-action accept \  
  --rule-number 1 \  
  --source-cidr-block 0.0.0.0/0 \  
  --traffic-direction ingress \  
  --traffic-mirror-filter-id tmf-04812ff784b25ae67
```

Ausgabe:

```
{  
  "TrafficMirrorFilterRule": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "TrafficMirrorFilterId": "tmf-04812ff784b25ae67",  
    "TrafficMirrorFilterRuleId": "tmfr-02d20d996673f3732",  
    "SourceCidrBlock": "0.0.0.0/0",  
    "TrafficDirection": "ingress",  
    "Description": "TCP Rule",  
    "RuleNumber": 1,  
    "RuleAction": "accept",  
    "Protocol": 6  
  },  
  "ClientToken": "4752b573-40a6-4eac-a8a4-a72058761219"  
}
```

Weitere Informationen finden Sie im Traffic [Mirroring-Handbuch unter Erstellen eines AWS Traffic Mirroring-Filters](#).

- API-Einheiten finden Sie unter [CreateTrafficMirrorFilterRule AWS CLI-Befehlsreferenz](#).

create-traffic-mirror-filter

Das folgende Codebeispiel zeigt die Verwendung `create-traffic-mirror-filter`.

AWS CLI

Um einen Traffic Mirror Filter zu erstellen

Im folgenden `create-traffic-mirror-filter` Beispiel wird ein Traffic Mirror-Filter erstellt. Nachdem Sie den Filter erstellt haben, verwenden Sie `create-traffic-mirror-filter-rule` ihn, um dem Filter Regeln hinzuzufügen.

```
aws ec2 create-traffic-mirror-filter \  
  --description "TCP Filter"
```

Ausgabe:

```
{  
  "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",  
  "TrafficMirrorFilter": {  
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",  
    "Description": "TCP Filter",  
    "EgressFilterRules": [],  
    "IngressFilterRules": [],  
    "Tags": [],  
    "NetworkServices": []  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Traffic Mirroring-Filters](#) im AWS Traffic Mirroring Guide.

- API-Einheiten finden Sie unter [CreateTrafficMirrorFilter AWS CLI-Befehlsreferenz](#).

create-traffic-mirror-session

Das folgende Codebeispiel zeigt die Verwendung `create-traffic-mirror-session`.

AWS CLI

Um eine Traffic Mirror-Sitzung zu erstellen

Mit dem folgenden `create-traffic-mirror-session` Befehl werden Traffic Mirror-Sitzungen für die angegebene Quelle und das angegebene Ziel für 25 Byte des Pakets erstellt.

```
aws ec2 create-traffic-mirror-session \  
  --description "example session" \  
  --traffic-mirror-target-id tmt-07f75d8feeEXAMPLE \  
  --network-interface-id eni-070203f901EXAMPLE \  
  --session-number 1 \  
  --packet-length 25 \  
  --traffic-mirror-filter-id tmf-04812ff784EXAMPLE
```

Ausgabe:

```
{  
  "TrafficMirrorSession": {  
    "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",  
    "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",  
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",  
    "NetworkInterfaceId": "eni-070203f901EXAMPLE",  
    "OwnerId": "111122223333",  
    "PacketLength": 25,  
    "SessionNumber": 1,  
    "VirtualNetworkId": 7159709,  
    "Description": "example session",  
    "Tags": []  
  },  
  "ClientToken": "5236cffc-ee13-4a32-bb5b-388d9da09d96"  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Traffic Mirror-Sitzung](#) im AWS Traffic Mirroring Guide.

- API-Einheiten finden Sie unter [CreateTrafficMirrorSession AWS CLI Befehlsreferenz](#).

create-traffic-mirror-target

Das folgende Codebeispiel zeigt die Verwendung `create-traffic-mirror-target`.

AWS CLI

So erstellen Sie ein Network Load Balancer Traffic Mirror-Ziel

Im folgenden `create-traffic-mirror-target` Beispiel wird ein Network Load Balancer Traffic Mirror-Ziel erstellt.

```
aws ec2 create-traffic-mirror-target \  
  --description "Example Network Load Balancer Target" \  
  --network-load-balancer-arn arn:aws:elasticloadbalancing:us-  
east-1:111122223333:loadbalancer/net/NLB/7cdec873EXAMPLE
```

Ausgabe:

```
{  
  "TrafficMirrorTarget": {  
    "Type": "network-load-balancer",  
    "Tags": [],  
    "Description": "Example Network Load Balancer Target",  
    "OwnerId": "111122223333",  
    "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
east-1:724145273726:loadbalancer/net/NLB/7cdec873EXAMPLE",  
    "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE"  
  },  
  "ClientToken": "d5c090f5-8a0f-49c7-8281-72c796a21f72"  
}
```

Um ein Netzwerk-Traffic-Mirror-Ziel zu erstellen

Im folgenden `create-traffic-mirror-target` Beispiel wird ein Traffic Mirror-Ziel für die Netzwerkschnittstelle erstellt.

```
aws ec2 create-traffic-mirror-target --description „Ziel der Netzwerkschnittstelle“ -- eni-  
eni-01f6f631e network-interface-id EXAMPLE
```

Ausgabe:

```
{  
  "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",  
  "TrafficMirrorTarget": {  
    "Description": "Network interface target",  
    "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",
```

```

    "TrafficMirrorTargetId": "tmt-02dcdb2abEXAMPLE",
    "OwnerId": "111122223333",
    "Type": "network-interface",
    "Tags": []
  }
}

```

[Weitere Informationen finden Sie im Traffic Mirroring Guide unter Create a Traffic Mirroring Target.AWS](#)

- API-Einheiten finden Sie unter [CreateTrafficMirrorTarget AWS CLI-Befehlsreferenz](#).

create-transit-gateway-connect-peer

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-connect-peer`.

AWS CLI

So erstellen Sie einen Transit Gateway Connect-Peer

Im folgenden `create-transit-gateway-connect-peer` Beispiel wird ein Connect-Peer erstellt.

```

aws ec2 create-transit-gateway-connect-peer \
  --transit-gateway-attachment-id tgw-attach-0f0927767cEXAMPLE \
  --peer-address 172.31.1.11 \
  --inside-cidr-blocks 169.254.6.0/29

```

Ausgabe:

```

{
  "TransitGatewayConnectPeer": {
    "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
    "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
    "State": "pending",
    "CreationTime": "2021-10-13T03:35:17.000Z",
    "ConnectPeerConfiguration": {
      "TransitGatewayAddress": "10.0.0.234",
      "PeerAddress": "172.31.1.11",
      "InsideCidrBlocks": [
        "169.254.6.0/29"
      ]
    }
  },
}

```

```

    "Protocol": "gre",
    "BgpConfigurations": [
      {
        "TransitGatewayAsn": 64512,
        "PeerAsn": 64512,
        "TransitGatewayAddress": "169.254.6.2",
        "PeerAddress": "169.254.6.1",
        "BgpStatus": "down"
      },
      {
        "TransitGatewayAsn": 64512,
        "PeerAsn": 64512,
        "TransitGatewayAddress": "169.254.6.3",
        "PeerAddress": "169.254.6.1",
        "BgpStatus": "down"
      }
    ]
  }
}
}
}

```

Weitere Informationen finden Sie unter [Transit Gateway Connect-Anlagen und Transit Gateway Connect-Peers](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [CreateTransitGatewayConnectPeer AWS CLI Befehlsreferenz](#).

create-transit-gateway-connect

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-connect`.

AWS CLI

So erstellen Sie einen Transit Gateway Connect-Anhang

Im folgenden `create-transit-gateway-connect` Beispiel wird ein Connect-Anhang mit dem Protokoll „gre“ für den angegebenen Anhang erstellt.

```

aws ec2 create-transit-gateway-connect \
  --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
  --options "Protocol=gre"

```

Ausgabe:

```
{
  "TransitGatewayConnect": {
    "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
    "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "State": "pending",
    "CreationTime": "2021-03-09T19:59:17+00:00",
    "Options": {
      "Protocol": "gre"
    }
  }
}
```

Weitere Informationen finden Sie unter [Transit Gateway Connect-Anlagen und Transit Gateway Connect-Peers](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [CreateTransitGatewayConnect AWS CLI Befehlsreferenz](#).

create-transit-gateway-multicast-domain

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-multicast-domain`.

AWS CLI

Beispiel 1: Um eine IGMP Multicast-Domäne zu erstellen

Im folgenden `create-transit-gateway-multicast-domain` Beispiel wird eine Multicast-Domäne für das angegebene Transit-Gateway erstellt. Wenn statische Quellen deaktiviert sind, können alle Instances in Subnetzen, die der Multicast-Domäne zugeordnet sind, Multicast-Verkehr senden. Wenn mindestens ein Mitglied das IGMP Protokoll verwendet, müssen Sie die Unterstützung aktivieren. IGMPv2

```
aws ec2 create-transit-gateway-multicast-domain \
  --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
  --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

Ausgabe:

```
{
```

```

    "TransitGatewayMulticastDomain": {
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",
      "TransitGatewayId": "tgw-0bf0bfffefaEXAMPLE",
      "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-0c9e29e2a7EXAMPLE",
      "OwnerId": "123456789012",
      "Options": {
        "Icmpv2Support": "enable",
        "StaticSourcesSupport": "disable",
        "AutoAcceptSharedAssociations": "disable"
      },
      "State": "pending",
      "CreationTime": "2021-09-29T22:17:13.000Z"
    }
  }
}

```

Beispiel 2: So erstellen Sie eine statische Multicast-Domäne

Im folgenden `create-transit-gateway-multicast-domain` Beispiel wird eine Multicast-Domäne für das angegebene Transit-Gateway erstellt. Wenn statische Quellen aktiviert sind, müssen Sie Quellen statisch hinzufügen.

```

aws ec2 create-transit-gateway-multicast-domain \
  --transit-gateway-id tgw-0bf0bfffefaEXAMPLE \
  --options StaticSourcesSupport=enable,Icmpv2Support=disable

```

Ausgabe:

```

{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bfffefaEXAMPLE",
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
    "OwnerId": "123456789012",
    "Options": {
      "Icmpv2Support": "disable",
      "StaticSourcesSupport": "enable",
      "AutoAcceptSharedAssociations": "disable"
    },
  },
}

```

```

    "State": "pending",
    "CreationTime": "2021-09-29T22:20:19.000Z"
  }
}

```

Weitere Informationen finden Sie unter [Verwaltung von Multicast-Domänen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie [CreateTransitGatewayMulticastDomain](#) in der AWS CLI Befehlsreferenz.

create-transit-gateway-peering-attachment

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-peering-attachment`.

AWS CLI

Um einen Transit-Gateway-Peering-Anhang zu erstellen

Im folgenden `create-transit-gateway-peering-attachment` Beispiel wird eine Peering-Verbindungsanforderung zwischen den beiden angegebenen Transit-Gateways erstellt.

```

aws ec2 create-transit-gateway-peering-attachment \
  --transit-gateway-id tgw-123abc05e04123abc \
  --peer-transit-gateway-id tgw-11223344aabbcc112 \
  --peer-account-id 123456789012 \
  --peer-region us-east-2

```

Ausgabe:

```

{
  "TransitGatewayPeeringAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
    "RequesterTgwInfo": {
      "TransitGatewayId": "tgw-123abc05e04123abc",
      "OwnerId": "123456789012",
      "Region": "us-west-2"
    },
    "AcceptorTgwInfo": {
      "TransitGatewayId": "tgw-11223344aabbcc112",

```

```

        "OwnerId": "123456789012",
        "Region": "us-east-2"
    },
    "State": "initiatingRequest",
    "CreationTime": "2019-12-09T11:38:05.000Z"
}
}

```

Weitere Informationen finden Sie unter [Transit Gateway Peering Attachments](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Create Transit Gateway Peering Attachment AWS CLI Befehlsreferenz](#).

create-transit-gateway-policy-table

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-policy-table`.

AWS CLI

Um eine Richtlinientabelle für das Transit Gateway zu erstellen

Im folgenden `create-transit-gateway-policy-table` Beispiel wird eine Richtlinientabelle für das Transit-Gateway für das angegebene Transit-Gateway erstellt.

```

aws ec2 create-transit-gateway-policy-table \
  --transit-gateway-id tgw-067f8505c18f0bd6e

```

Ausgabe:

```

{
  "TransitGatewayPolicyTable": {
    "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
    "TransitGatewayId": "tgw-067f8505c18f0bd6e",
    "State": "pending",
    "CreationTime": "2023-11-28T16:36:43+00:00"
  }
}

```

Weitere Informationen finden Sie in den [Richtlinientabellen für Transit Gateway](#) im Transit Gateway-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateTransitGatewayPolicyTable](#) unter AWS CLI Befehlsreferenz.

create-transit-gateway-prefix-list-reference

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-prefix-list-reference`.

AWS CLI

Um einen Verweis auf eine Präfixliste zu erstellen

Im folgenden `create-transit-gateway-prefix-list-reference` Beispiel wird ein Verweis auf die angegebene Präfixliste in der angegebenen Transit-Gateway-Routentabelle erstellt.

```
aws ec2 create-transit-gateway-prefix-list-reference \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --prefix-list-id pl-11111122222222333 \
  --transit-gateway-attachment-id tgw-attach-aaaaaabbbbb11111
```

Ausgabe:

```
{
  "TransitGatewayPrefixListReference": {
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-11111122222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "pending",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aaaaaabbbbb11111",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}
```

Weitere Informationen finden Sie unter [Referenzen zur Präfixliste](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [CreateTransitGatewayPrefixListReference AWS CLI](#) Befehlsreferenz.

create-transit-gateway-route-table

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-route-table`.

AWS CLI

So erstellen Sie eine Transit Gateway Gateway-Routentabelle

Im folgenden `create-transit-gateway-route-table` Beispiel wird eine Routentabelle für das angegebene Transit-Gateway erstellt.

```
aws ec2 create-transit-gateway-route-table \  
--transit-gateway-id tgw-0262a0e521EXAMPLE
```

Ausgabe:

```
{  
  "TransitGatewayRouteTable": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",  
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",  
    "State": "pending",  
    "DefaultAssociationRouteTable": false,  
    "DefaultPropagationRouteTable": false,  
    "CreationTime": "2019-07-10T19:01:46.000Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Transit-Gateway-Routentabelle](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [CreateTransitGatewayRouteTable AWS CLI Befehlsreferenz](#).

create-transit-gateway-route

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-route`.

AWS CLI

Um eine Transit-Gateway-Route zu erstellen

Im folgenden `create-transit-gateway-route` Beispiel wird eine Route mit dem angegebenen Ziel für die angegebene Routentabelle erstellt.

```
aws ec2 create-transit-gateway-route \  
  --destination-cidr-block 10.0.2.0/24 \  
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Ausgabe:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  
      {  
        "ResourceId": "vpc-0065acced4EXAMPLE",  
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
        "ResourceType": "vpc"  
      }  
    ],  
    "Type": "static",  
    "State": "active"  
  }  
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [CreateTransitGatewayRoute AWS CLI Befehlsreferenz](#).

create-transit-gateway-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway-vpc-attachment`.

AWS CLI

Beispiel 1: Um ein Transit-Gateway einem zuzuordnen VPC

Im folgenden `create-transit-gateway-vpc-attachment` Beispiel wird ein Transit-Gateway-Anhang zum angegebenen Objekt erstellt VPC.

```
aws ec2 create-transit-gateway-vpc-attachment \  
  --transit-gateway-id tgw-0262a0e521EXAMPLE \  
  --vpc-id vpc-07e8ffd50f49335df \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

```
--subnet-id subnet-0752213d59EXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "VpcId": "vpc-07e8ffd50fEXAMPLE",
    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
      "subnet-0752213d59EXAMPLE"
    ],
    "CreationTime": "2019-07-10T17:33:46.000Z",
    "Options": {
      "DnsSupport": "enable",
      "Ipv6Support": "disable"
    }
  }
}
```

Weitere Informationen finden Sie unter [Erstellen eines Transit-Gateway-Anhangs zu einem VPC](#) im Transit Gateways Guide.

Beispiel 2: Um ein Transit-Gateway mehreren Subnetzen in einem zuzuordnen VPC

Im folgenden `create-transit-gateway-vpc-attachment` Beispiel wird ein Transit-Gateway-Anhang zu den angegebenen VPC Subnetzen erstellt.

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-02f776b1a7EXAMPLE \
  --vpc-id vpc-3EXAMPLE \
  --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

Ausgabe:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0e141e0bebEXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "VpcId": "vpc-3EXAMPLE",
```

```

    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
        "subnet-6EXAMPLE",
        "subnet-dEXAMPLE"
    ],
    "CreationTime": "2019-12-17T20:07:52.000Z",
    "Options": {
        "DnsSupport": "enable",
        "Ipv6Support": "disable"
    }
}
}

```

Weitere Informationen finden Sie unter [Erstellen eines Transit-Gateway-Anhangs zu einem VPC](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [CreateTransitGatewayVpcAttachment AWS CLI Befehlsreferenz](#).

create-transit-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-transit-gateway`.

AWS CLI

Um ein Transit-Gateway zu erstellen

Im folgenden `create-transit-gateway` Beispiel wird ein Transit-Gateway erstellt.

```

aws ec2 create-transit-gateway \
  --description MyTGW \
  --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=

```

Ausgabe:

```

{
  "TransitGateway": {
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-gateway/tgw-0262a0e521EXAMPLE",
    "State": "pending",
  }
}

```

```

    "OwnerId": "111122223333",
    "Description": "MyTGW",
    "CreationTime": "2019-07-10T14:02:12.000Z",
    "Options": {
      "AmazonSideAsn": 64516,
      "AutoAcceptSharedAttachments": "enable",
      "DefaultRouteTableAssociation": "enable",
      "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
      "DefaultRouteTablePropagation": "enable",
      "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
      "VpnEcmpSupport": "enable",
      "DnsSupport": "enable"
    }
  }
}

```

Weitere Informationen finden Sie unter [Create a Transit Gateway](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Create Transit Gateway AWS CLI Befehlsreferenz](#).

create-verified-access-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-verified-access-endpoint`.

AWS CLI

Um einen Verified Access-Endpunkt zu erstellen

Im folgenden `create-verified-access-endpoint` Beispiel wird ein Verified Access-Endpunkt für die angegebene Verified Access-Gruppe erstellt. Die angegebene Netzwerkschnittstelle und die Sicherheitsgruppe müssen derselben VPC Gruppe angehören.

```

aws ec2 create-verified-access-endpoint \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --endpoint-type network-interface \
  --attachment-type vpc \
  --domain-certificate-arn arn:aws:acm:us-east-2:123456789012:certificate/  

eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE \
  --application-domain example.com \
  --endpoint-domain-prefix my-ava-app \
  --security-group-ids sg-004915970c4c8f13a \
  --network-interface-  

options NetworkInterfaceId=eni-0aec70418c8d87a0f,Protocol=https,Port=443 \

```

```
--tag-specifications ResourceType=verified-access-  
endpoint,Tags=[{Key=Name,Value=my-va-endpoint}]
```

Ausgabe:

```
{  
  "VerifiedAccessEndpoint": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",  
    "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",  
    "ApplicationDomain": "example.com",  
    "EndpointType": "network-interface",  
    "AttachmentType": "vpc",  
    "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/  
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",  
    "EndpointDomain": "my-ava-  
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-  
east-2.amazonaws.com",  
    "SecurityGroupIds": [  
      "sg-004915970c4c8f13a"  
    ],  
    "NetworkInterfaceOptions": {  
      "NetworkInterfaceId": "eni-0aec70418c8d87a0f",  
      "Protocol": "https",  
      "Port": 443  
    },  
    "Status": {  
      "Code": "pending"  
    },  
    "Description": "",  
    "CreationTime": "2023-08-25T20:54:43",  
    "LastUpdatedTime": "2023-08-25T20:54:43",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-endpoint"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Verified Access-Endpoints](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVerifiedAccessEndpoint](#) in der AWS CLI Befehlsreferenz.

create-verified-access-group

Das folgende Codebeispiel zeigt die Verwendung `create-verified-access-group`.

AWS CLI

Um eine Gruppe mit verifiziertem Zugriff zu erstellen

Im folgenden `create-verified-access-group` Beispiel wird eine Verified Access-Gruppe für die angegebene Verified Access-Instanz erstellt.

```
aws ec2 create-verified-access-group \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --tag-specifications ResourceType=verified-access-
group, Tags=[{Key=Name, Value=my-va-group}]
```

Ausgabe:

```
{
  "VerifiedAccessGroup": {
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "",
    "Owner": "123456789012",
    "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
    "CreationTime": "2023-08-25T19:55:19",
    "LastUpdatedTime": "2023-08-25T19:55:19",
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-va-group"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Verified Access-Gruppen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVerifiedAccessGroup](#) in der AWS CLI Befehlsreferenz.

create-verified-access-instance

Das folgende Codebeispiel zeigt die Verwendung `create-verified-access-instance`.

AWS CLI

Um eine Verified Access-Instanz zu erstellen

Im folgenden `create-verified-access-instance` Beispiel wird eine Verified Access-Instanz mit einem Name-Tag erstellt.

```
aws ec2 create-verified-access-instance \  
  --tag-specifications ResourceType=verified-access-  
instance,Tags=[{Key=Name,Value=my-va-instance}]
```

Ausgabe:

```
{  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "",  
    "VerifiedAccessTrustProviders": [],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-25T18:27:56",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-instance"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Verified Access-Instanzen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVerifiedAccessInstance](#) in der AWS CLI Befehlsreferenz.

create-verified-access-trust-provider

Das folgende Codebeispiel zeigt die Verwendung `create-verified-access-trust-provider`.

AWS CLI

So erstellen Sie einen Vertrauensanbieter mit verifiziertem Zugriff

Im folgenden `create-verified-access-trust-provider` Beispiel wird mithilfe von AWS Identity Center ein Verified Access-Vertrauensanbieter eingerichtet.

```
aws ec2 create-verified-access-trust-provider \  
  --trust-provider-type user \  
  --user-trust-provider-type iam-identity-center \  
  --policy-reference-name idc \  
  --tag-specifications ResourceType=verified-access-trust-  
provider,Tags=[{Key=Name,Value=my-va-trust-provider}]
```

Ausgabe:

```
{  
  "VerifiedAccessTrustProvider": {  
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
    "Description": "",  
    "TrustProviderType": "user",  
    "UserTrustProviderType": "iam-identity-center",  
    "PolicyReferenceName": "idc",  
    "CreationTime": "2023-08-25T18:40:36",  
    "LastUpdatedTime": "2023-08-25T18:40:36",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-trust-provider"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Vertrauensanbietern für verifizierten Zugriff](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVerifiedAccessTrustProvider](#) in der AWS CLI Befehlsreferenz.

create-volume

Das folgende Codebeispiel zeigt die Verwendung `create-volume`.

AWS CLI

Um ein leeres General Purpose SSD (GP2) -Volume zu erstellen

Im folgenden `create-volume` Beispiel wird ein 80-GiB-Volume für allgemeine Zwecke SSD (GP2) in der angegebenen Availability Zone erstellt. Beachten Sie, dass die aktuelle Region angegeben werden muss `us-east-1`, oder Sie können den `--region` Parameter hinzufügen, um die Region für den Befehl anzugeben.

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --availability-zone us-east-1a
```

Ausgabe:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": false,  
  "VolumeType": "gp2",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
  "Iops": 240,  
  "SnapshotId": "",  
  "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "Size": 80  
}
```

Wenn Sie keinen Volumetyp angeben, ist der Standard-Volumetyp `gp2`.

```
aws ec2 create-volume \  
  --size 80 \  
  --availability-zone us-east-1a
```

Beispiel 2: So erstellen Sie ein bereitgestelltes Volume IOPS SSD (io1) aus einem Snapshot

Im folgenden `create-volume` Beispiel wird mithilfe des angegebenen Snapshots ein bereitgestelltes Volume IOPS SSD (io1) erstellt, auf dem 1000 IOPS in der angegebenen Availability Zone bereitgestellt werden.

```
aws ec2 create-volume \  
  --volume-type io1 \  
  --iops 1000 \  
  --snapshot-id snap-066877671789bd71b \  
  --availability-zone us-east-1a
```

Ausgabe:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": false,  
  "VolumeType": "io1",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
  "Iops": 1000,  
  "SnapshotId": "snap-066877671789bd71b",  
  "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "Size": 500  
}
```

Beispiel 3: So erstellen Sie ein verschlüsseltes Volume

Im folgenden `create-volume` Beispiel wird ein verschlüsseltes Volume unter Verwendung der CMK EBS Standardverschlüsselung erstellt. Wenn die Verschlüsselung standardmäßig deaktiviert ist, müssen Sie den `--encrypted` Parameter wie folgt angeben.

```
aws ec2 create-volume \  
  --size 80 \  
  --encrypted \  
  --availability-zone us-east-1a
```

Ausgabe:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": true,  
  "VolumeType": "gp2",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
}
```

```

    "Iops": 240,
    "SnapshotId": "",
    "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "Size": 80
  }

```

Wenn die Standardverschlüsselung aktiviert ist, erstellt der folgende Beispielbefehl auch ohne den `--encrypted` Parameter ein verschlüsseltes Volume.

```

aws ec2 create-volume \
  --size 80 \
  --availability-zone us-east-1a

```

Wenn Sie den `--kms-key-id` Parameter verwenden, um ein vom Kunden verwaltetes Objekt anzugeben CMK, müssen Sie den `--encrypted` Parameter auch dann angeben, wenn die Verschlüsselung standardmäßig aktiviert ist.

```

aws ec2 create-volume \
  --volume-type gp2 \
  --size 80 \
  --encrypted \
  --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
  --availability-zone us-east-1a

```

Beispiel 4: Um ein Volume mit Tags zu erstellen

Das folgende `create-volume` Beispiel erstellt ein Volume und fügt zwei Tags hinzu.

```

aws ec2 create-volume \
  --availability-zone us-east-1a \
  --volume-type gp2 \
  --size 80 \
  --tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},
{Key=cost-center,Value=cc123}]'

```

- API-Einheiten finden Sie [CreateVolume](#) unter AWS CLI Befehlsreferenz.

create-vpc-endpoint-connection-notification

Das folgende Codebeispiel zeigt die Verwendung `create-vpc-endpoint-connection-notification`.

AWS CLI

Um eine Endpunktverbindungsbenachrichtigung zu erstellen

In diesem Beispiel wird eine Benachrichtigung für einen bestimmten Endpunktdienst erstellt, die Sie benachrichtigt, wenn Schnittstellenendpunkte eine Verbindung zu Ihrem Dienst hergestellt haben und wenn Endpunkte für Ihren Dienst akzeptiert wurden.

Befehl:

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-arn arn:aws:sns:us-east-2:123456789012:VpceNotification --connection-events Connect Accept --service-id vpce-svc-1237881c0d25a3abc
```

Ausgabe:

```
{
  "ConnectionNotification": {
    "ConnectionNotificationState": "Enabled",
    "ConnectionNotificationType": "Topic",
    "ServiceId": "vpce-svc-1237881c0d25a3abc",
    "ConnectionEvents": [
      "Accept",
      "Connect"
    ],
    "ConnectionNotificationId": "vpce-nfn-008776de7e03f5abc",
    "ConnectionNotificationArn": "arn:aws:sns:us-east-2:123456789012:VpceNotification"
  }
}
```

- API-Einheiten finden Sie [CreateVpcEndpointConnectionNotification](#) in der AWS CLI Befehlsreferenz.

create-vpc-endpoint-service-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-vpc-endpoint-service-configuration`.

AWS CLI

Beispiel 1: Um eine Endpunkt-Servicekonfiguration für einen Schnittstellenendpunkt zu erstellen

Im folgenden `create-vpc-endpoint-service-configuration` Beispiel wird mithilfe des Network Load Balancer `nlb-vpce` eine VPC Endpunktdienstkonfiguration erstellt. In diesem Beispiel wird auch angegeben, dass Anfragen zur Verbindung mit dem Dienst über einen Schnittstellenendpunkt akzeptiert werden müssen.

```
aws ec2 create-vpc-endpoint-service-configuration \  
  --network-load-balancer-arns arn:aws:elasticloadbalancing:us-  
east-1:123456789012:loadbalancer/net/nlb-vpce/e94221227f1ba532 \  
  --acceptance-required
```

Ausgabe:

```
{  
  "ServiceConfiguration": {  
    "ServiceType": [  
      {  
        "ServiceType": "Interface"  
      }  
    ],  
    "NetworkLoadBalancerArns": [  
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/  
nlb-vpce/e94221227f1ba532"  
    ],  
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-03d5ebb7d9579a2b3",  
    "ServiceState": "Available",  
    "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",  
    "AcceptanceRequired": true,  
    "AvailabilityZones": [  
      "us-east-1d"  
    ],  
    "BaseEndpointDnsNames": [  
      "vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com"  
    ]  
  }  
}
```

Beispiel 2: So erstellen Sie eine Endpunktdienstkonfiguration für einen Gateway Load Balancer-Endpunkt

Im folgenden `create-vpc-endpoint-service-configuration` Beispiel wird mithilfe des Gateway Load Balancer `GWLBService` eine VPC Endpunktdienstkonfiguration erstellt. Anfragen

zur Verbindung mit dem Dienst über einen Gateway Load Balancer-Endpunkt werden automatisch akzeptiert.

```
aws ec2 create-vpc-endpoint-service-configuration \  
  --gateway-load-balancer-arns arn:aws:elasticloadbalancing:us-  
east-1:123456789012:loadbalancer/gwy/GWLBSservice/123123123123abcc \  
  --no-acceptance-required
```

Ausgabe:

```
{  
  "ServiceConfiguration": {  
    "ServiceType": [  
      {  
        "ServiceType": "GatewayLoadBalancer"  
      }  
    ],  
    "ServiceId": "vpce-svc-123123a1c43abc123",  
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",  
    "ServiceState": "Available",  
    "AvailabilityZones": [  
      "us-east-1d"  
    ],  
    "AcceptanceRequired": false,  
    "ManagesVpcEndpoints": false,  
    "GatewayLoadBalancerArns": [  
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/gwy/  
GWLBSservice/123123123123abcc"  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [VPC Endpoint Services](#) im VPC Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateVpcEndpointServiceConfiguration](#) in der AWS CLI Befehlsreferenz.

create-vpc-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-vpc-endpoint`.

AWS CLI

Beispiel 1: Um einen Gateway-Endpunkt zu erstellen

Das folgende `create-vpc-endpoint` Beispiel erstellt einen VPC Gateway-Endpunkt zwischen VPC `vpc-1a2b3c4d` und Amazon S3 in der `us-east-1` Region und verknüpft die Routentabelle `rtb-11aa22bb` mit dem Endpunkt.

```
aws ec2 create-vpc-endpoint \  
  --vpc-id vpc-1a2b3c4d \  
  --service-name com.amazonaws.us-east-1.s3 \  
  --route-table-ids rtb-11aa22bb
```

Ausgabe:

```
{  
  "VpcEndpoint": {  
    "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",  
    \"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}\",  
    "VpcId": "vpc-1a2b3c4d",  
    "State": "available",  
    "ServiceName": "com.amazonaws.us-east-1.s3",  
    "RouteTableIds": [  
      "rtb-11aa22bb"  
    ],  
    "VpcEndpointId": "vpc-1a2b3c4d",  
    "CreationTimestamp": "2015-05-15T09:40:50Z"  
  }  
}
```

Weitere Informationen finden Sie im AWS PrivateLink Handbuch unter [Erstellen eines Gateway-Endpunkts](#).

Beispiel 2: So erstellen Sie einen Schnittstellen-Endpunkt

Das folgende `create-vpc-endpoint` Beispiel erstellt einen VPC Schnittstellenendpunkt zwischen VPC `vpc-1a2b3c4d` und Amazon S3 in der `us-east-1` Region. Der Befehl erstellt den Endpunkt im Subnetz `subnet-1a2b3c4d`, ordnet ihn einer Sicherheitsgruppe `sg-1a2b3c4d` zu und fügt ein Tag mit dem Schlüssel „Service“ und dem Wert „S3“ hinzu.

```
aws ec2 create-vpc-endpoint \  
  --vpc-id vpc-1a2b3c4d \  
  --service-name com.amazonaws.us-east-1.s3 \  
  --route-table-ids rtb-11aa22bb
```



```
--vpc-id vpc-1a2b3c4d \  
--vpc-endpoint-type Interface \  
--service-name com.amazonaws.us-east-1.s3 \  
--subnet-ids subnet-7b16de0c \  
--security-group-id sg-1a2b3c4d \  
--tag-specifications ResourceType=vpc-endpoint,Tags=[{Key=service,Value=S3}]
```

Ausgabe:

```
{  
  "VpcEndpoint": {  
    "VpcEndpointId": "vpce-1a2b3c4d5e6f1a2b3",  
    "VpcEndpointType": "Interface",  
    "VpcId": "vpc-1a2b3c4d",  
    "ServiceName": "com.amazonaws.us-east-1.s3",  
    "State": "pending",  
    "RouteTableIds": [],  
    "SubnetIds": [  
      "subnet-1a2b3c4d"  
    ],  
    "Groups": [  
      {  
        "GroupId": "sg-1a2b3c4d",  
        "GroupName": "default"  
      }  
    ],  
    "PrivateDnsEnabled": false,  
    "RequesterManaged": false,  
    "NetworkInterfaceIds": [  
      "eni-0b16f0581c8ac6877"  
    ],  
    "DnsEntries": [  
      {  
        "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg.s3.us-east-1.vpce.amazonaws.com",  
        "HostedZoneId": "Z7HUB22UULQXV"  
      },  
      {  
        "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg-us-east-1c.s3.us-east-1.vpce.amazonaws.com",  
        "HostedZoneId": "Z7HUB22UULQXV"  
      }  
    ],  
  },  
}
```

```

    "CreationTimestamp": "2021-03-05T14:46:16.030000+00:00",
    "Tags": [
      {
        "Key": "service",
        "Value": "S3"
      }
    ],
    "OwnerId": "123456789012"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im Benutzerhandbuch für AWS PrivateLink

Beispiel 3: So erstellen Sie einen Gateway Load Balancer Balancer-Endpunkt

Im folgenden `create-vpc-endpoint` Beispiel wird ein Gateway Load Balancer-Endpunkt zwischen VPC `vpc-111122223333aabbcc` und einem Dienst erstellt, der mit einem Gateway Load Balancer konfiguriert ist.

```

aws ec2 create-vpc-endpoint \
  --service-name com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123 \
  --vpc-endpoint-type GatewayLoadBalancer \
  --vpc-id vpc-111122223333aabbcc \
  --subnet-ids subnet-0011aabbcc2233445

```

Ausgabe:

```

{
  "VpcEndpoint": {
    "VpcEndpointId": "vpce-aabbaabbaabbaabba",
    "VpcEndpointType": "GatewayLoadBalancer",
    "VpcId": "vpc-111122223333aabbcc",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
    "State": "pending",
    "SubnetIds": [
      "subnet-0011aabbcc2233445"
    ],
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
      "eni-01010120203030405"
    ],
  },
}

```

```
    "CreationTimestamp": "2020-11-11T08:06:03.522Z",  
    "OwnerId": "123456789012"  
  }  
}
```

Weitere Informationen finden Sie unter [Gateway Load Balancer-Endpoints](#) im Benutzerhandbuch für AWS PrivateLink

- API-Einheiten finden Sie [CreateVpcEndpoint](#) in der AWS CLI Befehlsreferenz.

create-vpc-peering-connection

Das folgende Codebeispiel zeigt die Verwendung `create-vpc-peering-connection`.

AWS CLI

Um eine VPC Peering-Verbindung zwischen Ihrem herzustellen VPCs

In diesem Beispiel wird eine Peering-Verbindung zwischen Ihrem VPCs `vpc-1a2b3c4d` und `vpc-11122233` angefordert.

Befehl:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-  
id vpc-11122233
```

Ausgabe:

```
{  
  "VpcPeeringConnection": {  
    "Status": {  
      "Message": "Initiating Request to 444455556666",  
      "Code": "initiating-request"  
    },  
    "Tags": [],  
    "RequesterVpcInfo": {  
      "OwnerId": "444455556666",  
      "VpcId": "vpc-1a2b3c4d",  
      "CidrBlock": "10.0.0.0/28"  
    },  
    "VpcPeeringConnectionId": "pcx-111aaa111",  
  }  
}
```

```
"ExpirationTime": "2014-04-02T16:13:36.000Z",
"AccepterVpcInfo": {
  "OwnerId": "444455556666",
  "VpcId": "vpc-11122233"
}
}
```

Um VPC eine VPC Peering-Verbindung mit einem in einem anderen Konto herzustellen

In diesem Beispiel wird eine Peering-Verbindung zwischen Ihrem VPC (vpc-1a2b3c4d) und einem (vpc-11122233) angefordert, das zu Ihrem Konto 123456789012 gehört VPC. AWS

Befehl:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012
```

Um VPC eine VPC Peering-Verbindung mit einem in einer anderen Region herzustellen

In diesem Beispiel wird eine Peering-Verbindung zwischen Ihnen VPC in der aktuellen Region (vpc-1a2b3c4d) und einem VPC (vpc-11122233) in Ihrem Konto in der Region angefordert. us-west-2

Befehl:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-region us-west-2
```

In diesem Beispiel wird eine Peering-Verbindung zwischen Ihrem Konto VPC in der aktuellen Region (vpc-1a2b3c4d) und einem (vpc-11122233) angefordert, das zu Ihrem Konto 123456789012 gehört, das sich in der Region befindet. VPC AWS us-west-2

Befehl:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2
```

- [CreateVpcPeeringConnection AWS CLI](#) Einzelheiten finden Sie unter Befehlsreferenz. API

create-vpc

Das folgende Codebeispiel zeigt die Verwendung `create-vpc`.

AWS CLI

Beispiel 1: Um ein zu erstellen VPC

Im folgenden `create-vpc` Beispiel wird ein VPC mit dem angegebenen IPv4 CIDR Block und einem Namen-Tag erstellt.

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

Ausgabe:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-5EXAMPLE",  
    "State": "pending",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-07501b79ecEXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false,  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "MyVpc"  
      }  
    ]  
  }  
}
```

```
}
```

Beispiel 2: So erstellen Sie eine VPC mit einem dedizierten Mandantenverhältnis

Im folgenden `create-vpc` Beispiel wird ein Mandantenverhältnis VPC mit dem angegebenen IPv4 CIDR Block und einem dedizierten Tenancy erstellt.

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --instance-tenancy dedicated
```

Ausgabe:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-19edf471",  
    "State": "pending",  
    "VpcId": "vpc-0a53287fa4EXAMPLE",  
    "OwnerId": "111122223333",  
    "InstanceTenancy": "dedicated",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-00b24cc1c2EXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false  
  }  
}
```

Beispiel 3: Um eine VPC mit einem IPv6 CIDR Block zu erstellen

Im folgenden `create-vpc` Beispiel wird ein Block VPC mit einem von Amazon bereitgestellten Block IPv6 CIDR erstellt.

```
aws ec2 create-vpc \  
  --cidr-block 2001:db8:1:1::/64
```

```
--cidr-block 10.0.0.0/16 \
--amazon-provided-ipv6-cidr-block
```

Ausgabe:

```
{
  "Vpc": {
    "CidrBlock": "10.0.0.0/16",
    "DhcpOptionsId": "dopt-dEXAMPLE",
    "State": "pending",
    "VpcId": "vpc-0fc5e3406bEXAMPLE",
    "OwnerId": "123456789012",
    "InstanceTenancy": "default",
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",
        "Ipv6CidrBlock": "",
        "Ipv6CidrBlockState": {
          "State": "associating"
        },
        "Ipv6Pool": "Amazon",
        "NetworkBorderGroup": "us-west-2"
      }
    ],
    "CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ],
    "IsDefault": false
  }
}
```

Beispiel 4: Um einen VPC mit einem CIDR aus einem Pool zu erstellen IPAM

Im folgenden `create-vpc` Beispiel wird ein Pool VPC mit einem CIDR aus einem Amazon VPC IP Address Manager (IPAM) -Pool erstellt.

Linux und macOS:

```
aws ec2 create-vpc \  
  --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 \  
  --tag-specifications ResourceType=vpc,Tags='[{Key=Environment,Value="Preprod"}],  
{Key=Owner,Value="Build Team"}]'
```

Windows:

```
aws ec2 create-vpc ^  
  --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --tag-specifications ResourceType=vpc,Tags=[{Key=Environment,Value="Preprod"},  
{Key=Owner,Value="Build Team"}]
```

Ausgabe:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.1.0/24",  
    "DhcpOptionsId": "dopt-2afccf50",  
    "State": "pending",  
    "VpcId": "vpc-010e1791024eb0af9",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-0a77de1d803226d4b",  
        "CidrBlock": "10.0.1.0/24",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false,  
    "Tags": [  
      {  
        "Key": "Environment",  
        "Value": "Preprod"  
      },  
      {  
        "Key": "Owner",  
        "Value": "Build Team"  
      }  
    ]  
  }  
}
```



```

    ]
  }
}

```

Weitere Informationen finden Sie unter [Erstellen eines VPC, das einen IPAM Pool verwendet](#) CIDR im VPC IPAM Amazon-Benutzerhandbuch.

- API Einzelheiten finden Sie [CreateVpc](#) unter AWS CLI Befehlsreferenz.

create-vpn-connection-route

Das folgende Codebeispiel zeigt die Verwendung `create-vpn-connection-route`.

AWS CLI

Um eine statische Route für eine VPN Verbindung zu erstellen

In diesem Beispiel wird eine statische Route für die angegebene VPN Verbindung erstellt. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- API Einzelheiten finden Sie [CreateVpnConnectionRoute](#) in der AWS CLI Befehlsreferenz.

create-vpn-connection

Das folgende Codebeispiel zeigt die Verwendung `create-vpn-connection`.

AWS CLI

Beispiel 1: Um eine VPN Verbindung mit dynamischem Routing herzustellen

Im folgenden `create-vpn-connection` Beispiel wird eine VPN Verbindung zwischen dem angegebenen Virtual Private Gateway und dem angegebenen Kunden-Gateway hergestellt und Tags auf die VPN Verbindung angewendet. Die Ausgabe enthält die Konfigurationsinformationen für Ihr Kunden-Gateway-Gerät im XML Format.

```
aws ec2 create-vpn-connection \
  --type ipsec.1 \
```

```
--customer-gateway-id cgw-001122334455aabbc \  
--vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \  
--tag-specification 'ResourceType=vpn-connection,Tags=[{Key=Name,Value=BGP-VPN}]'
```

Ausgabe:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabbc",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abcab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": false,  
      "LocalIpv4NetworkCidr": "0.0.0.0/0",  
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
      "TunnelInsideIpVersion": "ipv4",  
      "TunnelOptions": [  
        {},  
        {}  
      ]  
    },  
    "Routes": [],  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "BGP-VPN"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [So AWS Site-to-Site VPN funktioniert](#) es.

Beispiel 2: Um eine VPN Verbindung mit statischem Routing herzustellen

Im folgenden `create-vpn-connection` Beispiel wird eine VPN Verbindung zwischen dem angegebenen Virtual Private Gateway und dem angegebenen Kunden-Gateway hergestellt. Die

Optionen spezifizieren statisches Routing. Die Ausgabe enthält die Konfigurationsinformationen für Ihr Kunden-Gateway-Gerät im XML Format.

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --customer-gateway-id cgw-001122334455aabbc \  
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \  
  --options "{\"StaticRoutesOnly\":true}"
```

Ausgabe:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "..configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabbc",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abcab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": true,  
      "LocalIpv4NetworkCidr": "0.0.0.0/0",  
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
      "TunnelInsideIpVersion": "ipv4",  
      "TunnelOptions": [  
        {},  
        {}  
      ]  
    },  
    "Routes": [],  
    "Tags": []  
  }  
}
```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [So AWS Site-to-Site VPN funktioniert](#) es.

Beispiel 3: So stellen Sie eine VPN Verbindung her und geben Ihren eigenen internen CIDR und vorab gemeinsam genutzten Schlüssel an

Im folgenden `create-vpn-connection` Beispiel wird eine VPN Verbindung hergestellt und der interne CIDR IP-Adressblock sowie ein benutzerdefinierter vorinstallierter Schlüssel für jeden Tunnel angegeben. Die angegebenen Werte werden in den `CustomerGatewayConfiguration` Informationen zurückgegeben.

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --customer-gateway-id cgw-001122334455aabb \  
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b \  
  --options  
  TunnelOptions='[{TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},  
{TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'
```

Ausgabe:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "..configuration information..",  
    "CustomerGatewayId": "cgw-001122334455aabb",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abcab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": false,  
      "LocalIpv4NetworkCidr": "0.0.0.0/0",  
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
      "TunnelInsideIpVersion": "ipv4",  
      "TunnelOptions": [  
        {  
          "OutsideIpAddress": "203.0.113.3",  
          "TunnelInsideCidr": "169.254.12.0/30",  
          "PreSharedKey": "ExamplePreSharedKey1"  
        },  
        {  
          "OutsideIpAddress": "203.0.113.5",  
          "TunnelInsideCidr": "169.254.13.0/30",  
          "PreSharedKey": "ExamplePreSharedKey2"  
        }  
      ]  
    }  
  },  
}
```

```

    "Routes": [],
    "Tags": []
  }
}

```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [So AWS Site-to-Site VPN funktioniert](#) es.

Beispiel 4: So erstellen Sie eine VPN Verbindung, die IPv6 Traffic unterstützt

Im folgenden `create-vpn-connection` Beispiel wird eine VPN Verbindung erstellt, die den IPv6 Verkehr zwischen dem angegebenen Transit-Gateway und dem angegebenen Kunden-Gateway unterstützt. Die Tunneloptionen für beide Tunnel geben an, dass die IKE Verhandlung eingeleitet AWS werden muss.

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --transit-gateway-id tgw-12312312312312312 \
  --customer-gateway-id cgw-001122334455aabbc \
  --options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},
{StartupAction=start}]

```

Ausgabe:

```

{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbc",
    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-1111111112222222",
    "TransitGatewayId": "tgw-12312312312312312",
    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": false,
      "LocalIpv6NetworkCidr": "::/0",
      "RemoteIpv6NetworkCidr": "::/0",
      "TunnelInsideIpVersion": "ipv6",
      "TunnelOptions": [
        {
          "OutsideIpAddress": "203.0.113.3",
          "StartupAction": "start"
        }
      ]
    }
  }
}

```

```
        },
        {
            "OutsideIpAddress": "203.0.113.5",
            "StartupAction": "start"
        }
    ]
},
"Routes": [],
"Tags": []
}
}
```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [So AWS Site-to-Site VPN funktioniert](#) es.

- API Einzelheiten finden Sie [CreateVpnConnection](#) in der AWS CLI Befehlsreferenz.

create-vpn-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-vpn-gateway`.

AWS CLI

Um ein virtuelles privates Gateway zu erstellen

In diesem Beispiel wird ein virtuelles privates Gateway erstellt.

Befehl:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

Ausgabe:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 64512,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

Um ein virtuelles privates Gateway mit einer bestimmten Amazon-Seite zu erstellen ASN

In diesem Beispiel wird ein virtuelles privates Gateway erstellt und die Autonomous System Number (ASN) für die Amazon-Seite der BGP Sitzung angegeben.

Befehl:

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

Ausgabe:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 65001,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

- API-Einheiten finden Sie [CreateVpnGateway](#) in der AWS CLI Befehlsreferenz.

delete-carrier-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-carrier-gateway`.

AWS CLI

Um Ihr Carrier-Gateway zu löschen

Im folgenden `delete-carrier-gateway` Beispiel wird das angegebene Carrier-Gateway gelöscht.

```
aws ec2 delete-carrier-gateway \
  --carrier-gateway-id cagw-0465cdEXAMPLE1111
```

Ausgabe:

```
{
  "CarrierGateway": {
```

```
"CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
  "VpcId": "vpc-0c529aEXAMPLE1111",
  "State": "deleting",
  "OwnerId": "123456789012"
}
}
```

Weitere Informationen finden Sie unter [Carrier Gateways](#) im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteCarrierGateway AWS CLI Befehlsreferenz](#).

delete-client-vpn-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-client-vpn-endpoint`.

AWS CLI

Um einen VPN Client-Endpunkt zu löschen

Im folgenden `delete-client-vpn-endpoint` Beispiel wird der angegebene VPN Client-Endpunkt gelöscht.

```
aws ec2 delete-client-vpn-endpoint \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Ausgabe:

```
{
  "Status": {
    "Code": "deleting"
  }
}
```

Weitere Informationen finden Sie unter [VPNClient-Endpunkte](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteClientVpnEndpoint](#) in der AWS CLI Befehlsreferenz.

delete-client-vpn-route

Das folgende Codebeispiel zeigt die Verwendung `delete-client-vpn-route`.

AWS CLI

Um eine Route für einen VPN Client-Endpunkt zu löschen

Im folgenden `delete-client-vpn-route` Beispiel wird die `0.0.0.0/0` Route für das angegebene Subnetz eines VPN Client-Endpunkts gelöscht.

```
aws ec2 delete-client-vpn-route \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --destination-cidr-block 0.0.0.0/0 \  
  --target-vpc-subnet-id subnet-0123456789abcabca
```

Ausgabe:

```
{  
  "Status": {  
    "Code": "deleting"  
  }  
}
```

Weitere Informationen finden Sie im AWS VPNClient-Administratorhandbuch unter [Routes](#).

- API-Einheiten finden Sie [DeleteClientVpnRoute](#) in der AWS CLI Befehlsreferenz.

`delete-coip-cidr`

Das folgende Codebeispiel zeigt die Verwendung `delete-coip-cidr`.

AWS CLI

Um einen Bereich von kundeneigenen IP-Adressen (CoIP) zu löschen

Im folgenden `delete-coip-cidr` Beispiel wird der angegebene CoIP-Adressbereich im angegebenen CoIP-Pool gelöscht.

```
aws ec2 delete-coip-cidr \  
  --cidr 14.0.0.0/24 \  
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Ausgabe:

```
{
  "CoipCidr": {
    "Cidr": "14.0.0.0/24",
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
  }
}
```

Weitere Informationen finden Sie unter [Kundeneigene IP-Adressen](#) im AWS -Outposts-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter Befehlsreferenz [DeleteCoipCidr](#).AWS CLI

delete-coip-pool

Das folgende Codebeispiel zeigt die Verwendung `delete-coip-pool`.

AWS CLI

Um einen Pool von kundeneigenen IP-Adressen (CoIP) zu löschen

Im folgenden `delete-coip-pool` Beispiel wird ein CoIP-Pool mit CoIP-Adressen gelöscht.

```
aws ec2 delete-coip-pool \
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Ausgabe:

```
{
  "CoipPool": {
    "PoolId": "ipv4pool-coip-1234567890abcdefg",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-coip-1234567890abcdefg"
  }
}
```

Weitere Informationen finden Sie unter [Kundeneigene IP-Adressen](#) im AWS -Outposts-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter Befehlsreferenz [DeleteCoipPool](#).AWS CLI

delete-customer-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-customer-gateway`.

AWS CLI

Um ein Kunden-Gateway zu löschen

In diesem Beispiel wird das angegebene Kunden-Gateway gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- API-Einheiten finden Sie [DeleteCustomerGateway](#) in der AWS CLI Befehlsreferenz.

delete-dhcp-options

Das folgende Codebeispiel zeigt die Verwendung `delete-dhcp-options`.

AWS CLI

Um einen DHCP Optionssatz zu löschen

In diesem Beispiel wird der angegebene DHCP Optionssatz gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- API-Einheiten finden Sie unter [DeleteDhcpOptions AWS CLI](#) Befehlsreferenz.

delete-egress-only-internet-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-egress-only-internet-gateway`.

AWS CLI

Um ein Internet-Gateway nur für ausgehenden Datenverkehr zu löschen

In diesem Beispiel wird das angegebene Internet-Gateway gelöscht, das nur für ausgehenden Datenverkehr bestimmt ist.

Befehl:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

Ausgabe:

```
{
  "ReturnCode": true
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [DeleteEgressOnlyInternetGateway](#).AWS CLI

delete-fleets

Das folgende Codebeispiel zeigt die Verwendung `delete-fleets`.

AWS CLI

Beispiel 1: Um eine EC2 Flotte zu löschen und die zugehörigen Instances zu beenden

Das folgende `delete-fleets` Beispiel löscht die angegebene EC2 Flotte und beendet die zugehörigen On-Demand-Instances und Spot-Instances.

```
aws ec2 delete-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
  --terminate-instances
```

Ausgabe:

```
{
  "SuccessfulFleetDeletions": [
    {
      "CurrentFleetState": "deleted_terminating",
      "PreviousFleetState": "active",
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
    }
  ],
}
```

```
"UnsuccessfulFleetDeletions": []
}
```

Weitere Informationen finden Sie unter [Löschen einer EC2 Flotte](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: Um eine EC2 Flotte zu löschen, ohne die zugehörigen Instances zu beenden

Im folgenden `delete-fleets` Beispiel wird die angegebene EC2 Flotte gelöscht, ohne die zugehörigen On-Demand-Instances und Spot-Instances zu beenden.

```
aws ec2 delete-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
  --no-terminate-instances
```

Ausgabe:

```
{
  "SuccessfulFleetDeletions": [
    {
      "CurrentFleetState": "deleted_running",
      "PreviousFleetState": "active",
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
    }
  ],
  "UnsuccessfulFleetDeletions": []
}
```

Weitere Informationen finden Sie unter [Löschen einer EC2 Flotte](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DeleteFleets](#) unter AWS CLI Befehlsreferenz.

delete-flow-logs

Das folgende Codebeispiel zeigt die Verwendung `delete-flow-logs`.

AWS CLI

Um ein Flow-Protokoll zu löschen

Im folgenden `delete-flow-logs` Beispiel wird das angegebene Flow-Protokoll gelöscht.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

Ausgabe:

```
{  
  "Unsuccessful": []  
}
```

- API-Einheiten finden Sie [DeleteFlowLogs](#) in der AWS CLI Befehlsreferenz.

delete-fpga-image

Das folgende Codebeispiel zeigt die Verwendung `delete-fpga-image`.

AWS CLI

Um ein FPGA Amazon-Bild zu löschen

In diesem Beispiel wird das angegebene AFI gelöscht.

Befehl:

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

Ausgabe:

```
{  
  "Return": true  
}
```

- API-Einheiten finden Sie [DeleteFpgaImage](#) in der AWS CLI Befehlsreferenz.

delete-instance-connect-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-instance-connect-endpoint`.

AWS CLI

So löschen Sie einen EC2 Instance Connect-Endpoint

Im folgenden `delete-instance-connect-endpoint` Beispiel wird der angegebene EC2 Instance Connect-Endpunkt gelöscht.

```
aws ec2 delete-instance-connect-endpoint \  
  --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

Ausgabe:

```
{  
  "InstanceConnectEndpoint": {  
    "OwnerId": "111111111111",  
    "InstanceConnectEndpointId": "eice-0123456789example",  
    "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-  
connect-endpoint/eice-0123456789example",  
    "State": "delete-in-progress",  
    "StateMessage": "",  
    "NetworkInterfaceIds": [],  
    "VpcId": "vpc-0123abcd",  
    "AvailabilityZone": "us-east-1d",  
    "CreatedAt": "2023-02-07T12:05:37+00:00",  
    "SubnetId": "subnet-0123abcd"  
  }  
}
```

Weitere Informationen finden [Sie unter Entfernen des EC2 Instance Connect-Endpunkts](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteInstanceConnectEndpoint](#) in der AWS CLI Befehlsreferenz.

delete-instance-event-window

Das folgende Codebeispiel zeigt die Verwendung `delete-instance-event-window`.

AWS CLI

Beispiel 1: Um ein Ereignisfenster zu löschen

Das folgende `delete-instance-event-window` Beispiel löscht ein Ereignisfenster.

```
aws ec2 delete-instance-event-window \  
  --region us-east-1 \  
  --event-window-id ew-0123456789example
```

```
--instance-event-window-id iew-0abcdef1234567890
```

Ausgabe:

```
{
  "InstanceEventWindowState": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "State": "deleting"
  }
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 2: So erzwingen Sie das Löschen eines Ereignisfensters

Das folgende `delete-instance-event-window` Beispiel erzwingt das Löschen eines Ereignisfensters, wenn das Ereignisfenster derzeit Zielen zugeordnet ist.

```
aws ec2 delete-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --force-delete
```

Ausgabe:

```
{
  "InstanceEventWindowState": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "State": "deleting"
  }
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

- API-Einheiten finden Sie [DeleteInstanceEventWindow](#) in der AWS CLI Befehlsreferenz.

delete-internet-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-internet-gateway`.

AWS CLI

Um ein Internet-Gateway zu löschen

Im folgenden `delete-internet-gateway` Beispiel wird das angegebene Internet-Gateway gelöscht.

```
aws ec2 delete-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Internet-Gateways](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteInternetGateway](#) in der AWS CLI Befehlsreferenz.

`delete-ipam-pool`

Das folgende Codebeispiel zeigt die Verwendung `delete-ipam-pool`.

AWS CLI

Um einen IPAM Pool zu löschen

In diesem Beispiel sind Sie ein IPAM delegierter Administrator, der einen IPAM Pool löschen möchte, den Sie nicht mehr benötigen, aber für den Pool wurde ein CIDR bereitgestellter Pool bereitgestellt. Sie können einen Pool, für den er CIDRs bereitgestellt wurde, nicht löschen, es sei denn, Sie verwenden die `--cascade` Option, also verwenden Sie. `--cascade`

Gehen Sie wie folgt vor, um diese Anfrage abzuschließen:

Sie benötigen die IPAM Pool-ID, die Sie erhalten können. `--region` Es muss sich um [describe-ipam-pools](#) die IPAM Heimatregion handeln.

Das folgende `delete-ipam-pool` Beispiel löscht einen IPAM Pool in Ihrem AWS Konto.

```
aws ec2 delete-ipam-pool \  
  --ipam-pool-id ipam-pool-050c886a3ca41cd5b \  
  --cascade \  
  --region us-east-1
```

Ausgabe:

```
{
  "IpamPool": {
    "OwnerId": "320805250157",
    "IpamPoolId": "ipam-pool-050c886a3ca41cd5b",
    "IpamPoolArn": "arn:aws:ec2::320805250157:ipam-pool/ipam-
pool-050c886a3ca41cd5b",
    "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0a158dde35c51107b",
    "IpamScopeType": "private",
    "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
    "IpamRegion": "us-east-1",
    "Locale": "None",
    "PoolDepth": 1,
    "State": "delete-in-progress",
    "Description": "example",
    "AutoImport": false,
    "AddressFamily": "ipv4",
    "AllocationMinNetmaskLength": 0,
    "AllocationMaxNetmaskLength": 32
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines Pools](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteIpamPool](#) in der AWS CLI Befehlsreferenz.

delete-ipam-resource-discovery

Das folgende Codebeispiel zeigt die Verwendung `delete-ipam-resource-discovery`.

AWS CLI

Um eine Ressourcenerkennung zu löschen

In diesem Beispiel sind Sie ein IPAM delegierter Administrator, der eine nicht standardmäßige Ressourcensuche löschen möchte, die Sie erstellt haben, um sie während der Integration mit Konten außerhalb Ihrer Organisation IPAM mit einem anderen IPAM Administrator zu teilen.

Um diese Anfrage abzuschließen:

Dies `--region` muss die Region sein, in der Sie die Ressourcenerkennung erstellt haben. Sie können eine standardmäßige Ressourcenerkennung nicht löschen, wenn. `"IsDefault": true`

Eine Standardressourcensuche wird automatisch in dem Konto erstellt, das eine erstellt. IPAM Um eine Standardressourcensuche zu löschen, müssen Sie die löschenIPAM.

Im folgenden `delete-ipam-resource-discovery` Beispiel wird eine Ressourcenerkennung gelöscht.

```
aws ec2 delete-ipam-resource-discovery \  
  --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \  
  --region us-east-1
```

Ausgabe:

```
{  
  "IpamResourceDiscovery": {  
    "OwnerId": "149977607591",  
    "IpamResourceDiscoveryId": "ipam-res-disco-0e39761475298ee0f",  
    "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-  
discovery/ipam-res-disco-0e39761475298ee0f",  
    "IpamResourceDiscoveryRegion": "us-east-1",  
    "OperatingRegions": [  
      {  
        "RegionName": "us-east-1"  
      }  
    ],  
    "IsDefault": false,  
    "State": "delete-in-progress"  
  }  
}
```

Weitere Informationen zu Ressourcenentdeckungen finden Sie unter [Arbeiten mit Ressourcenentdeckungen](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeletelIpamResourceDiscovery](#)unter AWS CLI Befehlsreferenz.

delete-ipam-scope

Das folgende Codebeispiel zeigt die Verwendung`delete-ipam-scope`.

AWS CLI

Um einen IPAM Bereich zu löschen

Im folgenden `delete-ipam-scope` Beispiel wird ein IPAM gelöscht.

```
aws ec2 delete-ipam-scope \  
  --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4
```

Ausgabe:

```
{  
  "IpamScope": {  
    "OwnerId": "123456789012",  
    "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",  
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-01c1ebab2b63bd7e4",  
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
    "IpamRegion": "us-east-1",  
    "IpamScopeType": "private",  
    "IsDefault": false,  
    "Description": "Example description",  
    "PoolCount": 0,  
    "State": "delete-in-progress"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen eines Bereichs](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteIpamScope](#) in der AWS CLI Befehlsreferenz.

delete-ipam

Das folgende Codebeispiel zeigt die Verwendung `delete-ipam`.

AWS CLI

Um ein zu löschen IPAM

Das folgende `delete-ipam` Beispiel löscht eine IPAM.

```
aws ec2 delete-ipam \  
  --ipam-id ipam-036486dfa6af58ee0
```

Ausgabe:

```
{
  "Ipam": {
    "OwnerId": "123456789012",
    "IpamId": "ipam-036486dfa6af58ee0",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
    "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
    "ScopeCount": 2,
    "OperatingRegions": [
      {
        "RegionName": "us-east-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-west-1"
      }
    ],
    "State": "delete-in-progress"
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines IPAM](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteIpam](#) in der AWS CLI Befehlsreferenz.

delete-key-pair

Das folgende Codebeispiel zeigt die Verwendung `delete-key-pair`.

AWS CLI

So löschen Sie ein Schlüsselpaar

Im folgenden `delete-key-pair` Beispiel wird das angegebene key pair gelöscht.

```
aws ec2 delete-key-pair \
  --key-name my-key-pair
```

Ausgabe:

```
{
  "Return": true,
  "KeyId": "key-03c8d3aceb53b507"
}
```

Weitere Informationen finden Sie unter [Erstellen und Löschen von Schlüsselpaaren](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DeleteKeyPair](#) unter AWS CLI Befehlsreferenz.

delete-launch-template-versions

Das folgende Codebeispiel zeigt die Verwendung `delete-launch-template-versions`.

AWS CLI

Um eine Version einer Startvorlage zu löschen

In diesem Beispiel wird die angegebene Version der Startvorlage gelöscht.

Befehl:

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --  
versions 1
```

Ausgabe:

```
{
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],
  "SuccessfullyDeletedLaunchTemplateVersions": [
    {
      "LaunchTemplateName": "TestVersion",
      "VersionNumber": 1,
      "LaunchTemplateId": "lt-0abcd290751193123"
    }
  ]
}
```

- API-Einheiten finden Sie [DeleteLaunchTemplateVersions](#) in der AWS CLI Befehlsreferenz.

delete-launch-template

Das folgende Codebeispiel zeigt die Verwendung `delete-launch-template`.

AWS CLI

So löschen Sie eine Startvorlage

In diesem Beispiel wird die angegebene Startvorlage gelöscht.

Befehl:

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

Ausgabe:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "TestTemplate",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- API-Einheiten finden Sie [DeleteLaunchTemplate](#) in der AWS CLI Befehlsreferenz.

delete-local-gateway-route-table-virtual-interface-group-association

Das folgende Codebeispiel zeigt die Verwendung `delete-local-gateway-route-table-virtual-interface-group-association`.

AWS CLI

Um die Zuordnung einer lokalen Gateway-Routentabelle zu einer Gruppe virtueller Schnittstellen (VIFs) zu trennen

Im folgenden `delete-local-gateway-route-table-virtual-interface-group-association` Beispiel wird die Zuordnung zwischen der angegebenen lokalen Gateway-Routentabelle und VIF der angegebenen Gruppe gelöscht.

```
aws ec2 delete-local-gateway-route-table-virtual-interface-group-association \  
  --local-gateway-route-table-virtual-interface-group-association-id lgw-vif-grp-  
  assoc-exampleid12345678
```

Ausgabe:

```
{  
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {  
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-  
    assoc-exampleid12345678",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",  
    "LocalGatewayId": "lgw-exampleid11223344",  
    "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",  
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-  
    gateway-route-table/lgw-rtb-exampleidabcd1234",  
    "OwnerId": "111122223333",  
    "State": "disassociating",  
    "Tags": []  
  }  
}
```

Weitere Informationen finden Sie unter [VIFGruppenzuordnungen](#) im AWS Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#) in der AWS CLI Befehlsreferenz.

delete-local-gateway-route-table-vpc-association

Das folgende Codebeispiel zeigt die Verwendung `delete-local-gateway-route-table-vpc-association`.

AWS CLI

Um die Zuordnung einer lokalen Gateway-Routentabelle von einer zu trennen VPC

Im folgenden `delete-local-gateway-route-table-vpc-association` Beispiel wird die Zuordnung zwischen der angegebenen lokalen Gateway-Routentabelle und gelöscht. VPC

```
aws ec2 delete-local-gateway-route-table-vpc-association \  
  --local-gateway-route-table-vpc-association-id vpc-example0123456789
```


Ausgabe:

```
{
  "LocalGatewayRouteTableVpcAssociation": {
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-abcd1234wxyz56789",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:555555555555:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
    "LocalGatewayId": "lgw-exampleid01234567",
    "VpcId": "vpc-example0123456789",
    "OwnerId": "555555555555",
    "State": "disassociating"
  }
}
```

Weitere Informationen finden Sie unter [VPCVerknüpfungen](#) im AWS Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteLocalGatewayRouteTableVpcAssociation](#) in der AWS CLI Befehlsreferenz.

delete-local-gateway-route-table

Das folgende Codebeispiel zeigt die Verwendung `delete-local-gateway-route-table`.

AWS CLI

Um eine lokale Gateway-Routentabelle zu löschen

Im folgenden `delete-local-gateway-route-table` Beispiel wird eine lokale Gateway-Routentabelle mit dem direkten VPC Routingmodus erstellt.

```
aws ec2 delete-local-gateway-route-table \
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Ausgabe:

```
{
  "LocalGatewayRouteTable": {
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-abcdefg1234567890",
    "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
  }
}
```

```

    "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-021345abcdef67890",
    "OwnerId": "111122223333",
    "State": "deleting",
    "Tags": [],
    "Mode": "direct-vpc-routing"
  }
}

```

Weitere Informationen finden Sie unter [Roouting-Tabellen für lokale Gateways](#) im AWS -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteLocalGatewayRoute](#) unter AWS CLI Befehlsreferenz.

delete-local-gateway-route

Das folgende Codebeispiel zeigt die Verwendung `delete-local-gateway-route`.

AWS CLI

Um eine Route aus einer lokalen Gateway-Routentabelle zu löschen

Im folgenden `delete-local-gateway-route` Beispiel wird die angegebene Route aus der angegebenen lokalen Gateway-Routentabelle gelöscht.

```

aws ec2 delete-local-gateway-route \
  --destination-cidr-block 0.0.0.0/0 \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE

```

Ausgabe:

```

{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
  }
}

```

- API-Einheiten finden Sie unter [DeleteLocalGatewayRoute AWS CLI](#) Befehlsreferenz.

delete-managed-prefix-list

Das folgende Codebeispiel zeigt die Verwendung `delete-managed-prefix-list`.

AWS CLI

Um eine Präfixliste zu löschen

Im folgenden `delete-managed-prefix-list` Beispiel wird die angegebene Präfixliste gelöscht.

```
aws ec2 delete-managed-prefix-list \
  --prefix-list-id pl-0123456abcabcabc1
```

Ausgabe:

```
{
  "PrefixList": {
    "PrefixListId": "pl-0123456abcabcabc1",
    "AddressFamily": "IPv4",
    "State": "delete-in-progress",
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
    "PrefixListName": "test",
    "MaxEntries": 10,
    "Version": 1,
    "OwnerId": "123456789012"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteManagedPrefixList](#) in der AWS CLI Befehlsreferenz.

delete-nat-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-nat-gateway`.

AWS CLI

Um ein NAT Gateway zu löschen

In diesem Beispiel wird ein NAT Gateway `nat-04ae55e711cec5680` gelöscht.

Befehl:

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

Ausgabe:

```
{  
  "NatGatewayId": "nat-04ae55e711cec5680"  
}
```

- API-Einheiten finden Sie [DeleteNatGateway](#) in der AWS CLI Befehlsreferenz.

delete-network-acl-entry

Das folgende Codebeispiel zeigt die Verwendung `delete-network-acl-entry`.

AWS CLI

Um einen ACL Netzwerkeintrag zu löschen

In diesem Beispiel wird die Eingangsregel Nummer 100 aus dem angegebenen Netzwerk gelöscht. ACL Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- API-Einheiten finden Sie unter [DeleteNetworkAcEntry AWS CLI](#) Befehlsreferenz.

delete-network-acl

Das folgende Codebeispiel zeigt die Verwendung `delete-network-acl`.

AWS CLI

Um ein Netzwerk zu löschen ACL

In diesem Beispiel wird das angegebene Netzwerk ACL gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- API-Einheiten finden Sie [DeleteNetworkAcl](#) in der AWS CLI Befehlsreferenz.

delete-network-insights-access-scope-analysis

Das folgende Codebeispiel zeigt die Verwendung `delete-network-insights-access-scope-analysis`.

AWS CLI

Um eine Network Access Scope-Analyse zu löschen

Im folgenden `delete-network-insights-access-scope-analysis` Beispiel wird die angegebene Analyse des Netzwerkzugriffsbereichs gelöscht.

```
aws ec2 delete-network-insights-access-scope-analysis \  
  --network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

Ausgabe:

```
{  
  "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer using the AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [DeleteNetworkInsightsAccessScopeAnalysis](#) unter AWS CLI Befehlsreferenz.

delete-network-insights-access-scope

Das folgende Codebeispiel zeigt die Verwendung `delete-network-insights-access-scope`.

AWS CLI

Um einen Netzwerkzugriffsbereich zu löschen

Im folgenden `delete-network-insights-access-scope` Beispiel wird der angegebene Netzwerkzugriffsbereich gelöscht.

```
aws ec2 delete-network-insights-access-scope \  
  --network-insights-access-scope-id nis-123456789abc01234
```

Ausgabe:

```
{  
  "NetworkInsightsAccessScopeId": "nis-123456789abc01234"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer unter Verwendung des AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [DeleteNetworkInsightsAccessScope](#) unter AWS CLI Befehlsreferenz.

`delete-network-insights-analysis`

Das folgende Codebeispiel zeigt die Verwendung `delete-network-insights-analysis`.

AWS CLI

Um eine Pfadanalyse zu löschen

Im folgenden `delete-network-insights-analysis` Beispiel wird die angegebene Analyse gelöscht.

```
aws ec2 delete-network-insights-analysis \  
  --network-insights-analysis-id nia-02207aa13eb480c7a
```

Ausgabe:

```
{  
  "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS CLI](#) im Reachability Analyzer Guide.

- API-Einheiten finden Sie [DeleteNetworkInsightsAnalysis](#) in der AWS CLI Befehlsreferenz.

delete-network-insights-path

Das folgende Codebeispiel zeigt die Verwendung `delete-network-insights-path`.

AWS CLI

Um einen Pfad zu löschen

Im folgenden `delete-network-insights-path` Beispiel wird der angegebene Pfad gelöscht. Bevor Sie einen Pfad löschen können, müssen Sie alle zugehörigen Analysen mithilfe des `delete-network-insights-analysis` Befehls löschen.

```
aws ec2 delete-network-insights-path \
  --network-insights-path-id nip-0b26f224f1d131fa8
```

Ausgabe:

```
{
  "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS CLI](#) im Reachability Analyzer Guide.

- API-Einheiten finden Sie [DeleteNetworkInsightsPath](#) in der AWS CLI Befehlsreferenz.

delete-network-interface-permission

Das folgende Codebeispiel zeigt die Verwendung `delete-network-interface-permission`.

AWS CLI

Um eine Netzwerkschnittstellenberechtigung zu löschen

In diesem Beispiel wird die angegebene Netzwerkschnittstellenberechtigung gelöscht.

Befehl:

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-perm-06fd19020ede149ea
```

Ausgabe:

```
{  
  "Return": true  
}
```

- API-Einheiten finden Sie [DeleteNetworkInterfacePermission](#) in der AWS CLI Befehlsreferenz.

delete-network-interface

Das folgende Codebeispiel zeigt die Verwendung `delete-network-interface`.

AWS CLI

Um eine Netzwerkschnittstelle zu löschen

In diesem Beispiel wird die angegebene Netzwerkschnittstelle gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- API-Einheiten finden Sie [DeleteNetworkInterface](#) in der AWS CLI Befehlsreferenz.

delete-placement-group

Das folgende Codebeispiel zeigt die Verwendung `delete-placement-group`.

AWS CLI

Um eine Platzierungsgruppe zu löschen

Dieser Beispielbefehl löscht die angegebene Platzierungsgruppe.

Befehl:

```
aws ec2 delete-placement-group --group-name my-cluster
```

- API-Einheiten finden Sie unter [DeletePlacementGroup AWS CLI Befehlsreferenz](#).

delete-queued-reserved-instances

Das folgende Codebeispiel zeigt die Verwendung `delete-queued-reserved-instances`.

AWS CLI

Um einen Kauf in der Warteschlange zu löschen

Im folgenden `delete-queued-reserved-instances` Beispiel wird die angegebene Reserved Instance gelöscht, die sich in der Warteschlange zum Kauf befand.

```
aws ec2 delete-queued-reserved-instances \  
  --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

Ausgabe:

```
{  
  "SuccessfulQueuedPurchaseDeletions": [  
    {  
      "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"  
    }  
  ],  
  "FailedQueuedPurchaseDeletions": []  
}
```

- API-Einheiten finden Sie unter [DeleteQueuedReservedInstances AWS CLI Befehlsreferenz](#).

delete-route-table

Das folgende Codebeispiel zeigt die Verwendung `delete-route-table`.

AWS CLI

Um eine Routentabelle zu löschen

In diesem Beispiel wird die angegebene Routentabelle gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- API-Einheiten finden Sie [DeleteRouteTable](#) in der AWS CLI Befehlsreferenz.

delete-route

Das folgende Codebeispiel zeigt die Verwendung `delete-route`.

AWS CLI

Um eine Route zu löschen

In diesem Beispiel wird die angegebene Route aus der angegebenen Routentabelle gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0
```

- API-Einheiten finden Sie unter [DeleteRoute AWS CLI](#) Befehlsreferenz.

delete-security-group

Das folgende Codebeispiel zeigt die Verwendung `delete-security-group`.

AWS CLI

[EC2-Classic] Um eine Sicherheitsgruppe zu löschen

In diesem Beispiel wird die Sicherheitsgruppe mit dem Namen `MySecurityGroup` gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] Um eine Sicherheitsgruppe zu löschen

In diesem Beispiel wird die Sicherheitsgruppe mit der ID `sg-903004f8` gelöscht. Beachten Sie, dass Sie für EC2 - VPC nicht anhand des Namens auf eine Sicherheitsgruppe verweisen können. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-security-group --group-id sg-903004f8
```

Weitere Informationen finden Sie unter Verwenden von Sicherheitsgruppen im Benutzerhandbuch für die AWS -Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [DeleteSecurityGroup](#) in der AWS CLI Befehlsreferenz.

delete-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-snapshot`.

AWS CLI

So löschen Sie einen Snapshot

Dieser Beispielbefehl löscht einen Snapshot mit der Snapshot-ID von `snap-1234567890abcdef0`. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- API-Einheiten finden Sie [DeleteSnapshot](#) in der AWS CLI Befehlsreferenz.

delete-spot-datafeed-subscription

Das folgende Codebeispiel zeigt die Verwendung `delete-spot-datafeed-subscription`.

AWS CLI

Um ein Spot-Instance-Datenfeed-Abonnement zu kündigen

Dieser Beispielbefehl löscht ein Spot-Datenfeed-Abonnement für das Konto. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-spot-datafeed-subscription
```

- API-Einheiten finden Sie [DeleteSpotDatafeedSubscription](#) in der AWS CLI Befehlsreferenz.

delete-subnet-cidr-reservation

Das folgende Codebeispiel zeigt die Verwendung `delete-subnet-cidr-reservation`.

AWS CLI

Um eine CIDR Subnetzreservierung zu löschen

Im folgenden `delete-subnet-cidr-reservation` Beispiel wird die angegebene CIDR Subnetzreservierung gelöscht.

```
aws ec2 delete-subnet-cidr-reservation \
  --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

Ausgabe:

```
{
  "DeletedSubnetCidrReservation": {
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",
    "Cidr": "10.1.0.16/28",
    "ReservationType": "prefix",
    "OwnerId": "123456789012"
  }
}
```

Weitere Informationen finden Sie unter [CIDR Subnetzreservierungen](#) im VPC Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSubnetCidrReservation](#) in der AWS CLI Befehlsreferenz.

delete-subnet

Das folgende Codebeispiel zeigt die Verwendung `delete-subnet`.

AWS CLI

Um ein Subnetz zu löschen

In diesem Beispiel wird das angegebene Subnetz gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- API-Einheiten finden Sie unter [DeleteSubnet AWS CLI](#) Befehlsreferenz.

delete-tags

Das folgende Codebeispiel zeigt die Verwendung `delete-tags`.

AWS CLI

Beispiel 1: Um ein Tag aus einer Ressource zu löschen

Im folgenden `delete-tags` Beispiel wird das Tag `Stack=Test` aus dem angegebenen Bild gelöscht. Wenn Sie sowohl einen Wert als auch einen Schlüsselnamen angeben, wird das Tag nur gelöscht, wenn der Wert des Tags dem angegebenen Wert entspricht.

```
aws ec2 delete-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=Test
```

Es ist optional, den Wert für ein Tag anzugeben. Im folgenden `delete-tags` Beispiel wird das Tag mit dem Schlüsselnamen `purpose` aus der angegebenen Instanz gelöscht, unabhängig vom Tag-Wert für das Tag.

```
aws ec2 delete-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags purpose
```

```
--resources i-1234567890abcdef0 \  
--tags Key=purpose
```

Wenn Sie die leere Zeichenfolge als Tag-Wert angeben, wird das Tag nur gelöscht, wenn der Tag-Wert eine leere Zeichenfolge ist. Im folgenden `delete-tags` Beispiel wird die leere Zeichenfolge als Tag-Wert für das zu löschende Tag angegeben.

```
aws ec2 delete-tags \  
--resources i-1234567890abcdef0 \  
--tags Key=Name, Value=
```

Beispiel 2: Um ein Tag aus mehreren Ressourcen zu löschen

Das folgende `delete-tags` Beispiel löscht das Tag `purpose=test` sowohl aus einer Instanz als auch aus einer AMI. Wie im vorherigen Beispiel gezeigt, können Sie den Tag-Wert im Befehl weglassen.

```
aws ec2 delete-tags \  
--resources i-1234567890abcdef0 ami-1234567890abcdef0 \  
--tags Key=Purpose
```

- API-Einheiten finden Sie unter [DeleteTags AWS CLI](#) Befehlsreferenz.

delete-traffic-mirror-filter-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-traffic-mirror-filter-rule`.

AWS CLI

Um eine Traffic Mirror-Filterregel zu löschen

Im folgenden `delete-traffic-mirror-filter-rule` Beispiel wird die angegebene Filterregel für den Traffic Mirror gelöscht.

```
aws ec2 delete-traffic-mirror-filter-rule \  
--traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

Ausgabe:

```
{
```

```
"TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Ändern Ihrer Traffic Mirroring-Filterregeln](#) im AWS Traffic Mirroring Guide.

- API-Einheiten finden Sie unter [DeleteTrafficMirrorFilterRule AWS CLI](#) Befehlsreferenz.

delete-traffic-mirror-filter

Das folgende Codebeispiel zeigt die Verwendung `delete-traffic-mirror-filter`.

AWS CLI

Um einen Traffic Mirrorfilter zu löschen

Im folgenden `delete-traffic-mirror-filter` Beispiel wird der angegebene Traffic-Spiegelfilter gelöscht.

```
aws ec2 delete-traffic-mirror-filter \  
  --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

Ausgabe:

```
{  
  "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"  
}
```

Weitere Informationen finden Sie im Traffic Mirroring-Handbuch unter [Löschen eines Traffic Mirroring-Filters](#).AWS

- API-Einheiten finden Sie unter [DeleteTrafficMirrorFilter AWS CLI](#) Befehlsreferenz.

delete-traffic-mirror-session

Das folgende Codebeispiel zeigt die Verwendung `delete-traffic-mirror-session`.

AWS CLI

Um eine Traffic Mirror-Sitzung zu löschen

Im folgenden `delete-traffic-mirror-session` Beispiel wird die angegebene Traffic-Spiegelsitzung gelöscht.

```
aws ec2 delete-traffic-mirror-session \  
  --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

Ausgabe:

```
{  
  "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Löschen einer Traffic Mirror-Sitzung im Traffic Mirroring](#) AWS Guide.

- API-Einheiten finden Sie unter [DeleteTrafficMirrorSession AWS CLI](#) Befehlsreferenz.

delete-traffic-mirror-target

Das folgende Codebeispiel zeigt die Verwendung `delete-traffic-mirror-target`.

AWS CLI

Um ein Traffic-Mirror-Ziel zu löschen

Im folgenden `delete-traffic-mirror-target` Beispiel wird das angegebene Traffic-Mirror-Ziel gelöscht.

```
aws ec2 delete-traffic-mirror-target \  
  --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

Ausgabe:

```
{  
  "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Löschen eines Traffic Mirror-Ziels](#) im AWS Traffic Mirroring Guide.

- API-Einheiten finden Sie [DeleteTrafficMirrorTarget](#) in der AWS CLI Befehlsreferenz.

delete-transit-gateway-connect-peer

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-connect-peer`.

AWS CLI

So löschen Sie einen Transit Gateway Connect-Peer

Im folgenden `delete-transit-gateway-connect-peer` Beispiel wird der angegebene Connect-Peer gelöscht.

```
aws ec2 delete-transit-gateway-connect-peer \
  --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayConnectPeer": {
    "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
    "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
    "State": "deleting",
    "CreationTime": "2021-10-13T03:35:17.000Z",
    "ConnectPeerConfiguration": {
      "TransitGatewayAddress": "10.0.0.234",
      "PeerAddress": "172.31.1.11",
      "InsideCidrBlocks": [
        "169.254.6.0/29"
      ],
      "Protocol": "gre",
      "BgpConfigurations": [
        {
          "TransitGatewayAsn": 64512,
          "PeerAsn": 64512,
          "TransitGatewayAddress": "169.254.6.2",
          "PeerAddress": "169.254.6.1",
          "BgpStatus": "down"
        },
        {
          "TransitGatewayAsn": 64512,
          "PeerAsn": 64512,
          "TransitGatewayAddress": "169.254.6.3",
          "PeerAddress": "169.254.6.1",
          "BgpStatus": "down"
        }
      ]
    }
  }
}
```


- API-Einheiten finden Sie unter [DeleteTransitGatewayConnect AWS CLI Befehlsreferenz](#).

delete-transit-gateway-multicast-domain

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-multicast-domain`.

AWS CLI

Um eine Transit-Gateway-Multicast-Domäne zu löschen

Im folgenden `delete-transit-gateway-multicast-domain` Beispiel wird die angegebene Multicast-Domäne gelöscht.

```
aws ec2 delete-transit-gateway-multicast-domain \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
    "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
    "State": "deleting",
    "CreationTime": "2019-11-20T22:02:03.000Z"
  }
}
```

Weitere Informationen finden Sie im Transit [Gateways Guide unter Managing Multicast-Domains](#).

- API-Einheiten finden Sie [DeleteTransitGatewayMulticastDomain](#) in der AWS CLI Befehlsreferenz.

delete-transit-gateway-peering-attachment

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-peering-attachment`.

AWS CLI

Um einen Transit-Gateway-Peering-Anhang zu löschen

Im folgenden `delete-transit-gateway-peering-attachment` Beispiel wird der angegebene Transit-Gateway-Peering-Anhang gelöscht.

```
aws ec2 delete-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

Ausgabe:

```
{
  "TransitGatewayPeeringAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
    "RequesterTgwInfo": {
      "TransitGatewayId": "tgw-123abc05e04123abc",
      "OwnerId": "123456789012",
      "Region": "us-west-2"
    },
    "AcceptorTgwInfo": {
      "TransitGatewayId": "tgw-11223344aabbcc112",
      "OwnerId": "123456789012",
      "Region": "us-east-2"
    },
    "State": "deleting",
    "CreationTime": "2019-12-09T11:38:31.000Z"
  }
}
```

Weitere Informationen finden Sie unter [Transit Gateway Peering Attachments](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DeleteTransitGatewayPeeringAttachment AWS CLI](#) Befehlsreferenz.

delete-transit-gateway-policy-table

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-policy-table`.

AWS CLI

Um eine Richtlinientabelle für ein Transit Gateway zu löschen

Im folgenden `delete-transit-gateway-policy-table` Beispiel wird die angegebene Richtlinientabelle für das Transit Gateway gelöscht.

```
aws ec2 delete-transit-gateway-policy-table \
  --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

Ausgabe:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",
      "State": "deleting",
      "CreationTime": "2023-11-28T16:36:43+00:00",
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie in den [Richtlinientabellen für Transit Gateway](#) im Transit Gateway-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteTransitGatewayPolicyTable](#) unter AWS CLI Befehlsreferenz.

delete-transit-gateway-prefix-list-reference

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-prefix-list-reference`.

AWS CLI

Um einen Verweis auf eine Präfixliste zu löschen

Im folgenden `delete-transit-gateway-prefix-list-reference` Beispiel wird die angegebene Präfixlistenreferenz gelöscht.

```
aws ec2 delete-transit-gateway-prefix-list-reference \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --prefix-list-id pl-11111122222222333
```

Ausgabe:

```
{
```

```

    "TransitGatewayPrefixListReference": {
      "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
      "PrefixListId": "pl-1111112222222333",
      "PrefixListOwnerId": "123456789012",
      "State": "deleting",
      "Blackhole": false,
      "TransitGatewayAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
        "ResourceType": "vpc",
        "ResourceId": "vpc-112233445566aabbcc"
      }
    }
  }
}

```

Weitere Informationen finden Sie unter [Referenzen zur Präfixliste](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Delete Transit Gateway Prefix List Reference AWS CLI Befehlsreferenz](#).

delete-transit-gateway-route-table

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-route-table`.

AWS CLI

Um eine Transit-Gateway-Routentabelle zu löschen

Im folgenden `delete-transit-gateway-route-table` Beispiel wird die angegebene Transit-Gateway-Routentabelle gelöscht.

```

aws ec2 delete-transit-gateway-route-table \
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE

```

Ausgabe:

```

{
  "TransitGatewayRouteTable": {
    "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "State": "deleting",
    "DefaultAssociationRouteTable": false,
    "DefaultPropagationRouteTable": false,
    "CreationTime": "2019-07-17T20:27:26.000Z"
  }
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Löschen einer Transit-Gateway-Routentabelle](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DeleteTransitGatewayRouteTable AWS CLI Befehlsreferenz](#).

delete-transit-gateway-route

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-route`.

AWS CLI

Um einen CIDR Block aus einer Routentabelle zu löschen

Im folgenden `delete-transit-gateway-route` Beispiel wird der CIDR Block aus der angegebenen Transit-Gateway-Routentabelle gelöscht.

```
aws ec2 delete-transit-gateway-route \  
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \  
  --destination-cidr-block 10.0.2.0/24
```

Ausgabe:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  
      {  
        "ResourceId": "vpc-0065acced4EXAMPLE",  
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
        "ResourceType": "vpc"  
      }  
    ],  
    "Type": "static",  
    "State": "deleted"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen einer statischen Route](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DeleteTransitGatewayRoute AWS CLI-Befehlsreferenz](#).

delete-transit-gateway-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway-vpc-attachment`.

AWS CLI

Um einen VPC Transit-Gateway-Anhang zu löschen

Im folgenden `delete-transit-gateway-vpc-attachment` Beispiel wird der angegebene VPC-Anhang gelöscht.

```
aws ec2 delete-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "VpcId": "vpc-0065acced4f61c651",
    "VpcOwnerId": "111122223333",
    "State": "deleting",
    "CreationTime": "2019-07-17T16:04:27.000Z"
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines VPC-Anhangs](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DeleteTransitGatewayVpcAttachment AWS CLI-Befehlsreferenz](#).

delete-transit-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-transit-gateway`.

AWS CLI

Um ein Transit-Gateway zu löschen

Im folgenden `delete-transit-gateway` Beispiel wird das angegebene Transit-Gateway gelöscht.

```
aws ec2 delete-transit-gateway \  
  --transit-gateway-id tgw-01f04542b2EXAMPLE
```

Ausgabe:

```
{  
  "TransitGateway": {  
    "TransitGatewayId": "tgw-01f04542b2EXAMPLE",  
    "State": "deleting",  
    "OwnerId": "123456789012",  
    "Description": "Example Transit Gateway",  
    "CreationTime": "2019-08-27T15:04:35.000Z",  
    "Options": {  
      "AmazonSideAsn": 64515,  
      "AutoAcceptSharedAttachments": "disable",  
      "DefaultRouteTableAssociation": "enable",  
      "AssociationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",  
      "DefaultRouteTablePropagation": "enable",  
      "PropagationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",  
      "VpnEcmpSupport": "enable",  
      "DnsSupport": "enable"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen eines Transit-Gateways](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DeleteTransitGateway AWS CLI Befehlsreferenz](#).

delete-verified-access-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-verified-access-endpoint`.

AWS CLI

Um einen Verified Access-Endpunkt zu löschen

Im folgenden `delete-verified-access-endpoint` Beispiel wird der angegebene Endpunkt für verifizierten Zugriff gelöscht.

```
aws ec2 delete-verified-access-endpoint \  
  --verified-access-endpoint-id vae-066fac616d4d546f2
```

Ausgabe:

```
{  
  "VerifiedAccessEndpoint": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",  
    "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",  
    "ApplicationDomain": "example.com",  
    "EndpointType": "network-interface",  
    "AttachmentType": "vpc",  
    "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/  
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",  
    "EndpointDomain": "my-ava-  
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-  
east-2.amazonaws.com",  
    "SecurityGroupIds": [  
      "sg-004915970c4c8f13a"  
    ],  
    "NetworkInterfaceOptions": {  
      "NetworkInterfaceId": "eni-0aec70418c8d87a0f",  
      "Protocol": "https",  
      "Port": 443  
    },  
    "Status": {  
      "Code": "deleting"  
    },  
    "Description": "Testing Verified Access",  
    "CreationTime": "2023-08-25T20:54:43",  
    "LastUpdatedTime": "2023-08-25T22:46:32"  
  }  
}
```

Weitere Informationen finden Sie unter [Verified Access-Endpoints](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVerifiedAccessEndpoint](#) in der AWS CLI Befehlsreferenz.

delete-verified-access-group

Das folgende Codebeispiel zeigt die Verwendung `delete-verified-access-group`.

AWS CLI

Um eine Gruppe mit verifiziertem Zugriff zu löschen

Im folgenden `delete-verified-access-group` Beispiel wird die angegebene Verified Access-Gruppe gelöscht.

```
aws ec2 delete-verified-access-group \
  --verified-access-group-id vagr-0dbe967baf14b7235
```

Ausgabe:

```
{
  "VerifiedAccessGroup": {
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "Owner": "123456789012",
    "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
    "CreationTime": "2023-08-25T19:55:19",
    "LastUpdatedTime": "2023-08-25T22:49:03",
    "DeletionTime": "2023-08-26T00:58:31"
  }
}
```

Weitere Informationen finden Sie unter [Verified Access-Gruppen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVerifiedAccessGroup](#) in der AWS CLI Befehlsreferenz.

delete-verified-access-instance

Das folgende Codebeispiel zeigt die Verwendung `delete-verified-access-instance`.

AWS CLI

Um eine Verified Access-Instanz zu löschen

Im folgenden `delete-verified-access-instance` Beispiel wird die angegebene Verified Access-Instanz gelöscht.

```
aws ec2 delete-verified-access-instance \  
  --verified-access-instance-id vai-0ce000c0b7643abea
```

Ausgabe:

```
{  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "Testing Verified Access",  
    "VerifiedAccessTrustProviders": [],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-26T01:00:18"  
  }  
}
```

Weitere Informationen finden Sie unter [Verified Access-Instanzen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVerifiedAccessInstance](#) in der AWS CLI Befehlsreferenz.

delete-verified-access-trust-provider

Das folgende Codebeispiel zeigt die Verwendung `delete-verified-access-trust-provider`.

AWS CLI

Um einen Vertrauensanbieter mit verifiziertem Zugriff zu löschen

Im folgenden `delete-verified-access-trust-provider` Beispiel wird der angegebene Verified Access-Vertrauensanbieter gelöscht.

```
aws ec2 delete-verified-access-trust-provider \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Ausgabe:

```
{  
  "VerifiedAccessTrustProvider": {
```

```
"VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
  "Description": "Testing Verified Access",
  "TrustProviderType": "user",
  "UserTrustProviderType": "iam-identity-center",
  "PolicyReferenceName": "idc",
  "CreationTime": "2023-08-25T18:40:36",
  "LastUpdatedTime": "2023-08-25T18:40:36"
}
```

Weitere Informationen finden Sie unter [Vertrauensanbietern für verifizierten Zugriff](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteVerifiedAccessTrustProvider](#) in der AWS CLI Befehlsreferenz.

delete-volume

Das folgende Codebeispiel zeigt die Verwendung `delete-volume`.

AWS CLI

Um ein Volume zu löschen

Dieser Beispielbefehl löscht ein verfügbares Volume mit der Volume-ID `vol-049df61146c4d7901`. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- API-Einheiten finden Sie [DeleteVolume](#) in der AWS CLI Befehlsreferenz.

delete-vpc-endpoint-connection-notifications

Das folgende Codebeispiel zeigt die Verwendung `delete-vpc-endpoint-connection-notifications`.

AWS CLI

Um eine Benachrichtigung über eine Endpunktverbindung zu löschen

In diesem Beispiel wird die angegebene Endpunktverbindungsbenachrichtigung gelöscht.

Befehl:

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

Ausgabe:

```
{  
  "Unsuccessful": []  
}
```

- API-Einheiten finden Sie [DeleteVpcEndpointConnectionNotifications](#) in der AWS CLI Befehlsreferenz.

delete-vpc-endpoint-service-configurations

Das folgende Codebeispiel zeigt die Verwendung `delete-vpc-endpoint-service-configurations`.

AWS CLI

Um eine Endpunkt-Servicekonfiguration zu löschen

In diesem Beispiel wird die angegebene Endpunktdienstkonfiguration gelöscht.

Befehl:

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

Ausgabe:

```
{  
  "Unsuccessful": []  
}
```

- API-Einheiten finden Sie [DeleteVpcEndpointServiceConfigurations](#) in der AWS CLI Befehlsreferenz.

delete-vpc-endpoints

Das folgende Codebeispiel zeigt die Verwendung `delete-vpc-endpoints`.

AWS CLI

Um einen Endpunkt zu löschen

In diesem Beispiel werden die Endpunkte `vpce-aa22bb33` und `vpce-1a2b3c4d` gelöscht. Wenn der Befehl teilweise erfolgreich oder nicht erfolgreich ist, wird eine Liste der erfolglosen Elemente zurückgegeben. Wenn der Befehl erfolgreich ist, ist die zurückgegebene Liste leer.

Befehl:

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

Ausgabe:

```
{
  "Unsuccessful": []
}
```

- API-Einheiten finden Sie [DeleteVpcEndpoints](#) in der AWS CLI Befehlsreferenz.

delete-vpc-peering-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-vpc-peering-connection`.

AWS CLI

Um eine VPC Peering-Verbindung zu löschen

In diesem Beispiel wird die angegebene VPC Peering-Verbindung gelöscht.

Befehl:

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden Sie unter [DeleteVpcPeeringConnection AWS CLI](#) Befehlsreferenz.

delete-vpc

Das folgende Codebeispiel zeigt die Verwendung `delete-vpc`.

AWS CLI

Um ein zu löschen VPC

In diesem Beispiel wird das angegebene VPC gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- API-Einheiten finden Sie [DeleteVpc](#) in der AWS CLI Befehlsreferenz.

delete-vpn-connection-route

Das folgende Codebeispiel zeigt die Verwendung `delete-vpn-connection-route`.

AWS CLI

Um eine statische Route aus einer VPN Verbindung zu löschen

In diesem Beispiel wird die angegebene statische Route aus der angegebenen VPN Verbindung gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-  
cidr-block 11.12.0.0/16
```

- API-Einheiten finden Sie unter [DeleteVpnConnectionRoute AWS CLI](#) Befehlsreferenz.

delete-vpn-connection

Das folgende Codebeispiel zeigt die Verwendung `delete-vpn-connection`.

AWS CLI

Um eine VPN Verbindung zu löschen

In diesem Beispiel wird die angegebene VPN Verbindung gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- API-Einheiten finden Sie unter [DeleteVpnConnection AWS CLI](#) Befehlsreferenz.

delete-vpn-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-vpn-gateway`.

AWS CLI

Um ein virtuelles privates Gateway zu löschen

In diesem Beispiel wird das angegebene virtuelle private Gateway gelöscht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- API-Einheiten finden Sie unter [DeleteVpnGateway AWS CLI](#) Befehlsreferenz.

deprovision-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `deprovision-byoip-cidr`.

AWS CLI

Um einen IP-Adressbereich aus der Verwendung zu entfernen

Im folgenden Beispiel wird der angegebene Adressbereich aus der Verwendung mit entfernt AWS.

```
aws ec2 deprovision-byoip-cidr \  
  --cidr 203.0.113.25/24
```

Ausgabe:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "State": "pending-deprovision"  
  }  
}
```

- API-Einheiten finden Sie [DeprovisionByoipCidr](#) unter AWS CLI Befehlsreferenz.

deprovision-ipam-pool-cidr

Das folgende Codebeispiel zeigt die Verwendung `deprovision-ipam-pool-cidr`.

AWS CLI

Um die Bereitstellung eines Pools aufzuheben IPAM CIDR

Im folgenden `deprovision-ipam-pool-cidr` Beispiel wird die CIDR-Bereitstellung eines bereitgestellten Elements für einen Pool aufgehoben. IPAM

(Linux):

```
aws ec2 deprovision-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-02ec043a19bbe5d08 \  
  --cidr 11.0.0.0/16
```

(Windows):

```
aws ec2 deprovision-ipam-pool-cidr ^  
  --ipam-pool-id ipam-pool-02ec043a19bbe5d08 ^  
  --cidr 11.0.0.0/16
```

Ausgabe:

```
{
  "IpamPoolCidr": {
    "Cidr": "11.0.0.0/16",
    "State": "pending-deprovision"
  }
}
```

Weitere Informationen finden Sie unter [Deprovision Pool CIDRs](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeprovisionIpamPoolCidr](#) in der AWS CLI Befehlsreferenz.

deregister-image

Das folgende Codebeispiel zeigt die Verwendung `deregister-image`.

AWS CLI

Um eine abzumelden AMI

In diesem Beispiel wird die angegebene Datei deregistriert. AMI Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- APIEinzelheiten finden Sie [DeregisterImage](#) in der AWS CLI Befehlsreferenz.

deregister-instance-event-notification-attributes

Das folgende Codebeispiel zeigt die Verwendung `deregister-instance-event-notification-attributes`.

AWS CLI

Beispiel 1: Um alle Tags aus Ereignisbenachrichtigungen zu entfernen

Im folgenden `deregister-instance-event-notification-attributes` Beispiel wird entfernt `IncludeAllTagsOfInstance=true`, was zur Folge hat, `IncludeAllTagsOfInstance` dass auf gesetzt wird `false`.

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute IncludeAllTagsOfInstance=true
```

Ausgabe:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [],  
    "IncludeAllTagsOfInstance": true  
  }  
}
```

Weitere Informationen finden Sie unter [Geplante Ereignisse für Ihre Instances](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: Um bestimmte Tags aus Ereignisbenachrichtigungen zu entfernen

Im folgenden `deregister-instance-event-notification-attributes` Beispiel wird das angegebene Tag aus den in den Ereignisbenachrichtigungen enthaltenen Tags entfernt. Um die verbleibenden Tags zu beschreiben, die in Ereignisbenachrichtigungen enthalten sind, verwenden `Siedescribe-instance-event-notification-attributes`.

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute InstanceTagKeys="tag-key2"
```

Ausgabe:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [  
      "tag-key2"  
    ],  
    "IncludeAllTagsOfInstance": false  
  }  
}
```

Weitere Informationen finden Sie unter [Geplante Ereignisse für Ihre Instances](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DeregisterInstanceEventNotificationAttributes](#) in der AWS CLI Befehlsreferenz.

deregister-transit-gateway-multicast-group-members

Das folgende Codebeispiel zeigt die Verwendung `deregister-transit-gateway-multicast-group-members`.

AWS CLI

Um Gruppenmitglieder von einer Multicast-Gruppe abzumelden

In diesem Beispiel wird das angegebene Mitglied der Netzwerkschnittstellengruppe von der Transit-Gateway-Multicast-Gruppe abgemeldet.

```
aws ec2 deregister-transit-gateway-multicast-group-members \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE \
  --group-ip-address 224.0.1.0 \
  --network-interface-ids eni-0e246d3269EXAMPLE
```

Ausgabe:

```
{
  "DeregisteredMulticastGroupMembers": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
    "RegisteredNetworkInterfaceIds": [
      "eni-0e246d3269EXAMPLE"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Weitere Informationen finden Sie unter [Mitglieder aus einer Multicast-Gruppe abmelden](#) im Transit Gateways-Benutzerhandbuch.AWS

- API-Einheiten finden Sie unter [DeregisterTransitGatewayMulticastGroupMembers](#) Befehlsreferenz.AWS CLI

deregister-transit-gateway-multicast-group-source

Das folgende Codebeispiel zeigt die Verwendung `deregister-transit-gateway-multicast-group-source`.

AWS CLI

Um eine Quelle von der Transit-Gateway-Multicast-Gruppe abzumelden

In diesem Beispiel wird die angegebene Netzwerkschnittstellengruppenquelle von der Multicast-Gruppe abgemeldet.

```
aws ec2 register-transit-gateway-multicast-group-sources \  
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \  
  --group-ip-address 224.0.1.0 \  
  --network-interface-ids eni-07f290fc3c090cbae
```

Ausgabe:

```
{  
  "DeregisteredMulticastGroupSources": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",  
    "DeregisteredNetworkInterfaceIds": [  
      "eni-07f290fc3c090cbae"  
    ],  
    "GroupIpAddress": "224.0.1.0"  
  }  
}
```

Weitere Informationen finden Sie unter [Abmelden von Quellen aus einer Multicast-Gruppe](#) im Transit Gateways-Benutzerhandbuch.AWS

- API-Einheiten finden Sie unter [DeregisterTransitGatewayMulticastGroupSource](#) Befehlsreferenz.AWS CLI

describe-account-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-account-attributes`.

AWS CLI

Um alle Attribute für Ihr AWS Konto zu beschreiben

In diesem Beispiel werden die Attribute für Ihr AWS Konto beschrieben.

Befehl:

aws ec2 describe-account-attributes

Ausgabe:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "vpc-max-security-groups-per-interface",
      "AttributeValues": [
        {
          "AttributeValue": "5"
        }
      ]
    },
    {
      "AttributeName": "max-instances",
      "AttributeValues": [
        {
          "AttributeValue": "20"
        }
      ]
    },
    {
      "AttributeName": "supported-platforms",
      "AttributeValues": [
        {
          "AttributeValue": "EC2"
        },
        {
          "AttributeValue": "VPC"
        }
      ]
    },
    {
      "AttributeName": "default-vpc",
      "AttributeValues": [
        {
          "AttributeValue": "none"
        }
      ]
    },
    {
      "AttributeName": "max-elastic-ips",
```

```
    "AttributeValues": [
      {
        "AttributeValue": "5"
      }
    ],
    {
      "AttributeName": "vpc-max-elastic-ips",
      "AttributeValues": [
        {
          "AttributeValue": "5"
        }
      ]
    }
  ]
}
```

Um ein einzelnes Attribut für Ihr AWS Konto zu beschreiben

Dieses Beispiel beschreibt das `supported-platforms` Attribut für Ihr AWS Konto.

Befehl:

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

Ausgabe:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "supported-platforms",
      "AttributeValues": [
        {
          "AttributeValue": "EC2"
        },
        {
          "AttributeValue": "VPC"
        }
      ]
    }
  ]
}
```


- API-Einheiten finden Sie [DescribeAccountAttributes](#) in der AWS CLI Befehlsreferenz.

describe-address-transfers

Das folgende Codebeispiel zeigt die Verwendung `describe-address-transfers`.

AWS CLI

Um eine Elastic IP-Adressübertragung zu beschreiben

Das folgende `describe-address-transfers` Beispiel beschreibt die Elastic IP-Adressübertragung für die angegebene Elastic IP-Adresse.

```
aws ec2 describe-address-transfers \
  --allocation-ids eipalloc-09ad461b0d03f6aaf
```

Ausgabe:

```
{
  "AddressTransfers": [
    {
      "PublicIp": "100.21.184.216",
      "AllocationId": "eipalloc-09ad461b0d03f6aaf",
      "TransferAccountId": "123456789012",
      "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",
      "AddressTransferStatus": "pending"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Transfer Elastic IP-Adressen](#) im VPC Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAddressTransfers](#) in der AWS CLI Befehlsreferenz.

describe-addresses-attribute

Das folgende Codebeispiel zeigt die Verwendung `describe-addresses-attribute`.

AWS CLI

Um die Attribute des Domainnamens anzuzeigen, der einer elastischen IP-Adresse zugeordnet ist

In den folgenden `describe-addresses-attribute` Beispielen werden die Attribute des Domainnamens zurückgegeben, der der elastischen IP-Adresse zugeordnet ist.

Linux:

```
aws ec2 describe-addresses-attribute \  
  --allocation-ids eipalloc-abcdef01234567890 \  
  --attribute domain-name
```

Windows:

```
aws ec2 describe-addresses-attribute ^  
  --allocation-ids eipalloc-abcdef01234567890 ^  
  --attribute domain-name
```

Ausgabe:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.com."  
    }  
  ]  
}
```

Um die Attribute einer Elastic IP-Adresse anzuzeigen, müssen Sie der Elastic IP-Adresse zunächst einen Domainnamen zugeordnet haben. Weitere Informationen finden Sie unter [Verwenden von Reverse DNS für E-Mail-Anwendungen](#) im EC2Amazon-Benutzerhandbuch oder [modify-address-attribute](#) in der AWS CLIBefehlsreferenz.

- APIEinzelheiten finden Sie [DescribeAddressesAttribute](#) unter AWS CLI Befehlsreferenz.

describe-addresses

Das folgende Codebeispiel zeigt die Verwendung `describe-addresses`.

AWS CLI

Beispiel 1: So rufen Sie Details über alle Ihre Elastic-IP-Adressen ab

Im folgenden `describe-addresses`-Beispiel werden Details zu Ihren Elastic-IP-Adressen angezeigt.

```
aws ec2 describe-addresses
```

Ausgabe:

```
{
  "Addresses": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "PublicIp": "198.51.100.0",
      "PublicIpv4Pool": "amazon",
      "Domain": "standard"
    },
    {
      "Domain": "vpc",
      "PublicIpv4Pool": "amazon",
      "InstanceId": "i-1234567890abcdef0",
      "NetworkInterfaceId": "eni-12345678",
      "AssociationId": "eipassoc-12345678",
      "NetworkInterfaceOwnerId": "123456789012",
      "PublicIp": "203.0.113.0",
      "AllocationId": "eipalloc-12345678",
      "PrivateIpAddress": "10.0.1.241"
    }
  ]
}
```

Beispiel 2: Um Details Ihrer Elastic IP-Adressen abzurufen für EC2 - VPC

Im folgenden `describe-addresses` Beispiel werden Details zu Ihren Elastic IP-Adressen für die Verwendung mit Instances in a angezeigtVPC.

```
aws ec2 describe-addresses \
  --filters "Name=domain,Values=vpc"
```

Ausgabe:

```
{
  "Addresses": [
    {
```

```

        "Domain": "vpc",
        "PublicIpv4Pool": "amazon",
        "InstanceId": "i-1234567890abcdef0",
        "NetworkInterfaceId": "eni-12345678",
        "AssociationId": "eipassoc-12345678",
        "NetworkInterfaceOwnerId": "123456789012",
        "PublicIp": "203.0.113.0",
        "AllocationId": "eipalloc-12345678",
        "PrivateIpAddress": "10.0.1.241"
    }
]
}

```

Beispiel 3: So rufen Sie Details über eine durch die Zuweisungs-ID spezifizierte Elastic-IP-Adresse ab

Im folgenden `describe-addresses` Beispiel werden Details zur Elastic IP-Adresse mit der angegebenen Zuweisungs-ID angezeigt, die einer Instance in EC2 - zugeordnet istVPC.

```

aws ec2 describe-addresses \
  --allocation-ids eipalloc-282d9641

```

Ausgabe:

```

{
  "Addresses": [
    {
      "Domain": "vpc",
      "PublicIpv4Pool": "amazon",
      "InstanceId": "i-1234567890abcdef0",
      "NetworkInterfaceId": "eni-1a2b3c4d",
      "AssociationId": "eipassoc-123abc12",
      "NetworkInterfaceOwnerId": "1234567891012",
      "PublicIp": "203.0.113.25",
      "AllocationId": "eipalloc-282d9641",
      "PrivateIpAddress": "10.251.50.12"
    }
  ]
}

```

Beispiel 4: Um Details zu einer Elastic IP-Adresse abzurufen, die durch ihre VPC private IP-Adresse spezifiziert ist

Im folgenden `describe-addresses` Beispiel werden Details über die Elastic IP-Adresse angezeigt, die einer bestimmten privaten IP-Adresse in EC2 - zugeordnet ist/VPC.

```
aws ec2 describe-addresses \  
  --filters "Name=private-ip-address,Values=10.251.50.12"
```

Beispiel 5: Um Details zu Elastic IP-Adressen in EC2 -Classic abzurufen

Im folgenden `describe-addresses` Beispiel werden Details zu Ihren Elastic IP-Adressen für die Verwendung in EC2 -Classic angezeigt.

```
aws ec2 describe-addresses \  
  --filters "Name=domain,Values=standard"
```

Ausgabe:

```
{  
  "Addresses": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "PublicIp": "203.0.110.25",  
      "PublicIpv4Pool": "amazon",  
      "Domain": "standard"  
    }  
  ]  
}
```

Beispiel 6: So rufen Sie Details über eine Elastic-IP-Adresse ab, die durch ihre öffentliche IP-Adresse angegeben ist

Im folgenden `describe-addresses` Beispiel werden Details zur Elastic IP-Adresse mit dem Wert angezeigt `203.0.110.25`, die einer Instance in EC2 -Classic zugeordnet ist.

```
aws ec2 describe-addresses \  
  --public-ips 203.0.110.25
```

Ausgabe:

```
{
```

```
"Addresses": [  
  {  
    "InstanceId": "i-1234567890abcdef0",  
    "PublicIp": "203.0.110.25",  
    "PublicIpv4Pool": "amazon",  
    "Domain": "standard"  
  }  
]  
}
```

- API-Einheiten finden Sie unter [DescribeAddresses AWS CLI](#) Befehlsreferenz.

describe-aggregate-id-format

Das folgende Codebeispiel zeigt die Verwendung `describe-aggregate-id-format`.

AWS CLI

Um die Einstellungen für das längere ID-Format für alle Ressourcentypen in einer Region zu beschreiben

Das folgende `describe-aggregate-id-format` Beispiel beschreibt den Gesamtstatus des langen ID-Formats für die aktuelle Region. Der `Deadline` Wert gibt an, dass die Fristen für die dauerhafte Umstellung dieser Ressourcen vom Short-ID-Format auf das Long-ID-Format abgelaufen sind. Der `UseLongIdsAggregated` Wert gibt an, dass alle IAM Benutzer und IAM Rollen so konfiguriert sind, dass sie das Long-ID-Format für alle Ressourcentypen verwenden.

```
aws ec2 describe-aggregate-id-format
```

Ausgabe:

```
{  
  "UseLongIdsAggregated": true,  
  "Statuses": [  
    {  
      "Deadline": "2018-08-13T02:00:00.000Z",  
      "Resource": "network-interface-attachment",  
      "UseLongIds": true  
    },  
    {  
      "Deadline": "2016-12-13T02:00:00.000Z",
```

```
        "Resource": "instance",
        "UseLongIds": true
    },
    {
        "Deadline": "2018-08-13T02:00:00.000Z",
        "Resource": "elastic-ip-association",
        "UseLongIds": true
    },
    ...
]
}
```

- API-Einheiten finden Sie [DescribeAggregatIdFormat](#) unter AWS CLI Befehlsreferenz.

describe-availability-zones

Das folgende Codebeispiel zeigt die Verwendung `describe-availability-zones`.

AWS CLI

So beschreiben Sie Ihre Availability Zones

Das folgende Beispiel `describe-availability-zones` zeigt Details zu den Availability Zones, die für Sie verfügbar sind. Die Antwort umfasst nur Availability Zones für die aktuelle Region. In diesem Beispiel wird die Standardregion `us-west-2` (Oregon) des Profils verwendet.

```
aws ec2 describe-availability-zones
```

Ausgabe:

```
{
  "AvailabilityZones": [
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2a",
      "ZoneId": "usw2-az1",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2"
    },
  ],
}
```

```
{
  "State": "available",
  "OptInStatus": "opt-in-not-required",
  "Messages": [],
  "RegionName": "us-west-2",
  "ZoneName": "us-west-2b",
  "ZoneId": "usw2-az2",
  "GroupName": "us-west-2",
  "NetworkBorderGroup": "us-west-2"
},
{
  "State": "available",
  "OptInStatus": "opt-in-not-required",
  "Messages": [],
  "RegionName": "us-west-2",
  "ZoneName": "us-west-2c",
  "ZoneId": "usw2-az3",
  "GroupName": "us-west-2",
  "NetworkBorderGroup": "us-west-2"
},
{
  "State": "available",
  "OptInStatus": "opt-in-not-required",
  "Messages": [],
  "RegionName": "us-west-2",
  "ZoneName": "us-west-2d",
  "ZoneId": "usw2-az4",
  "GroupName": "us-west-2",
  "NetworkBorderGroup": "us-west-2"
},
{
  "State": "available",
  "OptInStatus": "opted-in",
  "Messages": [],
  "RegionName": "us-west-2",
  "ZoneName": "us-west-2-lax-1a",
  "ZoneId": "usw2-lax1-az1",
  "GroupName": "us-west-2-lax-1",
  "NetworkBorderGroup": "us-west-2-lax-1"
}
]
```

- API-Einheiten finden Sie [DescribeAvailabilityZones](#) in der AWS CLI Befehlsreferenz.

describe-aws-network-performance-metric-subscription

Das folgende Codebeispiel zeigt die Verwendung `describe-aws-network-performance-metric-subscription`.

AWS CLI

Um Ihre metrischen Abonnements zu beschreiben

Im folgenden `describe-aws-network-performance-metric-subscriptions` Beispiel werden Ihre metrischen Abonnements beschrieben.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Ausgabe:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Abonnements verwalten](#) im Infrastructure Performance User Guide.

- API-Einheiten finden Sie [DescribeAwsNetworkPerformanceMetricSubscription](#) in der AWS CLI Befehlsreferenz.

describe-aws-network-performance-metric-subscriptions

Das folgende Codebeispiel zeigt die Verwendung `describe-aws-network-performance-metric-subscriptions`.

AWS CLI

Um Ihre metrischen Abonnements zu beschreiben

Im folgenden `describe-aws-network-performance-metric-subscriptions` Beispiel werden Ihre metrischen Abonnements beschrieben.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Ausgabe:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Abonnements verwalten](#) im Infrastructure Performance User Guide.

- API-Einheiten finden Sie [DescribeAwsNetworkPerformanceMetricSubscriptions](#) in der AWS CLI Befehlsreferenz.

describe-bundle-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-bundle-tasks`.

AWS CLI

Um Ihre Bundle-Aufgaben zu beschreiben

In diesem Beispiel werden alle Ihre Bundle-Aufgaben beschrieben.

Befehl:

```
aws ec2 describe-bundle-tasks
```

Ausgabe:

```
{
  "BundleTasks": [
    {
      "UpdateTime": "2015-09-15T13:26:54.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "Storage": {
        "S3": {
          "Prefix": "winami",
          "Bucket": "bundletasks"
        }
      },
      "State": "bundling",
      "StartTime": "2015-09-15T13:24:35.000Z",
      "Progress": "3%",
      "BundleId": "bun-2a4e041c"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeBundleTasks](#) in der AWS CLI Befehlsreferenz.

describe-byoip-cidrs

Das folgende Codebeispiel zeigt die Verwendung `describe-byoip-cidrs`.

AWS CLI

Um Ihre bereitgestellten Adressbereiche zu beschreiben

Im folgenden `describe-byoip-cidrs` Beispiel werden Details zu den öffentlichen IPv4 Adressbereichen angezeigt, die Sie für die Verwendung durch bereitgestellt haben. AWS

```
aws ec2 describe-byoip-cidrs
```

Ausgabe:

```
{
  "ByoipCidrs": [
    {
      "Cidr": "203.0.113.25/24",
      "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    }
  ]
}
```

```

        "State": "provisioned"
      }
    ]
  }

```

- API-Einheiten finden Sie unter [DescribeByoipCidrs AWS CLI](#) Befehlsreferenz.

describe-capacity-reservation-fleets

Das folgende Codebeispiel zeigt die Verwendung `describe-capacity-reservation-fleets`.

AWS CLI

So zeigen Sie eine Flotte mit Kapazitätsreservierungen an

Das folgende `describe-capacity-reservation-fleets` Beispiel listet die Konfiguration und Kapazitätsinformationen für die angegebene Kapazitätsreservierungsflotte auf. Außerdem werden Details zu den einzelnen Kapazitätsreservierungen innerhalb der Flotte aufgeführt. :

```

aws ec2 describe-capacity-reservation-fleets \
  --capacity-reservation-fleet-ids crf-abcdef01234567890

```

Ausgabe:

```

{
  "CapacityReservationFleets": [
    {
      "Status": "active",
      "EndDate": "2022-12-31T23:59:59.000Z",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "CapacityReservationFleetId": "crf-abcdef01234567890",
      "Tenancy": "default",
      "InstanceTypeSpecifications": [
        {
          "CapacityReservationId": "cr-1234567890abcdef0",
          "AvailabilityZone": "us-east-1a",
          "FulfilledCapacity": 5.0,
          "Weight": 1.0,
          "CreateDate": "2022-07-02T08:34:33.398Z",
          "InstancePlatform": "Linux/UNIX",

```

```
        "TotalInstanceCount": 5,
        "Priority": 1,
        "EbsOptimized": true,
        "InstanceType": "m5.xlarge"
      }
    ],
    "TotalTargetCapacity": 5,
    "TotalFulfilledCapacity": 5.0,
    "CreateTime": "2022-07-02T08:34:33.397Z",
    "AllocationStrategy": "prioritized"
  }
]
```

Weitere Informationen zu Kapazitätsreservierungsflotten finden Sie unter [Kapazitätsreservierungsflotten](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [DescribeCapacityReservationFleets AWS CLI](#) Befehlsreferenz.

describe-capacity-reservations

Das folgende Codebeispiel zeigt die Verwendung `describe-capacity-reservations`.

AWS CLI

Beispiel 1: Um eine oder mehrere Ihrer Kapazitätsreservierungen zu beschreiben

Im folgenden `describe-capacity-reservations` Beispiel werden Details zu all Ihren Kapazitätsreservierungen in der aktuellen AWS Region angezeigt.

```
aws ec2 describe-capacity-reservations
```

Ausgabe:

```
{
  "CapacityReservations": [
    {
      "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
      "EndDateType": "unlimited",
      "AvailabilityZone": "eu-west-1a",
      "InstanceMatchCriteria": "open",
      "Tags": [],
```

```

    "EphemeralStorage": false,
    "CreateDate": "2019-08-16T09:03:18.000Z",
    "AvailableInstanceCount": 1,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 1,
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": true,
    "InstanceType": "a1.medium"
  },
  {
    "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
    "EndDateType": "unlimited",
    "AvailabilityZone": "eu-west-1a",
    "InstanceMatchCriteria": "open",
    "Tags": [],
    "EphemeralStorage": false,
    "CreateDate": "2019-08-07T11:34:19.000Z",
    "AvailableInstanceCount": 3,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 3,
    "State": "cancelled",
    "Tenancy": "default",
    "EbsOptimized": true,
    "InstanceType": "m5.large"
  }
]
}

```

Beispiel 2: Um eine oder mehrere Ihrer Kapazitätsreservierungen zu beschreiben

Im folgenden `describe-capacity-reservations` Beispiel werden Details zur angegebenen Kapazitätsreservierung angezeigt.

```

aws ec2 describe-capacity-reservations \
  --capacity-reservation-ids cr-1234abcd56EXAMPLE

```

Ausgabe:

```

{
  "CapacityReservations": [
    {

```

```
    "CapacityReservationId": "cr-1234abcd56EXAMPLE",
    "EndDateType": "unlimited",
    "AvailabilityZone": "eu-west-1a",
    "InstanceMatchCriteria": "open",
    "Tags": [],
    "EphemeralStorage": false,
    "CreateDate": "2019-08-16T09:03:18.000Z",
    "AvailableInstanceCount": 1,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 1,
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": true,
    "InstanceType": "a1.medium"
  }
]
```

Weitere Informationen finden Sie unter [Kapazitätsreservierung anzeigen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeCapacityReservations](#) unter AWS CLI Befehlsreferenz.

describe-carrier-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-carrier-gateways`.

AWS CLI

Um alle Carrier-Gateways zu beschreiben

Das folgende `describe-carrier-gateways` Beispiel listet alle Ihre Mobilfunkanbieter-Gateways auf.

```
aws ec2 describe-carrier-gateways
```

Ausgabe:

```
{
  "CarrierGateways": [
    {
      "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
```

```
    "VpcId": "vpc-0c529aEXAMPLE",
    "State": "available",
    "OwnerId": "123456789012",
    "Tags": [
      {
        "Key": "example",
        "Value": "tag"
      }
    ]
  }
]
```

Weitere Informationen finden Sie unter Carrier gateways< https://docs.aws.amazon.com/vpc/latest/userguide/Carrier_gateway.html> im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

- Einzelheiten finden Sie in der Befehlsreferenz. API [DescribeCarrierGateways](#)AWS CLI

describe-classic-link-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-classic-link-instances`.

AWS CLI

Um verknüpfte EC2 -Classic-Instanzen zu beschreiben

In diesem Beispiel werden alle Ihre verknüpften EC2 -Classic-Instanzen aufgeführt.

Befehl:

```
aws ec2 describe-classic-link-instances
```

Ausgabe:

```
{
  "Instances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "VpcId": "vpc-88888888",
      "Groups": [
```



```

        {
            "GroupId": "sg-11122233"
        }
    ],
    "Tags": [
        {
            "Value": "ClassicInstance",
            "Key": "Name"
        }
    ]
},
{
    "InstanceId": "i-0598c7d356eba48d7",
    "VpcId": "vpc-12312312",
    "Groups": [
        {
            "GroupId": "sg-aabbccdd"
        }
    ],
    "Tags": [
        {
            "Value": "ClassicInstance2",
            "Key": "Name"
        }
    ]
}
]
}

```

In diesem Beispiel werden alle Ihre verknüpften EC2 -Classic-Instanzen aufgeführt und die Antwort so gefiltert, dass sie nur Instanzen enthält, die mit vpc-88888888 verknüpft sind. VPC

Befehl:

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

Ausgabe:

```

{
    "Instances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "VpcId": "vpc-88888888",

```

```

        "Groups": [
            {
                "GroupId": "sg-11122233"
            }
        ],
        "Tags": [
            {
                "Value": "ClassicInstance",
                "Key": "Name"
            }
        ]
    }
}

```

- API-Einheiten finden Sie unter Befehlsreferenz. [DescribeClassicLinkInstances](#) AWS CLI

describe-client-vpn-authorization-rules

Das folgende Codebeispiel zeigt die Verwendung `describe-client-vpn-authorization-rules`.

AWS CLI

Um die Autorisierungsregeln für einen VPN Client-Endpunkt zu beschreiben

Im folgenden `describe-client-vpn-authorization-rules` Beispiel werden Details zu den Autorisierungsregeln für den angegebenen VPN Client-Endpunkt angezeigt.

```

aws ec2 describe-client-vpn-authorization-rules \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Ausgabe:

```

{
  "AuthorizationRules": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "GroupId": "",
      "AccessAll": true,
      "DestinationCidr": "0.0.0.0/0",
    }
  ]
}

```

```
        "Status": {
            "Code": "active"
        }
    ]
}
```

Weitere Informationen finden Sie unter [Autorisierungsregeln](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeClientVpnAuthorizationRules](#) in der AWS CLI Befehlsreferenz.

describe-client-vpn-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-client-vpn-connections`.

AWS CLI

Um die Verbindungen zu einem VPN Client-Endpunkt zu beschreiben

Im folgenden `describe-client-vpn-connections` Beispiel werden Details zu den Client-Verbindungen zum angegebenen VPN Client-Endpunkt angezeigt.

```
aws ec2 describe-client-vpn-connections \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Ausgabe:

```
{
  "Connections": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "Timestamp": "2019-08-12 07:58:34",
      "ConnectionId": "cvpn-connection-0e03eb24267165acd",
      "ConnectionEstablishedTime": "2019-08-12 07:57:14",
      "IngressBytes": "32302",
      "EgressBytes": "5696",
      "IngressPackets": "332",
      "EgressPackets": "67",
      "ClientIp": "172.31.0.225",
    }
  ]
}
```

```

        "CommonName": "client1.domain.tld",
        "Status": {
            "Code": "terminated"
        },
        "ConnectionEndTime": "2019-08-12 07:58:34"
    },
    {
        "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
        "Timestamp": "2019-08-12 08:02:54",
        "ConnectionId": "cvpn-connection-00668867a40f18253",
        "ConnectionEstablishedTime": "2019-08-12 08:02:53",
        "IngressBytes": "2951",
        "EgressBytes": "2611",
        "IngressPackets": "9",
        "EgressPackets": "6",
        "ClientIp": "172.31.0.226",
        "CommonName": "client1.domain.tld",
        "Status": {
            "Code": "active"
        },
        "ConnectionEndTime": "-"
    }
]
}

```

Weitere Informationen finden Sie unter [Client-Verbindungen](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeClientVpnConnections](#) in der AWS CLI Befehlsreferenz.

describe-client-vpn-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-client-vpn-endpoints`.

AWS CLI

Um Ihre VPN Client-Endpunkte zu beschreiben

Im folgenden `describe-client-vpn-endpoints` Beispiel werden Details zu all Ihren VPN Client-Endpunkten angezeigt.

```
aws ec2 describe-client-vpn-endpoints
```

Ausgabe:

```
{
  "ClientVpnEndpoints": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "Description": "Endpoint for Admin access",
      "Status": {
        "Code": "available"
      },
      "CreationTime": "2020-11-13T11:37:27",
      "DnsName": "*.cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com",
      "ClientCidrBlock": "172.31.0.0/16",
      "DnsServers": [
        "8.8.8.8"
      ],
      "SplitTunnel": false,
      "VpnProtocol": "openvpn",
      "TransportProtocol": "udp",
      "VpnPort": 443,
      "ServerCertificateArn": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "AuthenticationOptions": [
        {
          "Type": "certificate-authentication",
          "MutualAuthentication": {
            "ClientRootCertificateChain": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
          }
        }
      ],
      "ConnectionLogOptions": {
        "Enabled": true,
        "CloudwatchLogGroup": "Client-vpn-connection-logs",
        "CloudwatchLogStream": "cvpn-endpoint-123456789123abcde-ap-
south-1-2020/11/13-FCD8HEMVAccw"
      },
      "Tags": [
        {
          "Key": "Name",
          "Value": "Client VPN"
        }
      ],
    }
  ],
}
```

```

        "SecurityGroupIds": [
            "sg-aabbcc11223344567"
        ],
        "VpcId": "vpc-a87f92c1",
        "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/
endpoints/cvpn-endpoint-123456789123abcde",
        "ClientConnectOptions": {
            "Enabled": false
        }
    }
]
}

```

Weitere Informationen finden Sie unter [VPNClient-Endpunkte](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeClientVpnEndpoints](#) in der AWS CLI Befehlsreferenz.

describe-client-vpn-routes

Das folgende Codebeispiel zeigt die Verwendung `describe-client-vpn-routes`.

AWS CLI

Um die Routen für einen VPN Client-Endpunkt zu beschreiben

Im folgenden `describe-client-vpn-routes` Beispiel werden Details zu den Routen für den angegebenen VPN Client-Endpunkt angezeigt.

```

aws ec2 describe-client-vpn-routes \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Ausgabe:

```

{
  "Routes": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "DestinationCidr": "10.0.0.0/16",
      "TargetSubnet": "subnet-0123456789abcabca",
      "Type": "Nat",
      "Origin": "associate",
    }
  ]
}

```

```

        "Status": {
            "Code": "active"
        },
        "Description": "Default Route"
    },
    {
        "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
        "DestinationCidr": "0.0.0.0/0",
        "TargetSubnet": "subnet-0123456789abcabca",
        "Type": "Nat",
        "Origin": "add-route",
        "Status": {
            "Code": "active"
        }
    }
]
}

```

Weitere Informationen finden Sie im AWS VPNClient-Administratorhandbuch unter [Routen](#).

- API-Einheiten finden Sie [DescribeClientVpnRoutes](#) in der AWS CLI Befehlsreferenz.

describe-client-vpn-target-networks

Das folgende Codebeispiel zeigt die Verwendung `describe-client-vpn-target-networks`.

AWS CLI

Um die Zielnetzwerke für einen VPN Client-Endpunkt zu beschreiben

Im folgenden `describe-client-vpn-target-networks` Beispiel werden Details zu den Zielnetzwerken für den angegebenen VPN Client-Endpunkt angezeigt.

```
aws ec2 describe-client-vpn-target-networks \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Ausgabe:

```

{
  "ClientVpnTargetNetworks": [
    {
      "AssociationId": "cvpn-assoc-012e837060753dc3d",

```

```

    "VpcId": "vpc-1111122222233333",
    "TargetNetworkId": "subnet-0123456789abcabca",
    "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
    "Status": {
      "Code": "associating"
    },
    "SecurityGroups": [
      "sg-012345678910abcab"
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Zielnetzwerke](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeClientVpnTargetNetworks](#) in der AWS CLI Befehlsreferenz.

describe-coip-pools

Das folgende Codebeispiel zeigt die Verwendung `describe-coip-pools`.

AWS CLI

Um kundeneigene IP-Adresspools zu beschreiben

Im folgenden `describe-coip-pools` Beispiel werden die kundeneigenen IP-Adresspools in Ihrem AWS Konto beschrieben.

```
aws ec2 describe-coip-pools
```

Ausgabe:

```

{
  "CoipPools": [
    {
      "PoolId": "ipv4pool-coip-123a45678bEXAMPLE",
      "PoolCidrs": [
        "0.0.0.0/0"
      ],
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
      "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-coip-123a45678bEXAMPLE"
    }
  ]
}

```



```
]
}
```

Weitere Informationen finden Sie unter [Kundeneigene IP-Adressen](#) im AWS -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCoipPools](#) in der AWS CLI Befehlsreferenz.

describe-conversion-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-conversion-tasks`.

AWS CLI

Um den Status einer Konvertierungsaufgabe anzuzeigen

In diesem Beispiel wird der Status einer Konvertierungsaufgabe mit der ID `import-i-ffvko9js` zurückgegeben.

Befehl:

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

Ausgabe:

```
{
  "ConversionTasks": [
    {
      "ConversionTaskId": "import-i-ffvko9js",
      "ImportInstance": {
        "InstanceId": "i-1234567890abcdef0",
        "Volumes": [
          {
            "Volume": {
              "Id": "vol-049df61146c4d7901",
              "Size": 16
            },
            "Status": "completed",
            "Image": {
              "Size": 1300687360,
              "ImportManifestUrl": "https://s3.amazonaws.com/myimportbucket/411443cd-d620-4f1c-9d66-13144EXAMPLE/RHEL5.vmdkmanifest.xml?"
            }
          }
        ]
      }
    }
  ]
}
```

```

AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=140EXAMPLE&Signature=XYNhznHNgcqsjdXl9wRL
%2FJvEXAMPLE",
    "Format": "VMDK"
  },
  "BytesConverted": 1300682960,
  "AvailabilityZone": "us-east-1d"
}
]
},
"ExpirationTime": "2014-05-14T22:06:23Z",
"State": "completed"
}
]
}

```

- API-Einheiten finden Sie [DescribeConversionTasks](#) in der AWS CLI Befehlsreferenz.

describe-customer-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-customer-gateways`.

AWS CLI

Um Ihre Kunden-Gateways zu beschreiben

Dieses Beispiel beschreibt Ihre Kunden-Gateways.

Befehl:

```
aws ec2 describe-customer-gateways
```

Ausgabe:

```

{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-b4dc3961",
      "IpAddress": "203.0.113.12",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65000"
    }
  ],
}

```

```
{
  "CustomerGatewayId": "cgw-0e11f167",
  "IpAddress": "12.1.2.3",
  "State": "available",
  "Type": "ipsec.1",
  "BgpAsn": "65534"
}
]
```

Um ein bestimmtes Kunden-Gateway zu beschreiben

Dieses Beispiel beschreibt das angegebene Kunden-Gateway.

Befehl:

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

Ausgabe:

```
{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-0e11f167",
      "IpAddress": "12.1.2.3",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65534"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeCustomerGateways](#) in der AWS CLI Befehlsreferenz.

describe-dhcp-options

Das folgende Codebeispiel zeigt die Verwendung `describe-dhcp-options`.

AWS CLI

Beispiel 1: Um Ihre DHCP Optionen zu beschreiben

Im folgenden `describe-dhcp-options` Beispiel werden Details zu Ihren DHCP Optionen abgerufen.

```
aws ec2 describe-dhcp-options
```

Ausgabe:

```
{
  "DhcpOptions": [
    {
      "DhcpConfigurations": [
        {
          "Key": "domain-name",
          "Values": [
            {
              "Value": "us-east-2.compute.internal"
            }
          ]
        },
        {
          "Key": "domain-name-servers",
          "Values": [
            {
              "Value": "AmazonProvidedDNS"
            }
          ]
        }
      ],
      "DhcpOptionsId": "dopt-19edf471",
      "OwnerId": "111122223333"
    },
    {
      "DhcpConfigurations": [
        {
          "Key": "domain-name",
          "Values": [
            {
              "Value": "us-east-2.compute.internal"
            }
          ]
        },
        {
          "Key": "domain-name-servers",
```

```

        "Values": [
            {
                "Value": "AmazonProvidedDNS"
            }
        ]
    },
    ],
    "DhcpOptionsId": "dopt-fEXAMPLE",
    "OwnerId": "111122223333"
}
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit DHCP Optionssätzen](#) im AWS VPCBenutzerhandbuch.

Beispiel 2: Um Ihre DHCP Optionen zu beschreiben und die Ausgabe zu filtern

Das folgende `describe-dhcp-options` Beispiel beschreibt Ihre DHCP Optionen und verwendet einen Filter, um nur DHCP Optionen zurückzugeben, die `example.com` für den Domain-Namensserver verfügbar sind. Das Beispiel verwendet den `--query` Parameter, um nur die Konfigurationsinformationen und die ID in der Ausgabe anzuzeigen.

```

aws ec2 describe-dhcp-options \
  --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
  --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"

```

Ausgabe:

```

[
  [
    [
      {
        "Key": "domain-name",
        "Values": [
          {
            "Value": "example.com"
          }
        ]
      },
      {
        "Key": "domain-name-servers",

```

```
        "Values": [
          {
            "Value": "172.16.16.16"
          }
        ]
      },
    ],
    "dopt-001122334455667ab"
  ]
]
```

Weitere Informationen finden Sie unter [Arbeiten mit DHCP Optionssätzen](#) im AWS VPCBenutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeDhcpOptions](#) in der AWS CLI Befehlsreferenz.

describe-egress-only-internet-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-egress-only-internet-gateways`.

AWS CLI

Um Ihre Internet-Gateways nur für ausgehenden Datenverkehr zu beschreiben

In diesem Beispiel werden Ihre Internet-Gateways nur für ausgehenden Datenverkehr beschrieben.

Befehl:

```
aws ec2 describe-egress-only-internet-gateways
```

Ausgabe:

```
{
  "EgressOnlyInternetGateways": [
    {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [
        {
          "State": "attached",
          "VpcId": "vpc-0c62a468"
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

- API-Einheiten finden Sie in der Befehlsreferenz [DescribeEgressOnlyInternetGateways](#). AWS CLI

describe-elastic-gpus

Das folgende Codebeispiel zeigt die Verwendung `describe-elastic-gpus`.

AWS CLI

Um ein Elastic zu beschreiben GPU

Befehl:

```
aws ec2 describe-elastic-gpus --elastic-gpu-ids egpu-12345678901234567890abcdefghijkl
```

- API-Einheiten finden Sie [DescribeElasticGpus](#) in der AWS CLI Befehlsreferenz.

describe-export-image-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-export-image-tasks`.

AWS CLI

Um eine Aufgabe zum Exportieren von Bildern zu überwachen

Im folgenden `describe-export-image-tasks` Beispiel wird der Status der angegebenen Aufgabe zum Exportieren von Bildern überprüft. Die resultierende Bilddatei in Amazon S3 ist `my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk`.

```
aws ec2 describe-export-image-tasks \
  --export-image-task-ids export-ami-1234567890abcdef0
```

Ausgabe für eine Aufgabe zum Exportieren von Bildern, die gerade ausgeführt wird.

```
{
```

```
"ExportImageTasks": [  
  {  
    "ExportImageTaskId": "export-ami-1234567890abcdef0"  
    "Progress": "21",  
    "S3ExportLocation": {  
      "S3Bucket": "my-export-bucket",  
      "S3Prefix": "exports/"  
    },  
    "Status": "active",  
    "StatusMessage": "updating"  
  }  
]
```

Ausgabe für eine Aufgabe zum Exportieren von Bildern, die abgeschlossen ist.

```
{  
  "ExportImageTasks": [  
    {  
      "ExportImageTaskId": "export-ami-1234567890abcdef0"  
      "S3ExportLocation": {  
        "S3Bucket": "my-export-bucket",  
        "S3Prefix": "exports/"  
      },  
      "Status": "completed"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Exportieren einer VM aus einer AMI](#) im VM Import/Export User Guide.

- API-Einheiten finden Sie [DescribeExportImageTasks](#) in der AWS CLI Befehlsreferenz.

describe-export-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-export-tasks`.

AWS CLI

Um Details zu einer Instanzexportaufgabe aufzulisten

Dieses Beispiel beschreibt die Exportaufgabe mit der ID `export-i-fh 8sjjsq`.

Befehl:

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

Ausgabe:

```
{
  "ExportTasks": [
    {
      "State": "active",
      "InstanceExportDetails": {
        "InstanceId": "i-1234567890abcdef0",
        "TargetEnvironment": "vmware"
      },
      "ExportToS3Task": {
        "S3Bucket": "myexportbucket",
        "S3Key": "RHEL5export-i-fh8sjjsq.ova",
        "DiskImageFormat": "vmdk",
        "ContainerFormat": "ova"
      },
      "Description": "RHEL5 instance",
      "ExportTaskId": "export-i-fh8sjjsq"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeExportTasks](#) in AWS CLI der Befehlsreferenz.

describe-fast-launch-images

Das folgende Codebeispiel zeigt die Verwendung `describe-fast-launch-images`.

AWS CLI

Um die Details für Windows zu beschreiben AMIs, die für einen schnelleren Start konfiguriert sind

Das folgende `describe-fast-launch-images` Beispiel beschreibt die Details für alle AMIs in Ihrem Konto, die für einen schnelleren Start konfiguriert sind. Dazu gehören der Ressourcentyp, die Snapshot-Konfiguration, die Details der Startvorlage, die maximale Anzahl parallel Starts, die AMI Besitzer-ID, der Status der Schnellstartkonfiguration, der Grund für die Statusänderung und der Zeitpunkt der Statusänderung.

```
aws ec2 describe-fast-launch-images
```

Ausgabe:

```
{
  "FastLaunchImages": [
    {
      "ImageId": "ami-01234567890abcdef",
      "ResourceType": "snapshot",
      "SnapshotConfiguration": {},
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcdef",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
      },
      "MaxParallelLaunches": 6,
      "OwnerId": "0123456789123",
      "State": "enabled",
      "StateTransitionReason": "Client.UserInitiated",
      "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
    }
  ]
}
```

Weitere Informationen zur Konfiguration eines Windows AMI für einen schnelleren Start finden [Sie unter Konfiguration AMI für einen schnelleren Start](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeFastLaunchImages](#) in der AWS CLI Befehlsreferenz.

describe-fast-snapshot-restores

Das folgende Codebeispiel zeigt die Verwendung `describe-fast-snapshot-restores`.

AWS CLI

Um schnelle Snapshot-Wiederherstellungen zu beschreiben

Im folgenden `describe-fast-snapshot-restores` Beispiel werden Details für alle schnellen Snapshot-Wiederherstellungen mit dem Status `disabled` angezeigt.

```
aws ec2 describe-fast-snapshot-restores \
```

```
--filters Name=state,Values=disabled
```

Ausgabe:

```
{
  "FastSnapshotRestores": [
    {
      "SnapshotId": "snap-1234567890abcdef0",
      "AvailabilityZone": "us-west-2c",
      "State": "disabled",
      "StateTransitionReason": "Client.UserInitiated - Lifecycle state
transition",
      "OwnerId": "123456789012",
      "EnablingTime": "2020-01-25T23:57:49.596Z",
      "OptimizingTime": "2020-01-25T23:58:25.573Z",
      "EnabledTime": "2020-01-25T23:59:29.852Z",
      "DisablingTime": "2020-01-26T00:40:56.069Z",
      "DisabledTime": "2020-01-26T00:41:27.390Z"
    }
  ]
}
```

Das folgende `describe-fast-snapshot-restores` Beispiel beschreibt alle schnellen Snapshot-Wiederherstellungen.

```
aws ec2 describe-fast-snapshot-restores
```

- API-Einheiten finden Sie [DescribeFastSnapshotRestores](#) in der AWS CLI Befehlsreferenz.

describe-fleet-history

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet-history`.

AWS CLI

Um die Geschichte der EC2 Flotte zu beschreiben

Das folgende `describe-fleet-history` Beispiel gibt den Verlauf für die angegebene EC2 Flotte ab dem angegebenen Zeitpunkt zurück. Die Ausgabe bezieht sich auf eine EC2 Flotte mit zwei laufenden Instances.

```
aws ec2 describe-fleet-history \  
--fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
--start-time 2020-09-01T00:00:00Z
```

Ausgabe:

```
{  
  "HistoryRecords": [  
    {  
      "EventInformation": {  
        "EventSubType": "submitted"  
      },  
      "EventType": "fleetRequestChange",  
      "Timestamp": "2020-09-01T18:26:05.000Z"  
    },  
    {  
      "EventInformation": {  
        "EventSubType": "active"  
      },  
      "EventType": "fleetRequestChange",  
      "Timestamp": "2020-09-01T18:26:15.000Z"  
    },  
    {  
      "EventInformation": {  
        "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",  
        "EventSubType": "progress"  
      },  
      "EventType": "fleetRequestChange",  
      "Timestamp": "2020-09-01T18:26:17.000Z"  
    },  
    {  
      "EventInformation": {  
        "EventDescription": "{\"instanceType\":\"t2.small\", ...}",  
        "EventSubType": "launched",  
        "InstanceId": "i-083a1c446e66085d2"  
      },  
      "EventType": "instanceChange",  
      "Timestamp": "2020-09-01T18:26:17.000Z"  
    },  
    {  
      "EventInformation": {  
        "EventDescription": "{\"instanceType\":\"t2.small\", ...}",  
        "EventSubType": "launched",  
      },  
      "EventType": "instanceChange",  
      "Timestamp": "2020-09-01T18:26:17.000Z"  
    }  
  ]  
}
```

```

        "InstanceId": "i-090db02406cc3c2d6"
      },
      "EventType": "instanceChange",
      "Timestamp": "2020-09-01T18:26:17.000Z"
    }
  ],
  "LastEvaluatedTime": "2020-09-01T19:10:19.000Z",
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "StartTime": "2020-08-31T23:53:20.000Z"
}

```

Weitere Informationen finden Sie unter [Verwaltung einer EC2 Flotte](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeFleetHistory](#) in der AWS CLI Befehlsreferenz.

describe-fleet-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet-instances`.

AWS CLI

Um die laufenden Instances für eine EC2 Flotte zu beschreiben

Das folgende `describe-fleet-instances` Beispiel beschreibt die laufenden Instances für die angegebene EC2 Flotte.

```

aws ec2 describe-fleet-instances \
  --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE

```

Ausgabe:

```

{
  "ActiveInstances": [
    {
      "InstanceId": "i-090db02406cc3c2d6",
      "InstanceType": "t2.small",
      "SpotInstanceRequestId": "sir-a43gtpfk",
      "InstanceHealth": "healthy"
    },
    {
      "InstanceId": "i-083a1c446e66085d2",

```

```

        "InstanceType": "t2.small",
        "SpotInstanceRequestId": "sir-iwcit2nj",
        "InstanceHealth": "healthy"
    }
],
"FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}

```

Weitere Informationen finden Sie unter [Verwaltung einer EC2 Flotte](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeFleetInstances](#) in der AWS CLI Befehlsreferenz.

describe-fleets

Das folgende Codebeispiel zeigt die Verwendung `describe-fleets`.

AWS CLI

Um eine EC2 Flotte zu beschreiben

Das folgende `describe-fleets` Beispiel beschreibt die angegebene EC2 Flotte.

```

aws ec2 describe-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE

```

Ausgabe:

```

{
  "Fleets": [
    {
      "ActivityStatus": "pending_fulfillment",
      "CreateTime": "2020-09-01T18:26:05.000Z",
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
      "FleetState": "active",
      "ExcessCapacityTerminationPolicy": "termination",
      "FulfilledCapacity": 0.0,
      "FulfilledOnDemandCapacity": 0.0,
      "LaunchTemplateConfigs": [
        {
          "LaunchTemplateSpecification": {
            "LaunchTemplateId": "lt-0e632f2855a979cd5",

```

```
        "Version": "1"
      }
    ],
    "TargetCapacitySpecification": {
      "TotalTargetCapacity": 2,
      "OnDemandTargetCapacity": 0,
      "SpotTargetCapacity": 2,
      "DefaultTargetCapacityType": "spot"
    },
    "TerminateInstancesWithExpiration": false,
    "Type": "maintain",
    "ReplaceUnhealthyInstances": false,
    "SpotOptions": {
      "AllocationStrategy": "lowestPrice",
      "InstanceInterruptionBehavior": "terminate",
      "InstancePoolsToUseCount": 1
    },
    "OnDemandOptions": {
      "AllocationStrategy": "lowestPrice"
    }
  }
]
}
```

Weitere Informationen finden Sie unter [Verwaltung einer EC2 Flotte](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeFleets](#) in der AWS CLI Befehlsreferenz.

describe-flow-logs

Das folgende Codebeispiel zeigt die Verwendung `describe-flow-logs`.

AWS CLI

Beispiel 1: Um all Ihre Flow-Logs zu beschreiben

Im folgenden `describe-flow-logs` Beispiel werden Details für alle Ihre Flow-Logs angezeigt.

```
aws ec2 describe-flow-logs
```

Ausgabe:

```

{
  "FlowLogs": [
    {
      "CreationTime": "2018-02-21T13:22:12.644Z",
      "DeliverLogsPermissionArn": "arn:aws:iam::123456789012:role/flow-logs-
role",
      "DeliverLogsStatus": "SUCCESS",
      "FlowLogId": "fl-aabbccdd112233445",
      "MaxAggregationInterval": 600,
      "FlowLogStatus": "ACTIVE",
      "LogGroupName": "FlowLogGroup",
      "ResourceId": "subnet-12345678901234567",
      "TrafficType": "ALL",
      "LogDestinationType": "cloud-watch-logs",
      "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr}
${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end}
${action} ${log-status}"
    },
    {
      "CreationTime": "2020-02-04T15:22:29.986Z",
      "DeliverLogsStatus": "SUCCESS",
      "FlowLogId": "fl-01234567890123456",
      "MaxAggregationInterval": 60,
      "FlowLogStatus": "ACTIVE",
      "ResourceId": "vpc-00112233445566778",
      "TrafficType": "ACCEPT",
      "LogDestinationType": "s3",
      "LogDestination": "arn:aws:s3:::my-flow-log-bucket/custom",
      "LogFormat": "${version} ${vpc-id} ${subnet-id} ${instance-id}
${interface-id} ${account-id} ${type} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
${pkt-srcaddr} ${pkt-dstaddr} ${protocol} ${bytes} ${packets} ${start} ${end}
${action} ${tcp-flags} ${log-status}"
    }
  ]
}

```

Beispiel 2: Um eine Teilmenge Ihrer Flow-Logs zu beschreiben

Das folgende `describe-flow-logs` Beispiel verwendet einen Filter, um Details nur für die Flow-Logs anzuzeigen, die sich in der angegebenen Protokollgruppe in Amazon CloudWatch Logs befinden.

```
aws ec2 describe-flow-logs \
```



```
--filter "Name=log-group-name,Values=MyFlowLogs"
```

- API-Einheiten finden Sie [DescribeFlowLogs](#) in der AWS CLI Befehlsreferenz.

describe-fpga-image-attribute

Das folgende Codebeispiel zeigt die Verwendung `describe-fpga-image-attribute`.

AWS CLI

Um die Eigenschaften eines FPGA Amazon-Bildes zu beschreiben

In diesem Beispiel werden die Ladeberechtigungen für das angegebene Objekt beschrieben.

Befehl:

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --  
attribute LoadPermission
```

Ausgabe:

```
{  
  "FpgaImageAttribute": {  
    "FpgaImageId": "afi-0d123e123bfc85abc",  
    "LoadPermissions": [  
      {  
        "UserId": "123456789012"  
      }  
    ]  
  }  
}
```

- API-Einheiten finden Sie [DescribeFpgaImageAttribute](#) in der AWS CLI Befehlsreferenz.

describe-fpga-images

Das folgende Codebeispiel zeigt die Verwendung `describe-fpga-images`.

AWS CLI

Um FPGA Amazon-Bilder zu beschreiben

In diesem Beispiel wird beschriebenAFIs, welche dem Konto gehören123456789012.

Befehl:

```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

Ausgabe:

```
{
  "FpgaImages": [
    {
      "UpdateTime": "2017-12-22T12:09:14.000Z",
      "Name": "my-afi",
      "PciId": {
        "SubsystemVendorId": "0xfedd",
        "VendorId": "0x1d0f",
        "DeviceId": "0xf000",
        "SubsystemId": "0x1d51"
      },
      "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc",
      "Public": false,
      "State": {
        "Code": "available"
      },
      "ShellVersion": "0x071417d3",
      "OwnerId": "123456789012",
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "CreateTime": "2017-12-22T11:43:33.000Z",
      "Description": "my-afi"
    }
  ]
}
```

- APIEinzelheiten finden Sie [DescribeFpgaImages](#)in der AWS CLI Befehlsreferenz.

describe-host-reservation-offerings

Das folgende Codebeispiel zeigt die Verwendungdescribe-host-reservation-offerings.

AWS CLI

Zur Beschreibung von Reservierungsangeboten für dedizierte Hosts

In diesem Beispiel werden die Dedicated Host-Reservierungen für die M4-Instance-Familie beschrieben, die käuflich erworben werden können.

Befehl:

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

Ausgabe:

```
{
  "OfferingSet": [
    {
      "HourlyPrice": "1.499",
      "OfferingId": "hro-03f707bf363b6b324",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    },
    {
      "HourlyPrice": "1.045",
      "OfferingId": "hro-0ef9181cabdef7a02",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "UpfrontPrice": "0.000",
      "Duration": 94608000
    },
    {
      "HourlyPrice": "0.714",
      "OfferingId": "hro-04567a15500b92a51",
      "InstanceFamily": "m4",
      "PaymentOption": "PartialUpfront",
      "UpfrontPrice": "6254.000",
      "Duration": 31536000
    },
    {
      "HourlyPrice": "0.484",
      "OfferingId": "hro-0d5d7a9d23ed7fbfe",
      "InstanceFamily": "m4",
      "PaymentOption": "PartialUpfront",
      "UpfrontPrice": "12720.000",
      "Duration": 94608000
    }
  ],
}
```

```
{
  "HourlyPrice": "0.000",
  "OfferingId": "hro-05da4108ca998c2e5",
  "InstanceFamily": "m4",
  "PaymentOption": "AllUpfront",
  "UpfrontPrice": "23913.000",
  "Duration": 94608000
},
{
  "HourlyPrice": "0.000",
  "OfferingId": "hro-0a9f9be3b95a3dc8f",
  "InstanceFamily": "m4",
  "PaymentOption": "AllUpfront",
  "UpfrontPrice": "12257.000",
  "Duration": 31536000
}
]
```

- API-Einheiten finden Sie unter [DescribeHostReservationOfferings AWS CLI Befehlsreferenz](#).

describe-host-reservations

Das folgende Codebeispiel zeigt die Verwendung `describe-host-reservations`.

AWS CLI

Um Reservierungen für Dedicated Hosts in deinem Konto zu beschreiben

In diesem Beispiel werden die Reservierungen für Dedicated Hosts in deinem Konto beschrieben.

Befehl:

```
aws ec2 describe-host-reservations
```

Ausgabe:

```
{
  "HostReservationSet": [
    {
      "Count": 1,
      "End": "2019-01-10T12:14:09Z",
```

```

    "HourlyPrice": "1.499",
    "InstanceFamily": "m4",
    "OfferingId": "hro-03f707bf363b6b324",
    "PaymentOption": "NoUpfront",
    "State": "active",
    "HostIdSet": [
        "h-013abcd2a00cbd123"
    ],
    "Start": "2018-01-10T12:14:09Z",
    "HostReservationId": "hr-0d418a3a4ffc669ae",
    "UpfrontPrice": "0.000",
    "Duration": 31536000
  }
]
}

```

- API-Einheiten finden Sie [DescribeHostReservations](#) in der AWS CLI Befehlsreferenz.

describe-hosts

Das folgende Codebeispiel zeigt die Verwendung `describe-hosts`.

AWS CLI

Um Details zu Dedicated Hosts anzuzeigen

Im folgenden `describe-hosts` Beispiel werden Details zu den available Dedicated Hosts in Ihrem AWS Konto angezeigt.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

Ausgabe:

```

{
  "Hosts": [
    {
      "HostId": "h-07879acf49EXAMPLE",
      "Tags": [
        {
          "Value": "production",
          "Key": "purpose"
        }
      ]
    }
  ]
}

```

```
    }
  ],
  "HostProperties": {
    "Cores": 48,
    "TotalVCpus": 96,
    "InstanceType": "m5.large",
    "Sockets": 2
  },
  "Instances": [],
  "State": "available",
  "AvailabilityZone": "eu-west-1a",
  "AvailableCapacity": {
    "AvailableInstanceCapacity": [
      {
        "AvailableCapacity": 48,
        "InstanceType": "m5.large",
        "TotalCapacity": 48
      }
    ],
    "AvailableVCpus": 96
  },
  "HostRecovery": "on",
  "AllocationTime": "2019-08-19T08:57:44.000Z",
  "AutoPlacement": "off"
}
]
```

Weitere Informationen finden Sie unter [Viewing Dedicated Hosts](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeHosts](#) in der AWS CLI Befehlsreferenz.

describe-iam-instance-profile-associations

Das folgende Codebeispiel zeigt die Verwendung `describe-iam-instance-profile-associations`.

AWS CLI

Um die Zuordnungen von IAM Instanzprofilen zu beschreiben

In diesem Beispiel werden alle Ihre IAM Instanzprofilzuordnungen beschrieben.

Befehl:

```
aws ec2 describe-iam-instance-profile-associations
```

Ausgabe:

```
{
  "IamInstanceProfileAssociations": [
    {
      "InstanceId": "i-09eb09efa73ec1dee",
      "State": "associated",
      "AssociationId": "iip-assoc-0db249b1f25fa24b8",
      "IamInstanceProfile": {
        "Id": "AIPAJVQN4F5WVLGCJDRGM",
        "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
    },
    {
      "InstanceId": "i-0402909a2f4dff14",
      "State": "associating",
      "AssociationId": "iip-assoc-0d1ec06278d29f44a",
      "IamInstanceProfile": {
        "Id": "AGJAJVQN4F5WVLGCJABCM",
        "Arn": "arn:aws:iam::123456789012:instance-profile/user1-role"
      }
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeIamInstanceProfileAssociations](#) in der AWS CLI Befehlsreferenz.

describe-id-format

Das folgende Codebeispiel zeigt die Verwendung `describe-id-format`.

AWS CLI

Beispiel 1: Um das ID-Format einer Ressource zu beschreiben

Das folgende `describe-id-format` Beispiel beschreibt das ID-Format für Sicherheitsgruppen.

```
aws ec2 describe-id-format \  
  --resource security-group
```

In der folgenden Beispielausgabe gibt der `Deadline` Wert an, dass die Frist für den dauerhaften Wechsel dieses Ressourcentyps vom kurzen ID-Format zum langen ID-Format am 15. August 2018 um 00:00 Uhr UTC abgelaufen ist.

```
{  
  "Statuses": [  
    {  
      "Deadline": "2018-08-15T00:00:00.000Z",  
      "Resource": "security-group",  
      "UseLongIds": true  
    }  
  ]  
}
```

Beispiel 2: Um das ID-Format für alle Ressourcen zu beschreiben

Das folgende `describe-id-format` Beispiel beschreibt das ID-Format für alle Ressourcentypen. Alle Ressourcentypen, die das kurze ID-Format unterstützten, wurden auf das lange ID-Format umgestellt.

```
aws ec2 describe-id-format
```

- API-Einheiten finden Sie [Beschreibung des ID-Formats](#) in der AWS CLI Befehlsreferenz.

describe-identity-id-format

Das folgende Codebeispiel zeigt die Verwendung `describe-identity-id-format`.

AWS CLI

Um das ID-Format für eine IAM Rolle zu beschreiben

Das folgende `describe-identity-id-format` Beispiel beschreibt das ID-Format, das von Instanzen empfangen wird, die von der IAM Rolle `EC2Role` in Ihrem AWS Konto erstellt wurden.

```
aws ec2 describe-identity-id-format \  
  --resource iam-role
```



```
--principal-arn arn:aws:iam::123456789012:role/my-iam-role \  
--resource instance
```

Die folgende Ausgabe zeigt, dass Instanzen, die von dieser Rolle erstellt wurden, IDs im Long-ID-Format empfangen werden.

```
{  
  "Statuses": [  
    {  
      "Deadline": "2016-12-15T00:00:00Z",  
      "Resource": "instance",  
      "UseLongIds": true  
    }  
  ]  
}
```

Um das ID-Format für einen IAM Benutzer zu beschreiben

Im folgenden `describe-identity-id-format` Beispiel wird das ID-Format beschrieben, das von Snapshots empfangen wird, die vom IAM Benutzer `AdminUser` in Ihrem AWS Konto erstellt wurden.

```
aws ec2 describe-identity-id-format \  
--principal-arn arn:aws:iam::123456789012:user/AdminUser \  
--resource snapshot
```

Aus der Ausgabe geht hervor, dass von diesem Benutzer erstellte Schnappschüsse IDs im Long-ID-Format empfangen werden.

```
{  
  "Statuses": [  
    {  
      "Deadline": "2016-12-15T00:00:00Z",  
      "Resource": "snapshot",  
      "UseLongIds": true  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribeIdentityIdFormat](#) in der AWS CLI Befehlsreferenz.

describe-image-attribute

Das folgende Codebeispiel zeigt die Verwendung `describe-image-attribute`.

AWS CLI

Um die Startberechtigungen für einen zu beschreiben AMI

In diesem Beispiel werden die Startberechtigungen für die angegebene Datei beschrieben AMI.

Befehl:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --  
attribute LaunchPermission
```

Ausgabe:

```
{  
  "LaunchPermissions": [  
    {  
      "UserId": "123456789012"  
    }  
  ],  
  "ImageId": "ami-5731123e",  
}
```

Um die Produktcodes für eine zu beschreiben AMI

In diesem Beispiel werden die Produktcodes für die angegebenen Produkte beschrieben AMI.

Beachten Sie, dass AMI es hier keine Produktcodes gibt.

Befehl:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

Ausgabe:

```
{  
  "ProductCodes": [],  
  "ImageId": "ami-5731123e",  
}
```

- API-Einheiten finden Sie [DescribeImageAttribute](#) in der AWS CLI Befehlsreferenz.

describe-images

Das folgende Codebeispiel zeigt die Verwendung `describe-images`.

AWS CLI

Beispiel 1: Um ein zu beschreiben AMI

Das folgende `describe-images` Beispiel beschreibt die AMI in der angegebenen Region angegebene.

```
aws ec2 describe-images \  
  --region us-east-1 \  
  --image-ids ami-1234567890EXAMPLE
```

Ausgabe:

```
{  
  "Images": [  
    {  
      "VirtualizationType": "hvm",  
      "Description": "Provided by Red Hat, Inc.",  
      "PlatformDetails": "Red Hat Enterprise Linux",  
      "EnaSupport": true,  
      "Hypervisor": "xen",  
      "State": "available",  
      "SriovNetSupport": "simple",  
      "ImageId": "ami-1234567890EXAMPLE",  
      "UsageOperation": "RunInstances:0010",  
      "BlockDeviceMappings": [  
        {  
          "DeviceName": "/dev/sda1",  
          "Ebs": {  
            "SnapshotId": "snap-111222333444aaabb",  
            "DeleteOnTermination": true,  
            "VolumeType": "gp2",  
            "VolumeSize": 10,  
            "Encrypted": false  
          }  
        }  
      ]  
    }  
  ]  
}
```

```

    ],
    "Architecture": "x86_64",
    "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-
GP2",
    "RootDeviceType": "ebs",
    "OwnerId": "123456789012",
    "RootDeviceName": "/dev/sda1",
    "CreationDate": "2019-05-10T13:17:12.000Z",
    "Public": true,
    "ImageType": "machine",
    "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
  }
]
}

```

Weitere Informationen finden Sie unter [Amazon Machine Images \(AMI\)](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Zur Beschreibung AMIs anhand von Filtern

Das folgende `describe-images` Beispiel beschreibt Windows, das von Amazon AMIs bereitgestellt und von Amazon unterstützt wirdEBS.

```

aws ec2 describe-images \
  --owners amazon \
  --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"

```

Ein Beispiel für die Ausgabe von `describe-images` finden Sie in Beispiel 1.

Weitere Beispiele für die Verwendung von Filtern finden Sie unter [Ressourcen auflisten und filtern](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 3: Zur Beschreibung AMIs anhand von Stichwörtern

Das folgende `describe-images` Beispiel beschreibt alleAMIs, die das Tag `habenType=Custom`. In dem Beispiel wird der `--query` Parameter verwendet, um nur die anzuzeigen AMIIDs.

```

aws ec2 describe-images \
  --filters "Name=tag:Type,Values=Custom" \
  --query 'Images[*].[ImageId]' \

```

```
--output text
```

Ausgabe:

```
ami-1234567890EXAMPLE  
ami-0abcdef1234567890
```

Weitere Beispiele für die Verwendung von Tagfiltern finden Sie unter [Arbeiten mit Tags](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeImages](#) in der AWS CLI Befehlsreferenz.

describe-import-image-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-import-image-tasks`.

AWS CLI

Um eine Aufgabe zum Importieren von Bildern zu überwachen

Im folgenden `describe-import-image-tasks` Beispiel wird der Status der angegebenen Aufgabe zum Importieren von Bildern überprüft.

```
aws ec2 describe-import-image-tasks \  
  --import-task-ids import-ami-1234567890abcdef0
```

Ausgabe für eine Aufgabe zum Importieren von Bildern, die gerade ausgeführt wird.

```
{  
  "ImportImageTasks": [  
    {  
      "ImportTaskId": "import-ami-1234567890abcdef0",  
      "Progress": "28",  
      "SnapshotDetails": [  
        {  
          "DiskImageSize": 705638400.0,  
          "Format": "ova",  
          "Status": "completed",  
          "UserBucket": {  
            "S3Bucket": "my-import-bucket",  
            "S3Key": "vms/my-server-vm.ova"  
          }  
        }  
      ]  
    }  
  ]  
}
```

```

        }
      ],
      "Status": "active",
      "StatusMessage": "converting"
    }
  ]
}

```

Ausgabe für eine Aufgabe zum Importieren von Bildern, die abgeschlossen ist. Die ID des Ergebnisses AMI wird von bereitgestelltImageId.

```

{
  "ImportImageTasks": [
    {
      "ImportTaskId": "import-ami-1234567890abcdef0",
      "ImageId": "ami-1234567890abcdef0",
      "SnapshotDetails": [
        {
          "DiskImageSize": 705638400.0,
          "Format": "ova",
          "SnapshotId": "snap-1234567890abcdef0",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.ova"
          }
        }
      ],
      "Status": "completed"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeImportImageTasks](#) in der AWS CLI Befehlsreferenz.

describe-import-snapshot-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-import-snapshot-tasks`.

AWS CLI

Um eine Aufgabe zum Importieren eines Snapshots zu überwachen

Im folgenden `describe-import-snapshot-tasks` Beispiel wird der Status der angegebenen Import-Snapshot-Aufgabe überprüft.

```
aws ec2 describe-import-snapshot-tasks \
  --import-task-ids import-snap-1234567890abcdef0
```

Ausgabe für eine laufende Snapshot-Importaufgabe:

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "My server VMDK",
      "ImportTaskId": "import-snap-1234567890abcdef0",
      "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "705638400.0",
        "Format": "VMDK",
        "Progress": "42",
        "Status": "active",
        "StatusMessage": "downloading/convertng",
        "UserBucket": {
          "S3Bucket": "my-import-bucket",
          "S3Key": "vms/my-server-vm.vmdk"
        }
      }
    }
  ]
}
```

Ausgabe für eine abgeschlossene Import-Snapshot-Aufgabe. Die ID des resultierenden Snapshots wird von `bereitgestelltSnapshotId` bereitgestellt.

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "My server VMDK",
      "ImportTaskId": "import-snap-1234567890abcdef0",
      "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "705638400.0",
        "Format": "VMDK",
        "SnapshotId": "snap-1234567890abcdef0"
      }
    }
  ]
}
```

```

        "Status": "completed",
        "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.vmdk"
        }
    }
}
]
}

```

- API-Einheiten finden Sie [DescribeImportSnapshotTasks](#) in der AWS CLI Befehlsreferenz.

describe-instance-attribute

Das folgende Codebeispiel zeigt die Verwendung von `describe-instance-attribute`.

AWS CLI

Um den Instanztyp zu beschreiben

Dieses Beispiel beschreibt den Instanztyp der angegebenen Instanz.

Befehl:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute instanceType
```

Ausgabe:

```
{
  "InstanceId": "i-1234567890abcdef0"
  "InstanceType": {
    "Value": "t1.micro"
  }
}
```

Um das `disableApiTermination` Attribut zu beschreiben

Dieses Beispiel beschreibt das `disableApiTermination` Attribut der angegebenen Instanz.

Befehl:


```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute disableApiTermination
```

Ausgabe:

```
{  
  "InstanceId": "i-1234567890abcdef0"  
    "DisableApiTermination": {  
      "Value": "false"  
    }  
}
```

Um die Blockgerätezuweisung für eine Instanz zu beschreiben

Dieses Beispiel beschreibt das `blockDeviceMapping` Attribut der angegebenen Instanz.

Befehl:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute blockDeviceMapping
```

Ausgabe:

```
{  
  "InstanceId": "i-1234567890abcdef0"  
  "BlockDeviceMappings": [  
    {  
      "DeviceName": "/dev/sda1",  
      "Ebs": {  
        "Status": "attached",  
        "DeleteOnTermination": true,  
        "VolumeId": "vol-049df61146c4d7901",  
        "AttachTime": "2013-05-17T22:42:34.000Z"  
      }  
    },  
    {  
      "DeviceName": "/dev/sdf",  
      "Ebs": {  
        "Status": "attached",  
        "DeleteOnTermination": false,  
        "VolumeId": "vol-049df61146c4d7901",  
        "AttachTime": "2013-09-10T23:07:00.000Z"  
      }  
    }  
  ]  
}
```

```

    }
  ],
}

```

- API-Einheiten finden Sie [DescribeInstanceAttribute](#) in der AWS CLI Befehlsreferenz.

describe-instance-connect-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-connect-endpoints`.

AWS CLI

Um einen EC2 Instance Connect-Endpoint zu beschreiben

Das folgende `describe-instance-connect-endpoints` Beispiel beschreibt den angegebenen EC2 Instance Connect-Endpoint.

```

aws ec2 describe-instance-connect-endpoints \
  --region us-east-1 \
  --instance-connect-endpoint-ids eice-0123456789example

```

Ausgabe:

```

{
  "InstanceConnectEndpoints": [
    {
      "OwnerId": "111111111111",
      "InstanceConnectEndpointId": "eice-0123456789example",
      "InstanceConnectEndpointArn": "arn:aws:ec2:us-
east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
      "State": "create-complete",
      "StateMessage": "",
      "DnsName": "eice-0123456789example.b67b86ba.ec2-instance-connect-
endpoint.us-east-1.amazonaws.com",
      "NetworkInterfaceIds": [
        "eni-0123456789example"
      ],
      "VpcId": "vpc-0123abcd",
      "AvailabilityZone": "us-east-1d",
      "CreatedAt": "2023-02-07T12:05:37+00:00",
      "SubnetId": "subnet-0123abcd",
    }
  ]
}

```

```
    "Tags": []
  }
]
}
```

Weitere Informationen finden Sie unter [Erstellen eines EC2 Instance Connect-Endpunkts](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeInstanceConnectEndpoints](#) in der AWS CLI Befehlsreferenz.

describe-instance-credit-specifications

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-credit-specifications`.

AWS CLI

Um die Kreditoption für die CPU Nutzung einer oder mehrerer Instanzen zu beschreiben

Das folgende `describe-instance-credit-specifications` Beispiel beschreibt die CPU Kreditoption für die angegebene Instanz.

```
aws ec2 describe-instance-credit-specifications \
  --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{
  "InstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CpuCredits": "unlimited"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Burstable-Performance-Instances](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeInstanceCreditSpecifications](#) in der AWS CLI Befehlsreferenz.

describe-instance-event-notification-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-event-notification-attributes`.

AWS CLI

Um die Tags für Benachrichtigungen über geplante Ereignisse zu beschreiben

Im folgenden `describe-instance-event-notification-attributes` Beispiel werden die Tags beschrieben, die in Benachrichtigungen über geplante Ereignisse erscheinen sollen.

```
aws ec2 describe-instance-event-notification-attributes
```

Ausgabe:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [],
    "IncludeAllTagsOfInstance": true
  }
}
```

Weitere Informationen finden Sie unter [Geplante Ereignisse für Ihre Instances](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeInstanceEventNotificationAttributes](#) in der AWS CLI Befehlsreferenz.

describe-instance-event-windows

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-event-windows`.

AWS CLI

Beispiel 1: Um alle Eventfenster zu beschreiben

Das folgende `describe-instance-event-windows` Beispiel beschreibt alle Ereignisfenster in der angegebenen Region.

```
aws ec2 describe-instance-event-windows \
  --region us-east-1
```

Ausgabe:

```
{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "AssociationTarget": {
        "InstanceIds": [
          "i-1234567890abcdef0",
          "i-0598c7d356eba48d7"
        ],
        "Tags": [],
        "DedicatedHostIds": []
      },
      "State": "active",
      "Tags": []
    }
    ...
  ],
  "NextToken": "9d624e0c-388b-4862-a31e-a85c64fc1d4a"
}
```

Beispiel 2: Um ein bestimmtes Ereignisfenster zu beschreiben

Das folgende `describe-instance-event-windows` Beispiel beschreibt ein bestimmtes Ereignis, indem der `instance-event-window` Parameter verwendet wird, um ein bestimmtes Ereignisfenster zu beschreiben.

```
aws ec2 describe-instance-event-windows \
  --region us-east-1 \
  --instance-event-window-ids iew-0abcdef1234567890
```

Ausgabe:

```
{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
```

```

    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-0598c7d356eba48d7"
      ],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "active",
    "Tags": []
  }
}

```

Beispiel 3: Um Ereignisfenster zu beschreiben, die einem oder mehreren Filtern entsprechen

Im folgenden `describe-instance-event-windows` Beispiel werden mithilfe des `filter` Parameters Ereignisfenster beschrieben, die einem oder mehreren Filtern entsprechen. Der `instance-id` Filter wird verwendet, um alle Ereignisfenster zu beschreiben, die der angegebenen Instanz zugeordnet sind. Wenn ein Filter verwendet wird, führt er eine direkte Übereinstimmung durch. Der `instance-id`-Filter ist jedoch anders. Wenn es keine direkte Übereinstimmung mit der Instanz-ID gibt, wird auf indirekte Verknüpfungen mit dem Ereignisfenster zurückgegriffen, z. B. auf die Tags der Instanz oder die Dedicated Host-ID (wenn es sich bei der Instance um einen Dedicated Host handelt).

```

aws ec2 describe-instance-event-windows \
  --region us-east-1 \
  --filters Name=instance-id,Values=i-1234567890abcdef0 \
  --max-results 100 \
  --next-token <next-token-value>

```

Ausgabe:

```

{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0dbc0adb66f235982",
      "TimeRanges": [
        {
          "StartWeekDay": "sunday",

```

```

        "StartHour": 2,
        "EndWeekDay": "sunday",
        "EndHour": 8
      }
    ],
    "Name": "myEventWindowName",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": [
        "h-0140d9a7ecbd102dd"
      ]
    },
    "State": "active",
    "Tags": []
  }
]
}

```

In der Beispielausgabe befindet sich die Instance auf einem Dedicated Host, der dem Ereignisfenster zugeordnet ist.

Informationen zu Einschränkungen im Veranstaltungsfenster finden Sie unter [Überlegungen](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeInstanceEventWindows](#) in der AWS CLI Befehlsreferenz.

describe-instance-status

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-status`.

AWS CLI

So beschreiben Sie den Status einer Instance

Das folgende `describe-instance-status`-Beispiel beschreibt den aktuellen Status der angegebenen Instance.

```
aws ec2 describe-instance-status \
  --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{
  "InstanceStatuses": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "InstanceState": {
        "Code": 16,
        "Name": "running"
      },
      "AvailabilityZone": "us-east-1d",
      "SystemStatus": {
        "Status": "ok",
        "Details": [
          {
            "Status": "passed",
            "Name": "reachability"
          }
        ]
      },
      "InstanceStatus": {
        "Status": "ok",
        "Details": [
          {
            "Status": "passed",
            "Name": "reachability"
          }
        ]
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Überwachen des Status Ihrer Instances](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstanceStatus](#) in der AWS CLI Befehlsreferenz.

describe-instance-topology

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-topology`.

AWS CLI

Um die Instanztopologie all Ihrer Instances zu beschreiben

Das folgende `describe-instance-topology` Beispiel beschreibt die Topologie all Ihrer Instances, die den unterstützten Instance-Typen für diesen Befehl entsprechen.

```
aws ec2 describe-instance-topology \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "my-ml-cpg",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "p4d.24xlarge",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-3333333333example",  
      "InstanceType": "trn1.32xlarge",  
      "NetworkNodes": [  
        "nn-1212121212example",  
        "nn-1211122211example",  
        "nn-1311133311example"  
      ],  
      "ZoneId": "usw2-az4",  
      "AvailabilityZone": "us-west-2d"  
    }  
  ]  
}
```

```
    },
    {
      "InstanceId": "i-444444444example",
      "InstanceType": "trn1.2xlarge",
      "NetworkNodes": [
        "nn-111111111example",
        "nn-5434334334example",
        "nn-1235301234example"
      ],
      "ZoneId": "usw2-az2",
      "AvailabilityZone": "us-west-2a"
    }
  ],
  "NextToken": "SomeEncryptedToken"
}
```

Weitere Informationen, einschließlich weiterer Beispiele, finden Sie unter [EC2Amazon-Instance-Topologie](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstanceTopology](#) in der AWS CLI Befehlsreferenz.

describe-instance-type-offerings

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-type-offerings`.

AWS CLI

Beispiel 1: Um die in einer Region angebotenen Instance-Typen aufzulisten

Das folgende `describe-instance-type-offerings` Beispiel listet die Instanztypen auf, die in der Region angeboten werden, die als Standardregion für die konfiguriert ist AWS CLI.

```
aws ec2 describe-instance-type-offerings
```

Um die Instanztypen aufzulisten, die in einer anderen Region angeboten werden, geben Sie die Region mithilfe des `--region` Parameters an.

```
aws ec2 describe-instance-type-offerings \
  --region us-east-2
```

Ausgabe:

```
{
  "InstanceTypeOfferings": [
    {
      "InstanceType": "m5.2xlarge",
      "LocationType": "region",
      "Location": "us-east-2"
    },
    {
      "InstanceType": "t3.micro",
      "LocationType": "region",
      "Location": "us-east-2"
    },
    ...
  ]
}
```

Beispiel 2: Um die in einer Availability Zone angebotenen Instance-Typen aufzulisten

Das folgende `describe-instance-type-offerings` Beispiel listet die Instanztypen auf, die in der angegebenen Availability Zone angeboten werden. Die Availability Zone muss sich in der angegebenen Region befinden.

```
aws ec2 describe-instance-type-offerings \
  --location-type availability-zone \
  --filters Name=location,Values=us-east-2a \
  --region us-east-2
```

Beispiel 3: Um zu überprüfen, ob ein Instance-Typ unterstützt wird

Der folgende `describe-instance-type-offerings` Befehl gibt an, ob der `c5.xlarge` Instance-Typ in der angegebenen Region unterstützt wird.

```
aws ec2 describe-instance-type-offerings \
  --filters Name=instance-type,Values=c5.xlarge \
  --region us-east-2
```

Das folgende `describe-instance-type-offerings` Beispiel listet alle C5-Instance-Typen auf, die in der angegebenen Region unterstützt werden.

```
aws ec2 describe-instance-type-offerings \
  --filters Name=instance-type,Values=c5* \
```

```
--query "InstanceTypeOfferings[].InstanceType" \  
--region us-east-2
```

Ausgabe:

```
[  
  "c5d.12xlarge",  
  "c5d.9xlarge",  
  "c5n.xlarge",  
  "c5.xlarge",  
  "c5d.metal",  
  "c5n.metal",  
  "c5.large",  
  "c5d.2xlarge",  
  "c5n.4xlarge",  
  "c5.2xlarge",  
  "c5n.large",  
  "c5n.9xlarge",  
  "c5d.large",  
  "c5.18xlarge",  
  "c5d.18xlarge",  
  "c5.12xlarge",  
  "c5n.18xlarge",  
  "c5.metal",  
  "c5d.4xlarge",  
  "c5.24xlarge",  
  "c5d.xlarge",  
  "c5n.2xlarge",  
  "c5d.24xlarge",  
  "c5.9xlarge",  
  "c5.4xlarge"  
]
```

- API-Einheiten finden Sie unter [DescribeInstanceTypeOfferings AWS CLI Befehlsreferenz](#).

describe-instance-types

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-types`.

AWS CLI

Beispiel 1: So beschreiben Sie einen Instance-Typ

Im folgenden `describe-instance-types`-Beispiel werden die Details für den angegebenen Instance-Typ angezeigt.

```
aws ec2 describe-instance-types \  
  --instance-types t2.micro
```

Ausgabe:

```
{  
  "InstanceTypes": [  
    {  
      "InstanceType": "t2.micro",  
      "CurrentGeneration": true,  
      "FreeTierEligible": true,  
      "SupportedUsageClasses": [  
        "on-demand",  
        "spot"  
      ],  
      "SupportedRootDeviceTypes": [  
        "ebs"  
      ],  
      "BareMetal": false,  
      "Hypervisor": "xen",  
      "ProcessorInfo": {  
        "SupportedArchitectures": [  
          "i386",  
          "x86_64"  
        ],  
        "SustainedClockSpeedInGhz": 2.5  
      },  
      "VCpuInfo": {  
        "DefaultVCpus": 1,  
        "DefaultCores": 1,  
        "DefaultThreadsPerCore": 1,  
        "ValidCores": [  
          1  
        ],  
        "ValidThreadsPerCore": [  
          1  
        ]  
      },  
      "MemoryInfo": {  
        "SizeInMiB": 1024  
      }  
    }  
  ]  
}
```

```

    },
    "InstanceStorageSupported": false,
    "EbsInfo": {
      "EbsOptimizedSupport": "unsupported",
      "EncryptionSupport": "supported"
    },
    "NetworkInfo": {
      "NetworkPerformance": "Low to Moderate",
      "MaximumNetworkInterfaces": 2,
      "Ipv4AddressesPerInterface": 2,
      "Ipv6AddressesPerInterface": 2,
      "Ipv6Supported": true,
      "EnaSupport": "unsupported"
    },
    "PlacementGroupInfo": {
      "SupportedStrategies": [
        "partition",
        "spread"
      ]
    },
    "HibernationSupported": false,
    "BurstablePerformanceSupported": true,
    "DedicatedHostsSupported": false,
    "AutoRecoverySupported": true
  }
]
}

```

Weitere Informationen finden Sie unter [Instance-Typen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: So filtern Sie die verfügbaren Instance-Typen

Sie können einen Filter angeben, um die Ergebnisse auf Instance-Typen mit einem bestimmten Merkmal zu beschränken. Im folgenden `describe-instance-types`-Beispiel werden die Instance-Typen aufgeführt, die den Ruhezustand unterstützen.

```

aws ec2 describe-instance-types \
  --filters Name=hibernation-supported,Values=true --query
  'InstanceTypes[*].InstanceType'

```

Ausgabe:

```
[
  "m5.8xlarge",
  "r3.large",
  "c3.8xlarge",
  "r5.large",
  "m4.4xlarge",
  "c4.large",
  "m5.xlarge",
  "m4.xlarge",
  "c3.large",
  "c4.8xlarge",
  "c4.4xlarge",
  "c5.xlarge",
  "c5.12xlarge",
  "r5.4xlarge",
  "c5.4xlarge"
]
```

Weitere Informationen finden Sie unter [Instance-Typen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [DescribeInstanceTypes](#) in der AWS CLI Befehlsreferenz.

describe-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-instances`.

AWS CLI

Beispiel 1: So beschreiben Sie eine Instance

Das folgende `describe-instances`-Beispiel beschreibt die angegebene Instance.

```
aws ec2 describe-instances \
  --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{
  "Reservations": [
    {
      "Groups": [],
```

```
"Instances": [
  {
    "AmiLaunchIndex": 0,
    "ImageId": "ami-0abcdef1234567890",
    "InstanceId": "i-1234567890abcdef0",
    "InstanceType": "t3.nano",
    "KeyName": "my-key-pair",
    "LaunchTime": "2022-11-15T10:48:59+00:00",
    "Monitoring": {
      "State": "disabled"
    },
    "Placement": {
      "AvailabilityZone": "us-east-2a",
      "GroupName": "",
      "Tenancy": "default"
    },
    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
    "PrivateIpAddress": "10-0-0-157",
    "ProductCodes": [],
    "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
    "PublicIpAddress": "34.253.223.13",
    "State": {
      "Code": 16,
      "Name": "running"
    },
    "StateTransitionReason": "",
    "SubnetId": "subnet-04a636d18e83cfacb",
    "VpcId": "vpc-1234567890abcdef0",
    "Architecture": "x86_64",
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/xvda",
        "Ebs": {
          "AttachTime": "2022-11-15T10:49:00+00:00",
          "DeleteOnTermination": true,
          "Status": "attached",
          "VolumeId": "vol-02e6ccdca7de29cf2"
        }
      }
    ],
    "ClientToken": "1234abcd-1234-abcd-1234-d46a8903e9bc",
    "EbsOptimized": true,
    "EnaSupport": true,
```



```
    "Hypervisor": "xen",
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::111111111111:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
      "Id": "1111111111111111111111"
    },
    "NetworkInterfaces": [
      {
        "Association": {
          "IpOwnerId": "amazon",
          "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
          "PublicIp": "34.253.223.13"
        },
        "Attachment": {
          "AttachTime": "2022-11-15T10:48:59+00:00",
          "AttachmentId": "eni-attach-1234567890abcdefg",
          "DeleteOnTermination": true,
          "DeviceIndex": 0,
          "Status": "attached",
          "NetworkCardIndex": 0
        },
        "Description": "",
        "Groups": [
          {
            "GroupName": "launch-wizard-146",
            "GroupId": "sg-1234567890abcdefg"
          }
        ],
        "Ipv6Addresses": [],
        "MacAddress": "00:11:22:33:44:55",
        "NetworkInterfaceId": "eni-1234567890abcdefg",
        "OwnerId": "104024344472",
        "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
        "PrivateIpAddress": "10-0-0-157",
        "PrivateIpAddresses": [
          {
            "Association": {
              "IpOwnerId": "amazon",
              "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
              "PublicIp": "34.253.223.13"
            },
```

```
        "Primary": true,
        "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
        "PrivateIpAddress": "10-0-0-157"
    }
],
"SourceDestCheck": true,
"Status": "in-use",
"SubnetId": "subnet-1234567890abcdefg",
"VpcId": "vpc-1234567890abcdefg",
"InterfaceType": "interface"
}
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "launch-wizard-146",
        "GroupId": "sg-1234567890abcdefg"
    }
],
"SourceDestCheck": true,
"Tags": [
    {
        "Key": "Name",
        "Value": "my-instance"
    }
],
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 2
},
"CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
},
"HibernationOptions": {
    "Configured": false
},
"MetadataOptions": {
    "State": "applied",
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "enabled",
```

```

        "HttpProtocolIpv6": "disabled",
        "InstanceMetadataTags": "enabled"
    },
    "EnclaveOptions": {
        "Enabled": false
    },
    "PlatformDetails": "Linux/UNIX",
    "UsageOperation": "RunInstances",
    "UsageOperationUpdateTime": "2022-11-15T10:48:59+00:00",
    "PrivateDnsNameOptions": {
        "HostnameType": "ip-name",
        "EnableResourceNameDnsARecord": true,
        "EnableResourceNameDnsAAAARecord": false
    },
    "MaintenanceOptions": {
        "AutoRecovery": "default"
    }
}
],
"OwnerId": "111111111111",
"ReservationId": "r-1234567890abcdefg"
}
]
}

```

Beispiel 2: So filtern Sie nach Instances mit dem angegebenen Typ

Im folgenden `describe-instances`-Beispiel werden Filter verwendet, um die Ergebnisse auf Instances des angegebenen Typs zu beschränken.

```
aws ec2 describe-instances \
  --filters Name=instance-type,Values=m5.Large
```

Ein Beispiel für eine Ausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Auflisten und Filtern mithilfe von CLI im EC2 Amazon-Benutzerhandbuch](#).

Beispiel 3: So filtern Sie nach Instances mit dem angegebenen Typ und der angegebenen Availability Zone

Im folgenden `describe-instances`-Beispiel werden mehrere Filter verwendet, um die Ergebnisse auf Instances mit dem angegebenen Typ zu beschränken, die sich ebenfalls in der angegebenen Availability Zone befinden.

```
aws ec2 describe-instances \  
  --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-  
zone,Values=us-east-2c
```

Ein Beispiel für eine Ausgabe finden Sie in Beispiel 1.

Beispiel 4: Um mithilfe einer JSON Datei nach Instances mit dem angegebenen Typ und der angegebenen Availability Zone zu filtern

Das folgende `describe-instances` Beispiel verwendet eine JSON Eingabedatei, um dieselbe Filterung wie im vorherigen Beispiel durchzuführen. Wenn Filter komplizierter werden, können sie einfacher in einer JSON Datei angegeben werden.

```
aws ec2 describe-instances \  
  --filters file://filters.json
```

Inhalt von `filters.json`:

```
[  
  {  
    "Name": "instance-type",  
    "Values": ["t2.micro", "t3.micro"]  
  },  
  {  
    "Name": "availability-zone",  
    "Values": ["us-east-2c"]  
  }  
]
```

Ein Beispiel für eine Ausgabe finden Sie in Beispiel 1.

Beispiel 5: So filtern Sie nach Instances mit dem angegebenen Owner-Tag

Im folgenden `describe-instances`-Beispiel werden Tag-Filter verwendet, um die Ergebnisse unabhängig vom Tag-Wert auf Instances zu beschränken, die über ein Tag mit dem angegebenen Tag-Schlüssel (Owner) verfügen.

```
aws ec2 describe-instances \  
  --filters "Name=tag-key,Values=Owner"
```

Ein Beispiel für eine Ausgabe finden Sie in Beispiel 1.

Beispiel 6: So filtern Sie nach Instances mit dem angegebenen my-team-Tag-Wert

Im folgenden describe-instances-Beispiel werden Tag-Filter verwendet, um die Ergebnisse auf Instances zu beschränken, die ein Tag mit dem angegebenen Tag-Wert (my-team) haben, unabhängig vom Tag-Schlüssel.

```
aws ec2 describe-instances \  
  --filters "Name=tag-value,Values=my-team"
```

Ein Beispiel für eine Ausgabe finden Sie in Beispiel 1.

Beispiel 7: So filtern Sie nach Instances mit dem angegebenen Besitzer-Tag und my-team-Wert

Im folgenden describe-instances-Beispiel werden Tag-Filter verwendet, um die Ergebnisse auf Instances zu beschränken, die das angegebene Tag haben (Besitzer=my-team).

```
aws ec2 describe-instances \  
  --filters "Name=tag:Owner,Values=my-team"
```

Ein Beispiel für eine Ausgabe finden Sie in Beispiel 1.

Beispiel 8: Um nur Instanz und Subnetz IDs für alle Instanzen anzuzeigen

In den folgenden describe-instances Beispielen --query wird der Parameter verwendet, um nur die Instanz und das Subnetz IDs für alle Instanzen im JSON Format anzuzeigen.

Linux und macOS:

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}' \  
  --output json
```

Windows:

```
aws ec2 describe-instances ^  
  --query "Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}" ^
```

```
--output json
```

Ausgabe:

```
[
  {
    "Instance": "i-057750d42936e468a",
    "Subnet": "subnet-069beee9b12030077"
  },
  {
    "Instance": "i-001efd250faaa6ffa",
    "Subnet": "subnet-0b715c6b7db68927a"
  },
  {
    "Instance": "i-027552a73f021f3bd",
    "Subnet": "subnet-0250c25a1f4e15235"
  }
  ...
]
```

Beispiel 9: Um Instanzen des angegebenen Typs zu filtern und nur ihre Instanz anzuzeigen IDs

Im folgenden `describe-instances` Beispiel werden Filter verwendet, um die Ergebnisse auf Instanzen des angegebenen Typs zu beschränken, und der `--query` Parameter verwendet, um nur die Instanz anzuzeigen IDs.

```
aws ec2 describe-instances \
  --filters "Name=instance-type,Values=t2.micro" \
  --query "Reservations[*].Instances[*].[InstanceId]" \
  --output text
```

Ausgabe:

```
i-031c0dc19de2fb70c
i-00d8bfff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

Beispiel 10: Um Instances des angegebenen Typs zu filtern und nur ihre InstanzIDs, Availability Zone und den angegebenen Tag-Wert anzuzeigen

In den folgenden `describe-instances`-Beispielen werden die Instance-ID, die Availability Zone und der Wert des Name-Tags für Instances, die ein Tag mit dem Namen `tag-key` haben, im Tabellenformat angezeigt.

Linux und macOS:

```
aws ec2 describe-instances \
  --filters Name=tag-key,Values=Name \
  --query 'Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name `]]
[0].Value}' \
  --output table
```

Windows:

```
aws ec2 describe-instances ^
  --filters Name=tag-key,Values=Name ^
  --query "Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name `]]
[0].Value}" ^
  --output table
```

Ausgabe:

```
-----
|                               DescribeInstances                               |
+-----+-----+-----+
|      AZ      |      Instance      |      Name      |
+-----+-----+-----+
| us-east-2b  | i-057750d42936e468a | my-prod-server |
| us-east-2a  | i-001efd250faaa6ffa | test-server-1   |
| us-east-2a  | i-027552a73f021f3bd | test-server-2   |
+-----+-----+-----+
```

Beispiel 11: So beschreiben Sie Instances in einer Partition-Placement-Gruppe

Das folgende `describe-instances`-Beispiel beschreibt die angegebene Instance. Die Ausgabe enthält die Platzierungsinformationen für die Instance, die den Namen der Platzierungsgruppe und die Partitionsnummer für die Instance enthalten.

```
aws ec2 describe-instances \
```

```
--instance-ids i-0123a456700123456 \  
--query "Reservations[*].Instances[*].Placement"
```

Ausgabe:

```
[  
  [  
    {  
      "AvailabilityZone": "us-east-1c",  
      "GroupName": "HDFS-Group-A",  
      "PartitionNumber": 3,  
      "Tenancy": "default"  
    }  
  ]  
]
```

Weitere Informationen finden Sie unter [Beschreibung von Instances in einer Placement-Gruppe](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 12: So filtern Sie auf Instances mit der angegebenen Platzierungsgruppe und Partitionsnummer

Im folgenden `describe-instances`-Beispiel werden die Ergebnisse nur nach den Instances mit der angegebenen Platzierungsgruppe und Partitionsnummer gefiltert.

```
aws ec2 describe-instances \  
  --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-  
partition-number,Values=7"
```

Im Folgenden werden nur die relevanten Informationen aus der Ausgabe angezeigt.

```
"Instances": [  
  {  
    "InstanceId": "i-0123a456700123456",  
    "InstanceType": "r4.large",  
    "Placement": {  
      "AvailabilityZone": "us-east-1c",  
      "GroupName": "HDFS-Group-A",  
      "PartitionNumber": 7,  
      "Tenancy": "default"  
    }  
  }  
]
```



```

    },
    {
      "InstanceId": "i-9876a543210987654",
      "InstanceType": "r4.large",
      "Placement": {
        "AvailabilityZone": "us-east-1c",
        "GroupName": "HDFS-Group-A",
        "PartitionNumber": 7,
        "Tenancy": "default"
      }
    }
  ],

```

Weitere Informationen finden Sie unter [Beschreibung von Instances in einer Placement-Gruppe](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 13: So filtern Sie nach Instances, die so konfiguriert sind, dass sie den Zugriff auf Tags aus Instance-Metadaten erlauben

Im folgenden `describe-instances`-Beispiel werden die Ergebnisse nur nach den Instances gefiltert, die so konfiguriert sind, dass sie den Zugriff auf Instance-Tags aus den Instance-Metadaten ermöglichen.

```

aws ec2 describe-instances \
  --filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \
  --query "Reservations[*].Instances[*].InstanceId" \
  --output text

```

Im Folgenden wird die erwartete Ausgabe dargestellt.

```

i-1234567890abcdefg
i-abcdefg1234567890
i-11111111aaaaaaaaa
i-aaaaaaaa111111111

```

Weitere Informationen finden Sie unter [Arbeiten mit Instance-Tags in Instance-Metadaten](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstances](#) in der AWS CLI Befehlsreferenz.

describe-internet-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-internet-gateways`.

AWS CLI

Um ein Internet-Gateway zu beschreiben

Das folgende `describe-internet-gateways` Beispiel beschreibt das angegebene Internet-Gateway.

```
aws ec2 describe-internet-gateways \  
  --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

Ausgabe:

```
{  
  "InternetGateways": [  
    {  
      "Attachments": [  
        {  
          "State": "available",  
          "VpcId": "vpc-0a60eb65b4EXAMPLE"  
        }  
      ],  
      "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",  
      "OwnerId": "123456789012",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "my-igw"  
        }  
      ]  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Internet-Gateways](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeInternetGateways](#) in der AWS CLI Befehlsreferenz.

describe-ipam-pools

Das folgende Codebeispiel zeigt die Verwendung `describe-ipam-pools`.

AWS CLI

Um die Details für einen IPAM Pool anzuzeigen

Das folgende `describe-ipam-pools` Beispiel zeigt die Details für Pools.

(Linux):

```
aws ec2 describe-ipam-pools \  
  --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-  
scope-02fc38cd4c48e7d38
```

(Windows):

```
aws ec2 describe-ipam-pools ^  
  --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-  
scope-02fc38cd4c48e7d38
```

Ausgabe:

```
{  
  "IpamPools": [  
    {  
      "OwnerId": "123456789012",  
      "IpamPoolId": "ipam-pool-02ec043a19bbe5d08",  
      "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-  
pool-02ec043a19bbe5d08",  
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-02fc38cd4c48e7d38",  
      "IpamScopeType": "private",  
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
      "IpamRegion": "us-east-1",  
      "Locale": "None",  
      "PoolDepth": 1,  
      "State": "create-complete",  
      "AutoImport": true,  
      "AddressFamily": "ipv4",  
      "AllocationMinNetmaskLength": 16,  
      "AllocationMaxNetmaskLength": 26,  
      "AllocationDefaultNetmaskLength": 24,  
      "AllocationResourceTags": [  
        {  
          "Key": "Environment",
```

```
        "Value": "Preprod"
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "Preprod pool"
      }
    ]
  }
]
```

- API-Einheiten finden Sie [Beschreibung von IPAM-Pools](#) in der AWS CLI Befehlsreferenz.

describe-ipam-resource-discoveries

Das folgende Codebeispiel zeigt die Verwendung `describe-ipam-resource-discoveries`.

AWS CLI

Beispiel 1: Vollständige Details zu Ressourcenentdeckungen anzeigen

In diesem Beispiel sind Sie ein delegierter IPAM Administrator, der eine Ressourcenerkennung erstellen und mit dem IPAM Administrator in einer anderen AWS Organisation teilen möchte, damit der Administrator die IP-Adressen der Ressourcen in Ihrer Organisation verwalten und überwachen kann.

Dieses Beispiel kann nützlich sein, wenn:

Sie haben versucht, eine Ressourcensuche zu erstellen, haben aber die Fehlermeldung erhalten, dass Sie Ihr Limit von 1 erreicht haben. Sie stellen fest, dass Sie möglicherweise bereits eine Ressourcensuche erstellt haben und diese in Ihrem Konto anzeigen möchten. Sie haben Ressourcen in einer Region, die nicht von der erkannt werden. IPAM Sie möchten die für die Ressource `--operating-regions` definierten Ressourcen einsehen und sicherstellen, dass Sie die richtige Region als Betriebsregion hinzugefügt haben, damit die Ressourcen dort gefunden werden können.

Das folgende `describe-ipam-resource-discoveries` Beispiel listet die Details der Ressourcensuche in Ihrem AWS Konto auf. Sie können eine Ressourcenerkennung pro AWS Region durchführen.

```
aws ec2 describe-ipam-resource-discoveries \  
  --region us-east-1
```

Ausgabe:

```
{  
  "IpamResourceDiscoveries": [  
    {  
      "OwnerId": "149977607591",  
      "IpamResourceDiscoveryId": "ipam-res-disco-0f8bdee9067137c0d",  
      "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-  
discovery/ipam-res-disco-0f8bdee9067137c0d",  
      "IpamResourceDiscoveryRegion": "us-east-1",  
      "OperatingRegions": [  
        {  
          "RegionName": "us-east-1"  
        }  
      ],  
      "IsDefault": false,  
      "State": "create-complete",  
      "Tags": []  
    }  
  ]  
}
```

Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

Beispiel 2: Nur die Suche nach Ressourcen anzeigen IDs

Im folgenden `describe-ipam-resource-discoveries` Beispiel wird die ID der Ressourcenerkennung in Ihrem AWS Konto aufgeführt. Sie können pro AWS Region eine Ressourcenerkennung durchführen.

```
aws ec2 describe-ipam-resource-discoveries \  
  --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \  
  --output text
```

Ausgabe:

```
ipam-res-disco-0481e39b242860333
```

Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

- APIEinzelheiten finden Sie [DescribeIpamResourceDiscoveries](#) in der AWS CLI Befehlsreferenz.

describe-ipam-resource-discovery-associations

Das folgende Codebeispiel zeigt die Verwendung `describe-ipam-resource-discovery-associations`.

AWS CLI

Um alle Resource Discovery-Verknüpfungen mit Ihrem anzuzeigen IPAM

In diesem Beispiel sind Sie ein IPAM delegierter Administrator, der Ressourcenentdeckungen mit Ihrem verknüpft hat IPAM, um andere Konten mit Ihrem IPAM zu integrieren. Sie haben festgestellt, IPAM dass Sie die Ressourcen in den Betriebsregionen der Ressourcenerkennung nicht wie erwartet entdecken. Sie möchten den Status und den Status der Ressourcenerkennung überprüfen, um sicherzustellen, dass das Konto, mit dem sie erstellt wurde, immer noch aktiv ist und die Ressourcenerkennung weiterhin gemeinsam genutzt wird.

Das `--region` muss die Heimatregion Ihrer sein IPAM.

Im folgenden `describe-ipam-resource-discovery-associations` Beispiel werden die Resource Discovery-Verknüpfungen in Ihrem AWS Konto aufgeführt.

```
aws ec2 describe-ipam-resource-discovery-associations \
  --region us-east-1
```

Ausgabe:

```
{
  "IpamResourceDiscoveryAssociations": [
    {
      "OwnerId": "320805250157",
      "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-05e6b45eca5bf5cf7",
      "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-05e6b45eca5bf5cf7",
      "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
```

```

    "IpamId": "ipam-005f921c17ebd5107",
    "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
    "IpamRegion": "us-east-1",
    "IsDefault": true,
    "ResourceDiscoveryStatus": "active",
    "State": "associate-complete",
    "Tags": []
  },
  {
    "OwnerId": "149977607591",
    "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-0dfd21ae189ab5f62",
    "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::149977607591:ipam-
resource-discovery-association/ipam-res-disco-assoc-0dfd21ae189ab5f62",
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
    "IpamId": "ipam-005f921c17ebd5107",
    "IpamArn": "arn:aws:ec2::149977607591:ipam/ipam-005f921c17ebd5107",
    "IpamRegion": "us-east-1",
    "IsDefault": false,
    "ResourceDiscoveryStatus": "active",
    "State": "create-complete",
    "Tags": []
  }
]
}

```

In diesem Beispiel stellen Sie nach der Ausführung dieses Befehls fest, dass Sie über eine nicht standardmäßige Ressourcenerkennung verfügen ("IsDefault": false ``) that is ``"ResourceDiscoveryStatus": "not-found"und"State": "create-complete". Das Konto des Besitzers von Resource Discovery wurde geschlossen. Wenn Sie in einem anderen Fall feststellen, dass dies "ResourceDiscoveryStatus": "not-found" und ist"State": "associate-complete", deutet dies darauf hin, dass einer der folgenden Fälle eingetreten ist:

Die Ressourcensuche wurde vom Besitzer der Ressourcenermittlung gelöscht. Der Besitzer der Ressourcenermittlung hat die Freigabe der Ressourcensuche aufgehoben.

Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

- APIEinzelheiten finden Sie [BeschreibepamResourceDiscoveryAssociations](#)in der AWS CLI Befehlsreferenz.

describe-ipam-scopes

Das folgende Codebeispiel zeigt die Verwendung `describe-ipam-scopes`.

AWS CLI

Um die Details für einen IPAM Bereich anzuzeigen

Das folgende `describe-ipam-scopes` Beispiel zeigt die Details für Bereiche.

```
aws ec2 describe-ipam-scopes \
  --filters Name=owner-id,Values=123456789012 Name=ipam-
id,Values=ipam-08440e7a3acde3908
```

Ausgabe:

```
{
  "IpamScopes": [
    {
      "OwnerId": "123456789012",
      "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
      "IpamRegion": "us-east-1",
      "IpamScopeType": "private",
      "IsDefault": true,
      "PoolCount": 2,
      "State": "create-complete",
      "Tags": []
    },
    {
      "OwnerId": "123456789012",
      "IpamScopeId": "ipam-scope-0b9eed026396dbc16",
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-0b9eed026396dbc16",
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
      "IpamRegion": "us-east-1",
      "IpamScopeType": "public",
      "IsDefault": true,
      "PoolCount": 0,
      "State": "create-complete",
      "Tags": []
    }
  ],
}
```



```

    {
      "OwnerId": "123456789012",
      "IpamScopeId": "ipam-scope-0f1aff29486355c22",
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-0f1aff29486355c22",
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
      "IpamRegion": "us-east-1",
      "IpamScopeType": "private",
      "IsDefault": false,
      "Description": "Example description",
      "PoolCount": 0,
      "State": "create-complete",
      "Tags": [
        {
          "Key": "Name",
          "Value": "Example name value"
        }
      ]
    }
  ]
}

```

- API-Einheiten finden Sie [Beschreibung von IPAM-Scope](#) in der AWS CLI Befehlsreferenz.

describe-ipams

Das folgende Codebeispiel zeigt die Verwendung `describe-ipams`.

AWS CLI

Um die Details für ein anzugeben IPAM

Das folgende `describe-ipams` Beispiel zeigt die Details eines IPAM.

```

aws ec2 describe-ipams \
  --filters Name=owner-id,Values=123456789012

```

Ausgabe:

```

{
  "Ipams": [
    {
      "OwnerId": "123456789012",

```

```

    "IpamId": "ipam-08440e7a3acde3908",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
    "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
    "ScopeCount": 3,
    "OperatingRegions": [
      {
        "RegionName": "us-east-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-west-1"
      }
    ],
    "State": "create-complete",
    "Tags": [
      {
        "Key": "Name",
        "Value": "ExampleIPAM"
      }
    ]
  }
]
}

```

- API-Einheiten finden Sie [Beschreibung](#) in der AWS CLI Befehlsreferenz.

describe-ipv6-pools

Das folgende Codebeispiel zeigt die Verwendung `describe-ipv6-pools`.

AWS CLI

Um Ihre IPv6 Adresspools zu beschreiben

Im folgenden `describe-ipv6-pools` Beispiel werden Details für all Ihre IPv6 Adresspools angezeigt.

```
aws ec2 describe-ipv6-pools
```

Ausgabe:

```
{
  "Ipv6Pools": [
    {
      "PoolId": "ipv6pool-ec2-012345abc12345abc",
      "PoolCidrBlocks": [
        {
          "Cidr": "2001:db8:123::/48"
        }
      ],
      "Tags": [
        {
          "Key": "pool-1",
          "Value": "public"
        }
      ]
    }
  ]
}
```

- API-Einheiten finden Sie unter [Beschreiben von IPv6-Pools](#) in der AWS CLI-Befehlsreferenz.

describe-key-pairs

Das folgende Codebeispiel zeigt die Verwendung von `describe-key-pairs`.

AWS CLI

So zeigen Sie ein Schlüsselpaar an:

Im folgenden `describe-key-pairs`-Beispiel werden Informationen zu dem angegebenen Schlüsselpaar angezeigt.

```
aws ec2 describe-key-pairs \
  --key-names my-key-pair
```

Ausgabe:

```
{
  "KeyPairs": [
    {
```

```
    "KeyPairId": "key-0b94643da6EXAMPLE",
    "KeyFingerprint":
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
    "KeyName": "my-key-pair",
    "KeyType": "rsa",
    "Tags": [],
    "CreateTime": "2022-05-27T21:51:16.000Z"
  }
]
}
```

Weitere Informationen finden Sie unter [Describe public keys](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeKeyPairs](#) in der AWS CLI Befehlsreferenz.

describe-launch-template-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-launch-template-versions`.

AWS CLI

Um Versionen von Startvorlagen zu beschreiben

In diesem Beispiel werden die Versionen der angegebenen Startvorlage beschrieben.

Befehl:

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

Ausgabe:

```
{
  "LaunchTemplateVersions": [
    {
      "LaunchTemplateId": "lt-068f72b72934aff71",
      "LaunchTemplateName": "Webservers",
      "VersionNumber": 3,
      "CreatedBy": "arn:aws:iam::123456789102:root",
      "LaunchTemplateData": {
        "KeyName": "kp-us-east",
        "ImageId": "ami-6057e21a",
        "InstanceType": "t2.small",
        "NetworkInterfaces": [
```

```
        {
            "SubnetId": "subnet-7b16de0c",
            "DeviceIndex": 0,
            "Groups": [
                "sg-7c227019"
            ]
        }
    ],
    "DefaultVersion": false,
    "CreateTime": "2017-11-20T13:19:54.000Z"
},
{
    "LaunchTemplateId": "lt-068f72b72934aff71",
    "LaunchTemplateName": "Webservers",
    "VersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789102:root",
    "LaunchTemplateData": {
        "KeyName": "kp-us-east",
        "ImageId": "ami-6057e21a",
        "InstanceType": "t2.medium",
        "NetworkInterfaces": [
            {
                "SubnetId": "subnet-1a2b3c4d",
                "DeviceIndex": 0,
                "Groups": [
                    "sg-7c227019"
                ]
            }
        ]
    },
    "DefaultVersion": false,
    "CreateTime": "2017-11-20T13:12:32.000Z"
},
{
    "LaunchTemplateId": "lt-068f72b72934aff71",
    "LaunchTemplateName": "Webservers",
    "VersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789102:root",
    "LaunchTemplateData": {
        "UserData": "",
        "KeyName": "kp-us-east",
        "ImageId": "ami-aabbcc11",
        "InstanceType": "t2.medium",
```

```
    "NetworkInterfaces": [
      {
        "SubnetId": "subnet-7b16de0c",
        "DeviceIndex": 0,
        "DeleteOnTermination": false,
        "Groups": [
          "sg-7c227019"
        ],
        "AssociatePublicIpAddress": true
      }
    ],
    "DefaultVersion": true,
    "CreateTime": "2017-11-20T12:52:33.000Z"
  }
]
```

- API-Einheiten finden Sie [DescribeLaunchTemplateVersions](#) in der AWS CLI Befehlsreferenz.

describe-launch-templates

Das folgende Codebeispiel zeigt die Verwendung von `describe-launch-templates`.

AWS CLI

Um Startvorlagen zu beschreiben

In diesem Beispiel werden Ihre Startvorlagen beschrieben.

Befehl:

```
aws ec2 describe-launch-templates
```

Ausgabe:

```
{
  "LaunchTemplates": [
    {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0e06d290751193123",
      "LaunchTemplateName": "TemplateForWebServer",
```

```

    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-27T09:30:23.000Z"
  },
  {
    "LatestVersionNumber": 6,
    "LaunchTemplateId": "lt-0c45b5e061ec98456",
    "LaunchTemplateName": "DBServersTemplate",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-20T09:25:22.000Z"
  },
  {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0d47d774e8e52dabc",
    "LaunchTemplateName": "MyLaunchTemplate2",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-02T12:06:21.000Z"
  },
  {
    "LatestVersionNumber": 3,
    "LaunchTemplateId": "lt-01e5f948eb4f589d6",
    "LaunchTemplateName": "testingtemplate2",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:sts::123456789012:assumed-role/AdminRole/
i-03ee35176e2e5aabc",
    "CreateTime": "2017-12-01T08:19:48.000Z"
  },
]
}

```

- API-Einheiten finden Sie [DescribeLaunchTemplates](#) in der AWS CLI Befehlsreferenz.

describe-local-gateway-route-table-virtual-interface-group-associations

Das folgende Codebeispiel zeigt die Verwendung `describe-local-gateway-route-table-virtual-interface-group-associations`.

AWS CLI

Zur Beschreibung von Verknüpfungen zwischen virtuellen Schnittstellengruppen und Routentabellen für lokale Gateways

Im folgenden `describe-local-gateway-route-table-virtual-interface-group-associations` Beispiel werden die Verknüpfungen zwischen virtuellen Schnittstellengruppen und lokalen Gateway-Routentabellen in Ihrem AWS Konto beschrieben.

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

Ausgabe:

```
{
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociations": [
    {
      "LocalGatewayRouteTableVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
      "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
      "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:123456789012:local-gateway-route-table/lgw-rtb-059615ef7dEXAMPLE",
      "OwnerId": "123456789012",
      "State": "associated",
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit lokalen Gateways](#) im AWS Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations](#) in der AWS CLI Befehlsreferenz.

describe-local-gateway-route-table-vpc-associations

Das folgende Codebeispiel zeigt die Verwendung `describe-local-gateway-route-table-vpc-associations`.

AWS CLI

Zur Beschreibung der Verknüpfungen zwischen Routentabellen VPCs und lokalen Gateway-Routentabellen

Im folgenden `describe-local-gateway-route-table-vpc-associations` Beispiel werden Informationen zur angegebenen Zuordnung zwischen Routentabellen VPCs und lokalen Gateway-Routentabellen angezeigt.

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
  --local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

Ausgabe:

```
{
  "LocalGatewayRouteTableVpcAssociation": {
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af1EXAMPLE",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
    "VpcId": "vpc-0efe9bde08EXAMPLE",
    "State": "associated"
  }
}
```

Weitere Informationen finden Sie unter [Routing-Tabellen für lokale Gateways](#) im -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeLocalGatewayRouteTableVpcAssociations](#) unter AWS CLI Befehlsreferenz.

`describe-local-gateway-route-tables`

Das folgende Codebeispiel zeigt die Verwendung `describe-local-gateway-route-tables`.

AWS CLI

Um Ihre Local Gateway-Routentabellen zu beschreiben

Im folgenden `describe-local-gateway-route-tables` Beispiel werden Details zu den Routentabellen des lokalen Gateways angezeigt.

```
aws ec2 describe-local-gateway-route-tables
```

Ausgabe:

```
{
```

```
"LocalGatewayRouteTables": [  
  {  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",  
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",  
    "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/  
op-0dc11b66edEXAMPLE",  
    "State": "available"  
  }  
]  
}
```

- API-Einheiten finden Sie [DescribeLocalGatewayRouteTables](#) unter AWS CLI Befehlsreferenz.

describe-local-gateway-virtual-interface-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-local-gateway-virtual-interface-groups`.

AWS CLI

Um virtuelle Schnittstellengruppen für lokale Gateways zu beschreiben

Im folgenden `describe-local-gateway-virtual-interface-groups` Beispiel werden die virtuellen Schnittstellengruppen für das lokale Gateway in Ihrem AWS Konto beschrieben.

```
aws ec2 describe-local-gateway-virtual-interface-groups
```

Ausgabe:

```
{  
  "LocalGatewayVirtualInterfaceGroups": [  
    {  
      "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",  
      "LocalGatewayVirtualInterfaceIds": [  
        "lgw-vif-01a23bc4d5EXAMPLE",  
        "lgw-vif-543ab21012EXAMPLE"  
      ],  
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",  
      "OwnerId": "123456789012",  
      "Tags": []  
    }  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit lokalen Gateways](#) im AWS Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeLocalGatewayVirtualInterfaceGroups](#) in der AWS CLI Befehlsreferenz.

describe-local-gateway-virtual-interfaces

Das folgende Codebeispiel zeigt die Verwendung `describe-local-gateway-virtual-interfaces`.

AWS CLI

Um virtuelle Schnittstellen für lokale Gateways zu beschreiben

Im folgenden `describe-local-gateway-virtual-interfaces` Beispiel werden die virtuellen Schnittstellen des lokalen Gateways in Ihrem AWS Konto beschrieben.

```
aws ec2 describe-local-gateway-virtual-interfaces
```

Ausgabe:

```
{
  "LocalGatewayVirtualInterfaces": [
    {
      "LocalGatewayVirtualInterfaceId": "lgw-vif-01a23bc4d5EXAMPLE",
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "Vlan": 2410,
      "LocalAddress": "0.0.0.0/0",
      "PeerAddress": "0.0.0.0/0",
      "LocalBgpAsn": 65010,
      "PeerBgpAsn": 65000,
      "OwnerId": "123456789012",
      "Tags": []
    },
    {
      "LocalGatewayVirtualInterfaceId": "lgw-vif-543ab21012EXAMPLE",
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
```

```
        "Vlan": 2410,
        "LocalAddress": "0.0.0.0/0",
        "PeerAddress": "0.0.0.0/0",
        "LocalBgpAsn": 65010,
        "PeerBgpAsn": 65000,
        "OwnerId": "123456789012",
        "Tags": []
    }
]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit lokalen Gateways](#) im AWS Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeLocalGatewayVirtualInterfaces](#) in der AWS CLI Befehlsreferenz.

describe-local-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-local-gateways`.

AWS CLI

Um Ihre lokalen Gateways zu beschreiben

Im folgenden `describe-local-gateways` Beispiel werden Details zu den lokalen Gateways angezeigt, die Ihnen zur Verfügung stehen.

```
aws ec2 describe-local-gateways
```

Ausgabe:

```
{
  "LocalGateways": [
    {
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0dc11b66ed59f995a",
      "OwnerId": "123456789012",
      "State": "available"
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie unter [DescribeLocalGateways AWS CLI](#) Befehlsreferenz.

describe-locked-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-locked-snapshots`.

AWS CLI

Um den Sperrstatus eines Snapshots zu beschreiben

Das folgende `describe-locked-snapshots` Beispiel beschreibt den Sperrstatus des angegebenen Snapshots.

```
aws ec2 describe-locked-snapshots \
  --snapshot-ids snap-0b5e733b4a8df6e0d
```

Ausgabe:

```
{
  "Snapshots": [
    {
      "OwnerId": "123456789012",
      "SnapshotId": "snap-0b5e733b4a8df6e0d",
      "LockState": "governance",
      "LockDuration": 365,
      "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",
      "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00",
      "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Snapshot Lock](#) im EBS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeLockedSnapshots](#) in der AWS CLI Befehlsreferenz.

describe-managed-prefix-lists

Das folgende Codebeispiel zeigt die Verwendung `describe-managed-prefix-lists`.

AWS CLI

Um verwaltete Präfixlisten zu beschreiben

Im folgenden `describe-managed-prefix-lists` Beispiel werden die Präfixlisten beschrieben, die dem AWS Konto gehören `123456789012`.

```
aws ec2 describe-managed-prefix-lists \  
  --filters Name=owner-id,Values=123456789012
```

Ausgabe:

```
{  
  "PrefixLists": [  
    {  
      "PrefixListId": "pl-11223344556677aabb",  
      "AddressFamily": "IPv6",  
      "State": "create-complete",  
      "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-11223344556677aabb",  
      "PrefixListName": "vpc-ipv6-cidrs",  
      "MaxEntries": 25,  
      "Version": 1,  
      "Tags": [],  
      "OwnerId": "123456789012"  
    },  
    {  
      "PrefixListId": "pl-0123456abcabcabc1",  
      "AddressFamily": "IPv4",  
      "State": "active",  
      "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabcabc1",  
      "PrefixListName": "vpc-cidrs",  
      "MaxEntries": 10,  
      "Version": 1,  
      "Tags": [],  
      "OwnerId": "123456789012"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeManagedPrefixLists](#) in der AWS CLI Befehlsreferenz.

describe-moving-addresses

Das folgende Codebeispiel zeigt die Verwendung `describe-moving-addresses`.

AWS CLI

Um Ihre Umzugsadressen zu beschreiben

In diesem Beispiel werden alle Ihre verschiebenden Elastic-IP-Adressen beschrieben.

Befehl:

```
aws ec2 describe-moving-addresses
```

Ausgabe:

```
{
  "MovingAddressStatuses": [
    {
      "PublicIp": "198.51.100.0",
      "MoveStatus": "MovingToVpc"
    }
  ]
}
```

In diesem Beispiel werden alle Adressen beschrieben, die auf die EC2 - VPC -Plattform verschoben werden.

Befehl:

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- API-Einheiten finden Sie [DescribeMovingAddresses](#) in der AWS CLI Befehlsreferenz.

describe-nat-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-nat-gateways`.

AWS CLI

Beispiel 1: Um ein öffentliches NAT Gateway zu beschreiben

Das folgende `describe-nat-gateways` Beispiel beschreibt das angegebene öffentliche NAT Gateway.

```
aws ec2 describe-nat-gateways \  
  --nat-gateway-id nat-01234567890abcdef
```

Ausgabe:

```
{  
  "NatGateways": [  
    {  
      "CreateTime": "2023-08-25T01:56:51.000Z",  
      "NatGatewayAddresses": [  
        {  
          "AllocationId": "eipalloc-0790180cd2EXAMPLE",  
          "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
          "PrivateIp": "10.0.0.211",  
          "PublicIp": "54.85.121.213",  
          "AssociationId": "eipassoc-04d295cc9b8815b24",  
          "IsPrimary": true,  
          "Status": "succeeded"  
        },  
        {  
          "AllocationId": "eipalloc-0be6ecac95EXAMPLE",  
          "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
          "PrivateIp": "10.0.0.74",  
          "PublicIp": "3.211.231.218",  
          "AssociationId": "eipassoc-0f96bdca17EXAMPLE",  
          "IsPrimary": false,  
          "Status": "succeeded"  
        }  
      ],  
      "NatGatewayId": "nat-01234567890abcdef",  
      "State": "available",  
      "SubnetId": "subnet-655eab5f08EXAMPLE",  
      "VpcId": "vpc-098eb5ef58EXAMPLE",  
      "Tags": [  
        {  
          "Key": "Name",
```



```

        "Value": "public-nat"
      }
    ],
    "ConnectivityType": "public"
  }
]
}

```

Beispiel 2: Um ein privates NAT Gateway zu beschreiben

Das folgende `describe-nat-gateways` Beispiel beschreibt das angegebene private NAT Gateway.

```

aws ec2 describe-nat-gateways \
  --nat-gateway-id nat-1234567890abcdef0

```

Ausgabe:

```

{
  "NatGateways": [
    {
      "CreateTime": "2023-08-25T00:50:05.000Z",
      "NatGatewayAddresses": [
        {
          "NetworkInterfaceId": "eni-0065a61b324d1897a",
          "PrivateIp": "10.0.20.240",
          "IsPrimary": true,
          "Status": "succeeded"
        },
        {
          "NetworkInterfaceId": "eni-0065a61b324d1897a",
          "PrivateIp": "10.0.20.33",
          "IsPrimary": false,
          "Status": "succeeded"
        },
        {
          "NetworkInterfaceId": "eni-0065a61b324d1897a",
          "PrivateIp": "10.0.20.197",
          "IsPrimary": false,
          "Status": "succeeded"
        }
      ],
      "NatGatewayId": "nat-1234567890abcdef0",
    }
  ]
}

```

```
    "State": "available",
    "SubnetId": "subnet-08fc749671EXAMPLE",
    "VpcId": "vpc-098eb5ef58EXAMPLE",
    "Tags": [
      {
        "Key": "Name",
        "Value": "private-nat"
      }
    ],
    "ConnectivityType": "private"
  }
]
```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeNatGateways](#) in der AWS CLI Befehlsreferenz.

describe-network-acls

Das folgende Codebeispiel zeigt die Verwendung `describe-network-acls`.

AWS CLI

Um Ihr Netzwerk zu beschreiben ACLs

Im folgenden `describe-network-acls` Beispiel werden Details zu Ihrem Netzwerk ACLs abgerufen.

```
aws ec2 describe-network-acls
```

Ausgabe:

```
{
  "NetworkAcls": [
    {
      "Associations": [
        {
          "NetworkAclAssociationId": "aclassoc-0c1679dc41EXAMPLE",
          "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
          "SubnetId": "subnet-0931fc2fa5EXAMPLE"
        }
      ],

```

```
    "Entries": [  
      {  
        "CidrBlock": "0.0.0.0/0",  
        "Egress": true,  
        "Protocol": "-1",  
        "RuleAction": "allow",  
        "RuleNumber": 100  
      },  
      {  
        "CidrBlock": "0.0.0.0/0",  
        "Egress": true,  
        "Protocol": "-1",  
        "RuleAction": "deny",  
        "RuleNumber": 32767  
      },  
      {  
        "CidrBlock": "0.0.0.0/0",  
        "Egress": false,  
        "Protocol": "-1",  
        "RuleAction": "allow",  
        "RuleNumber": 100  
      },  
      {  
        "CidrBlock": "0.0.0.0/0",  
        "Egress": false,  
        "Protocol": "-1",  
        "RuleAction": "deny",  
        "RuleNumber": 32767  
      }  
    ],  
    "IsDefault": true,  
    "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",  
    "Tags": [],  
    "VpcId": "vpc-06e4ab6c6cEXAMPLE",  
    "OwnerId": "111122223333"  
  },  
  {  
    "Associations": [],  
    "Entries": [  
      {  
        "CidrBlock": "0.0.0.0/0",  
        "Egress": true,  
        "Protocol": "-1",  
        "RuleAction": "allow",
```

```
    "RuleNumber": 100
  },
  {
    "Egress": true,
    "Ipv6CidrBlock": "::/0",
    "Protocol": "-1",
    "RuleAction": "allow",
    "RuleNumber": 101
  },
  {
    "CidrBlock": "0.0.0.0/0",
    "Egress": true,
    "Protocol": "-1",
    "RuleAction": "deny",
    "RuleNumber": 32767
  },
  {
    "Egress": true,
    "Ipv6CidrBlock": "::/0",
    "Protocol": "-1",
    "RuleAction": "deny",
    "RuleNumber": 32768
  },
  {
    "CidrBlock": "0.0.0.0/0",
    "Egress": false,
    "Protocol": "-1",
    "RuleAction": "allow",
    "RuleNumber": 100
  },
  {
    "Egress": false,
    "Ipv6CidrBlock": "::/0",
    "Protocol": "-1",
    "RuleAction": "allow",
    "RuleNumber": 101
  },
  {
    "CidrBlock": "0.0.0.0/0",
    "Egress": false,
    "Protocol": "-1",
    "RuleAction": "deny",
    "RuleNumber": 32767
  },
}
```

```

        {
            "Egress": false,
            "Ipv6CidrBlock": "::/0",
            "Protocol": "-1",
            "RuleAction": "deny",
            "RuleNumber": 32768
        }
    ],
    "IsDefault": true,
    "NetworkAclId": "acl-0e2a78e4e2EXAMPLE",
    "Tags": [],
    "VpcId": "vpc-03914afb3eEXAMPLE",
    "OwnerId": "111122223333"
}
]
}

```

Weitere Informationen finden Sie ACLs im AWS VPCBenutzerhandbuch unter [Netzwerk](#).

- APIEinzelheiten finden Sie [DescribeNetworkAcls](#) in der AWS CLI Befehlsreferenz.

describe-network-insights-access-scope-analyses

Das folgende Codebeispiel zeigt die Verwendung `describe-network-insights-access-scope-analyses`.

AWS CLI

Um Network Insights zu beschreiben, greifen Sie auf Umfangsanalysen zu

Das folgende `describe-network-insights-access-scope-analyses` Beispiel beschreibt die Analyse des Zugriffsumfangs in Ihrem AWS Konto.

```
aws ec2 describe-network-insights-access-scope-analyses \
  --region us-east-1
```

Ausgabe:

```

{
  "NetworkInsightsAccessScopeAnalyses": [
    {
      "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111",

```

```

    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789111",
    "NetworkInsightsAccessScopeId": "nis-123456789222",
    "Status": "succeeded",
    "StartDate": "2022-01-25T19:45:36.842000+00:00",
    "FindingsFound": "true",
    "Tags": []
  }
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer using the AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [DescribeNetworkInsightsAccessScopeAnalyses](#) unter AWS CLI Befehlsreferenz.

describe-network-insights-access-scopes

Das folgende Codebeispiel zeigt die Verwendung `describe-network-insights-access-scopes`.

AWS CLI

Um die Zugriffsbereiche von Network Insights zu beschreiben

Das folgende `describe-network-insights-access-scopes` Beispiel beschreibt die Zugriffsumfangsanalysen in Ihrem Konto. AWS

```

aws ec2 describe-network-insights-access-scopes \
  --region us-east-1

```

Ausgabe:

```

{
  "NetworkInsightsAccessScopes": [
    {
      "NetworkInsightsAccessScopeId": "nis-123456789111",
      "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789111",
      "CreatedDate": "2021-11-29T21:12:41.416000+00:00",

```

```

        "UpdatedDate": "2021-11-29T21:12:41.416000+00:00",
        "Tags": []
      }
    ]
  }

```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer using the AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [DescribeNetworkInsightsAccessScopes](#) unter AWS CLI Befehlsreferenz.

describe-network-insights-analyses

Das folgende Codebeispiel zeigt die Verwendung `describe-network-insights-analyses`.

AWS CLI

Um die Ergebnisse einer Pfadanalyse anzuzeigen

Das folgende `describe-network-insights-analyses` Beispiel beschreibt die angegebene Analyse. In diesem Beispiel ist die Quelle ein Internet-Gateway, das Ziel eine EC2 Instanz und das Protokoll ist TCP. Die Analyse war erfolgreich (Status ist `succeeded`) und der Pfad ist nicht erreichbar (`NetworkPathFound` ist `false`). Der Erklärungscode `ENI_SG_RULES_MISMATCH` weist darauf hin, dass die Sicherheitsgruppe für die Instance keine Regel enthält, die Datenverkehr auf dem Zielport zulässt.

```

aws ec2 describe-network-insights-analyses \
  --network-insights-analysis-ids nia-02207aa13eb480c7a

```

Ausgabe:

```

{
  "NetworkInsightsAnalyses": [
    {
      "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
      "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-insights-analysis/nia-02207aa13eb480c7a",
      "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
      "StartDate": "2021-01-20T22:58:37.495Z",

```

```

    "Status": "succeeded",
    "NetworkPathFound": false,
    "Explanations": [
      {
        "Direction": "ingress",
        "ExplanationCode": "ENI_SG_RULES_MISMATCH",
        "NetworkInterface": {
          "Id": "eni-0a25edef15a6cc08c",
          "Arn": "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a25edef15a6cc08c"
        },
        "SecurityGroups": [
          {
            "Id": "sg-02f0d35a850ba727f",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-02f0d35a850ba727f"
          }
        ],
        "Subnet": {
          "Id": "subnet-004ff41eccb4d1194",
          "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
        },
        "Vpc": {
          "Id": "vpc-f1663d98ad28331c7",
          "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
        }
      }
    ],
    "Tags": []
  }
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS CLI](#) im Reachability Analyzer Guide.

- API-Einheiten finden Sie [DescribeNetworkInsightsAnalyses](#) in der AWS CLI Befehlsreferenz.

describe-network-insights-paths

Das folgende Codebeispiel zeigt die Verwendung `describe-network-insights-paths`.

AWS CLI

Um einen Pfad zu beschreiben

Das folgende `describe-network-insights-paths` Beispiel beschreibt den angegebenen Pfad.

```
aws ec2 describe-network-insights-paths \
  --network-insights-path-ids nip-0b26f224f1d131fa8
```

Ausgabe:

```
{
  "NetworkInsightsPaths": [
    {
      "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
      "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
      "CreateDate": "2021-01-20T22:43:46.933Z",
      "Source": "igw-0797cccdc9d73b0e5",
      "Destination": "i-0495d385ad28331c7",
      "Protocol": "tcp"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS CLI](#) im Reachability Analyzer Guide.

- API-Einheiten finden Sie [DescribeNetworkInsightsPaths](#) in der AWS CLI Befehlsreferenz.

`describe-network-interface-attribute`

Das folgende Codebeispiel zeigt die Verwendung `describe-network-interface-attribute`.

AWS CLI

Um das Attachment-Attribut einer Netzwerkschnittstelle zu beschreiben

Dieser Beispielbefehl beschreibt das `attachment` Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute attachment
```

Ausgabe:

```
{  
  "NetworkInterfaceId": "eni-686ea200",  
  "Attachment": {  
    "Status": "attached",  
    "DeviceIndex": 0,  
    "AttachTime": "2015-05-21T20:02:20.000Z",  
    "InstanceId": "i-1234567890abcdef0",  
    "DeleteOnTermination": true,  
    "AttachmentId": "eni-attach-43348162",  
    "InstanceOwnerId": "123456789012"  
  }  
}
```

Um das Beschreibungsattribut einer Netzwerkschnittstelle zu beschreiben

Dieser Beispielbefehl beschreibt das `description` Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute description
```

Ausgabe:

```
{  
  "NetworkInterfaceId": "eni-686ea200",  
  "Description": {  
    "Value": "My description"  
  }  
}
```

Um das `groupSet` Attribut einer Netzwerkschnittstelle zu beschreiben

Dieser Beispielbefehl beschreibt das `groupSet` Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute groupSet
```

Ausgabe:

```
{  
  "NetworkInterfaceId": "eni-686ea200",  
  "Groups": [  
    {  
      "GroupName": "my-security-group",  
      "GroupId": "sg-903004f8"  
    }  
  ]  
}
```

Um das sourceDestCheck Attribut einer Netzwerkschnittstelle zu beschreiben

Dieser Beispielbefehl beschreibt das sourceDestCheck Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute sourceDestCheck
```

Ausgabe:

```
{  
  "NetworkInterfaceId": "eni-686ea200",  
  "SourceDestCheck": {  
    "Value": true  
  }  
}
```

- API-Einheiten finden Sie [DescribeNetworkInterfaceAttribute](#) unter AWS CLI Befehlsreferenz.

describe-network-interface-permissions

Das folgende Codebeispiel zeigt die Verwendung `describe-network-interface-permissions`.

AWS CLI

Um Ihre Netzwerkschnittstellenberechtigungen zu beschreiben

In diesem Beispiel werden alle Ihre Netzwerkschnittstellenberechtigungen beschrieben.

Befehl:

```
aws ec2 describe-network-interface-permissions
```

Ausgabe:

```
{
  "NetworkInterfacePermissions": [
    {
      "PermissionState": {
        "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-b909511a",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeNetworkInterfacePermissions](#) in der AWS CLI Befehlsreferenz.

describe-network-interfaces

Das folgende Codebeispiel zeigt die Verwendung `describe-network-interfaces`.

AWS CLI

Um Ihre Netzwerkschnittstellen zu beschreiben

Dieses Beispiel beschreibt alle Ihre Netzwerkschnittstellen.

Befehl:

```
aws ec2 describe-network-interfaces
```

Ausgabe:

```
{
  "NetworkInterfaces": [
    {
      "Status": "in-use",
      "MacAddress": "02:2f:8f:b0:cf:75",
      "SourceDestCheck": true,
      "VpcId": "vpc-a01106c2",
      "Description": "my network interface",
      "Association": {
        "PublicIp": "203.0.113.12",
        "AssociationId": "eipassoc-0fbb766a",
        "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
        "IpOwnerId": "123456789012"
      },
      "NetworkInterfaceId": "eni-e5aa89a3",
      "PrivateIpAddresses": [
        {
          "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
          "Association": {
            "PublicIp": "203.0.113.12",
            "AssociationId": "eipassoc-0fbb766a",
            "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
            "IpOwnerId": "123456789012"
          },
          "Primary": true,
          "PrivateIpAddress": "10.0.1.17"
        }
      ],
      "RequesterManaged": false,
      "Ipv6Addresses": [],
      "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
      "AvailabilityZone": "us-east-1d",
      "Attachment": {
        "Status": "attached",
        "DeviceIndex": 1,
        "AttachTime": "2013-11-30T23:36:42.000Z",
        "InstanceId": "i-1234567890abcdef0",
        "DeleteOnTermination": false,
        "AttachmentId": "eni-attach-66c4350a",
        "InstanceOwnerId": "123456789012"
      },
      "Groups": [
```

```
    {
      "GroupName": "default",
      "GroupId": "sg-8637d3e3"
    }
  ],
  "SubnetId": "subnet-b61f49f0",
  "OwnerId": "123456789012",
  "TagSet": [],
  "PrivateIpAddress": "10.0.1.17"
},
{
  "Status": "in-use",
  "MacAddress": "02:58:f5:ef:4b:06",
  "SourceDestCheck": true,
  "VpcId": "vpc-a01106c2",
  "Description": "Primary network interface",
  "Association": {
    "PublicIp": "198.51.100.0",
    "IpOwnerId": "amazon"
  },
  "NetworkInterfaceId": "eni-f9ba99bf",
  "PrivateIpAddresses": [
    {
      "Association": {
        "PublicIp": "198.51.100.0",
        "IpOwnerId": "amazon"
      },
      "Primary": true,
      "PrivateIpAddress": "10.0.1.149"
    }
  ],
  "RequesterManaged": false,
  "Ipv6Addresses": [],
  "AvailabilityZone": "us-east-1d",
  "Attachment": {
    "Status": "attached",
    "DeviceIndex": 0,
    "AttachTime": "2013-11-30T23:35:33.000Z",
    "InstanceId": "i-0598c7d356eba48d7",
    "DeleteOnTermination": true,
    "AttachmentId": "eni-attach-1b9db777",
    "InstanceOwnerId": "123456789012"
  },
  "Groups": [
```

```

        {
            "GroupName": "default",
            "GroupId": "sg-8637d3e3"
        }
    ],
    "SubnetId": "subnet-b61f49f0",
    "OwnerId": "123456789012",
    "TagSet": [],
    "PrivateIpAddress": "10.0.1.149"
}
]
}

```

Dieses Beispiel beschreibt Netzwerkschnittstellen, die ein Tag mit dem Schlüssel Purpose und dem Wert haben Prod.

Befehl:

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

Ausgabe:

```

{
  "NetworkInterfaces": [
    {
      "Status": "available",
      "MacAddress": "12:2c:bd:f9:bf:17",
      "SourceDestCheck": true,
      "VpcId": "vpc-8941ebec",
      "Description": "ProdENI",
      "NetworkInterfaceId": "eni-b9a5ac93",
      "PrivateAddresses": [
        {
          "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
          "Primary": true,
          "PrivateIpAddress": "10.0.1.55"
        },
        {
          "PrivateDnsName": "ip-10-0-1-117.ec2.internal",
          "Primary": false,
          "PrivateIpAddress": "10.0.1.117"
        }
      ]
    }
  ],
}

```

```
    "RequesterManaged": false,
    "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
    "AvailabilityZone": "us-east-1d",
    "Ipv6Addresses": [],
    "Groups": [
      {
        "GroupName": "MySG",
        "GroupId": "sg-905002f5"
      }
    ],
    "SubnetId": "subnet-31d6c219",
    "OwnerId": "123456789012",
    "TagSet": [
      {
        "Value": "Prod",
        "Key": "Purpose"
      }
    ],
    "PrivateIpAddress": "10.0.1.55"
  }
]
```

- API-Einheiten finden Sie [DescribeNetworkInterfaces](#) unter AWS CLI Befehlsreferenz.

describe-placement-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-placement-groups`.

AWS CLI

Um Ihre Platzierungsgruppen zu beschreiben

Dieser Beispielbefehl beschreibt alle Ihre Platzierungsgruppen.

Befehl:

```
aws ec2 describe-placement-groups
```

Ausgabe:

```
{
```



```
"PlacementGroups": [  
  {  
    "GroupName": "my-cluster",  
    "State": "available",  
    "Strategy": "cluster"  
  },  
  ...  
]  
}
```

- API-Einheiten finden Sie [DescribePlacementGroups](#) in der AWS CLI Befehlsreferenz.

describe-prefix-lists

Das folgende Codebeispiel zeigt die Verwendung `describe-prefix-lists`.

AWS CLI

Um Präfixlisten zu beschreiben

In diesem Beispiel werden alle verfügbaren Präfixlisten für die Region aufgeführt.

Befehl:

```
aws ec2 describe-prefix-lists
```

Ausgabe:

```
{  
  "PrefixLists": [  
    {  
      "PrefixListName": "com.amazonaws.us-east-1.s3",  
      "Cidrs": [  
        "54.231.0.0/17"  
      ],  
      "PrefixListId": "pl-63a5400a"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribePrefixLists](#) in der AWS CLI Befehlsreferenz.

describe-principal-id-format

Das folgende Codebeispiel zeigt die Verwendung `describe-principal-id-format`.

AWS CLI

Zur Beschreibung des ID-Formats für IAM Benutzer und Rollen mit aktiviertem Long-ID-Format

Das folgende `describe-principal-id-format` Beispiel beschreibt das ID-Format für den Root-Benutzer, alle IAM Rollen und alle IAM Benutzer, für die das Long-ID-Format aktiviert ist.

```
aws ec2 describe-principal-id-format \  
  --resource instance
```

Ausgabe:

```
{  
  "Principals": [  
    {  
      "Arn": "arn:aws:iam::123456789012:root",  
      "Statuses": [  
        {  
          "Deadline": "2016-12-15T00:00:00.000Z",  
          "Resource": "reservation",  
          "UseLongIds": true  
        },  
        {  
          "Deadline": "2016-12-15T00:00:00.000Z",  
          "Resource": "instance",  
          "UseLongIds": true  
        },  
        {  
          "Deadline": "2016-12-15T00:00:00.000Z",  
          "Resource": "volume",  
          "UseLongIds": true  
        }  
      ],  
    },  
    ...  
  ]  
}
```

- API-Einheiten finden Sie [DescribePrincipalIdFormat](#) unter AWS CLI Befehlsreferenz.

describe-public-ipv4-pools

Das folgende Codebeispiel zeigt die Verwendung `describe-public-ipv4-pools`.

AWS CLI

Um Ihre öffentlichen IPv4 Adresspools zu beschreiben

Im folgenden `describe-public-ipv4-pools` Beispiel werden Details zu den Adresspools angezeigt, die erstellt wurden, als Sie öffentliche IPv4 Adressbereiche mithilfe von Bring Your Own IP Addresses (BYOIP) bereitgestellt haben.

```
aws ec2 describe-public-ipv4-pools
```

Ausgabe:

```
{
  "PublicIpv4Pools": [
    {
      "PoolId": "ipv4pool-ec2-1234567890abcdef0",
      "PoolAddressRanges": [
        {
          "FirstAddress": "203.0.113.0",
          "LastAddress": "203.0.113.255",
          "AddressCount": 256,
          "AvailableAddressCount": 256
        }
      ],
      "TotalAddressCount": 256,
      "TotalAvailableAddressCount": 256
    }
  ]
}
```

- API-Einheiten finden Sie unter [DescribePublicIpv4Pools](#) in der AWS CLI Befehlsreferenz.

describe-regions

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-regions`.

AWS CLI

Beispiel 1: So beschreiben Sie alle von Ihnen aktivierten Regionen

Im folgenden `describe-regions`-Beispiel werden alle Regionen beschrieben, die für Ihr Konto aktiviert sind.

```
aws ec2 describe-regions
```

Ausgabe:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-south-1.amazonaws.com",
      "RegionName": "ap-south-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-3.amazonaws.com",
      "RegionName": "eu-west-3",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-2.amazonaws.com",
      "RegionName": "eu-west-2",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-1.amazonaws.com",
      "RegionName": "eu-west-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
      "RegionName": "ap-northeast-3",
      "OptInStatus": "opt-in-not-required"
    }
  ],
}
```

```
{
  "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
  "RegionName": "ap-northeast-2",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
  "RegionName": "ap-northeast-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.sa-east-1.amazonaws.com",
  "RegionName": "sa-east-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ca-central-1.amazonaws.com",
  "RegionName": "ca-central-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
  "RegionName": "ap-southeast-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
  "RegionName": "ap-southeast-2",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.eu-central-1.amazonaws.com",
  "RegionName": "eu-central-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-east-1.amazonaws.com",
  "RegionName": "us-east-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-east-2.amazonaws.com",
  "RegionName": "us-east-2",
  "OptInStatus": "opt-in-not-required"
}
```

```
    },
    {
      "Endpoint": "ec2.us-west-1.amazonaws.com",
      "RegionName": "us-west-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.us-west-2.amazonaws.com",
      "RegionName": "us-west-2",
      "OptInStatus": "opt-in-not-required"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Regionen und Zonen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: So beschreiben Sie aktivierte Regionen mit einem Endpunkt, dessen Name eine bestimmte Zeichenfolge enthält

Das folgende `describe-regions`-Beispiel beschreibt alle Regionen, die Sie aktiviert haben und deren Endpunkt die Zeichenfolge „us“ enthält.

```
aws ec2 describe-regions \
  --filters "Name=endpoint,Values=*us*"
```

Ausgabe:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.us-east-1.amazonaws.com",
      "RegionName": "us-east-1"
    },
    {
      "Endpoint": "ec2.us-east-2.amazonaws.com",
      "RegionName": "us-east-2"
    },
    {
      "Endpoint": "ec2.us-west-1.amazonaws.com",
      "RegionName": "us-west-1"
    },
    {
      "Endpoint": "ec2.us-west-2.amazonaws.com",
```

```
        "RegionName": "us-west-2"
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Regionen und Zonen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 3: So beschreiben Sie alle Regionen

Das folgende `describe-regions`-Beispiel beschreibt alle verfügbaren Regionen, einschließlich Regionen, die deaktiviert sind.

```
aws ec2 describe-regions \
  --all-regions
```

Ausgabe:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-south-1.amazonaws.com",
      "RegionName": "ap-south-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-3.amazonaws.com",
      "RegionName": "eu-west-3",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-2.amazonaws.com",
      "RegionName": "eu-west-2",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-1.amazonaws.com",
      "RegionName": "eu-west-1",
      "OptInStatus": "opt-in-not-required"
    }
  ]
}
```

```
  },
  {
    "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
    "RegionName": "ap-northeast-3",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.me-south-1.amazonaws.com",
    "RegionName": "me-south-1",
    "OptInStatus": "not-opted-in"
  },
  {
    "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
    "RegionName": "ap-northeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
    "RegionName": "ap-northeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.sa-east-1.amazonaws.com",
    "RegionName": "sa-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ca-central-1.amazonaws.com",
    "RegionName": "ca-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-east-1.amazonaws.com",
    "RegionName": "ap-east-1",
    "OptInStatus": "not-opted-in"
  },
  {
    "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
    "RegionName": "ap-southeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
    "RegionName": "ap-southeast-2",
```



```
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-central-1.amazonaws.com",
    "RegionName": "eu-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-1.amazonaws.com",
    "RegionName": "us-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-2.amazonaws.com",
    "RegionName": "us-east-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-1.amazonaws.com",
    "RegionName": "us-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-2.amazonaws.com",
    "RegionName": "us-west-2",
    "OptInStatus": "opt-in-not-required"
  }
]
}
```

Weitere Informationen finden Sie unter [Regionen und Zonen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 4: So listen Sie nur die Namen der Regionen auf

Im folgenden `describe-regions`-Beispiel wird der `--query`-Parameter verwendet, um die Ausgabe zu filtern und nur die Namen der Regionen als Text zurückzugeben.

```
aws ec2 describe-regions \
  --all-regions \
  --query "Regions[].{Name:RegionName}" \
  --output text
```

Ausgabe:

```
eu-north-1
ap-south-1
eu-west-3
eu-west-2
eu-west-1
ap-northeast-3
ap-northeast-2
me-south-1
ap-northeast-1
sa-east-1
ca-central-1
ap-east-1
ap-southeast-1
ap-southeast-2
eu-central-1
us-east-1
us-east-2
us-west-1
us-west-2
```

Weitere Informationen finden Sie unter [Regionen und Zonen](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeRegions](#) in der AWS CLI Befehlsreferenz.

describe-replace-root-volume-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-replace-root-volume-tasks`.

AWS CLI

Beispiel 1: Um Informationen zu einer bestimmten Aufgabe zum Austausch eines Stammvolumens anzuzeigen

Im folgenden `describe-replace-root-volume-tasks` Beispiel wird die Aufgabe zum Austausch des Stammvolumens `replacevol-0111122223333abcd` beschrieben.

```
aws ec2 describe-replace-root-volume-tasks \
  --replace-root-volume-task-ids replacevol-0111122223333abcd
```

Ausgabe:

```
{
```

```

    "ReplaceRootVolumeTasks": [
      {
        "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
        "Tags": [],
        "InstanceId": "i-0123456789abcdefa",
        "TaskState": "succeeded",
        "StartTime": "2022-03-14T15:16:28Z",
        "CompleteTime": "2022-03-14T15:16:52Z"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Ersetzen eines Root-Volumes](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Beispiel 2: Um Informationen zu allen Aufgaben zum Austausch von Root-Volumes für eine bestimmte Instance anzuzeigen

Im folgenden `describe-replace-root-volume-tasks` Beispiel werden alle Aufgaben zum Austausch des Root-Volumes beschrieben, zum Beispiel `i-0123456789abcdefa`.

```

aws ec2 describe-replace-root-volume-tasks \
  --filters Name=instance-id,Values=i-0123456789abcdefa

```

Ausgabe:

```

{
  "ReplaceRootVolumeTasks": [
    {
      "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
      "Tags": [],
      "InstanceId": "i-0123456789abcdefa",
      "TaskState": "succeeded",
      "StartTime": "2022-03-14T15:06:38Z",
      "CompleteTime": "2022-03-14T15:07:03Z"
    },
    {
      "ReplaceRootVolumeTaskId": "replacevol-0444455555555abcd",
      "Tags": [],
      "InstanceId": "i-0123456789abcdefa",
      "TaskState": "succeeded",
      "StartTime": "2022-03-14T15:16:28Z",

```

```
        "CompleteTime": "2022-03-14T15:16:52Z"
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Ersetzen eines Root-Volumes](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReplaceRootVolumeTasks](#) in der AWS CLI Befehlsreferenz.

describe-reserved-instances-listings

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-instances-listings`.

AWS CLI

Um eine Reserved Instance-Liste zu beschreiben

Im folgenden `describe-reserved-instances-listings` Beispiel werden Informationen über die angegebene Reserved Instance-Liste abgerufen.

```
aws ec2 describe-reserved-instances-listings \
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DescribeReservedInstancesListings AWS CLI](#) Befehlsreferenz.

describe-reserved-instances-modifications

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-instances-modifications`.

AWS CLI

Um die Änderungen an Reserved Instances zu beschreiben

Dieser Beispielbefehl beschreibt alle Änderungsanfragen für Reserved Instances, die für Ihr Konto eingereicht wurden.

Befehl:

aws ec2 describe-reserved-instances-modifications

Ausgabe:

```
{
  "ReservedInstancesModifications": [
    {
      "Status": "fulfilled",
      "ModificationResults": [
        {
          "ReservedInstancesId": "93bbbca2-62f1-4d9d-b225-16bada29e6c7",
          "TargetConfiguration": {
            "AvailabilityZone": "us-east-1b",
            "InstanceType": "m1.large",
            "InstanceCount": 3
          }
        },
        {
          "ReservedInstancesId": "1ba8e2e3-aabb-46c3-bcf5-3fe2fda922e6",
          "TargetConfiguration": {
            "AvailabilityZone": "us-east-1d",
            "InstanceType": "m1.xlarge",
            "InstanceCount": 1
          }
        }
      ],
      "EffectiveDate": "2015-08-12T17:00:00.000Z",
      "CreateDate": "2015-08-12T17:52:52.630Z",
      "UpdateDate": "2015-08-12T18:08:06.698Z",
      "ClientToken": "c9adb218-3222-4889-8216-0cf0e52dc37e",
      "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687",
      "ReservedInstancesIds": [
        {
          "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342f5bd7c02"
        }
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeReservedInstancesModifications](#) unter AWS CLI Befehlsreferenz.

describe-reserved-instances-offerings

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-instances-offerings`.

AWS CLI

Um die Angebote von Reserved Instances zu beschreiben

Dieser Beispielbefehl beschreibt alle Reserved Instances, die in der Region zum Kauf angeboten werden.

Befehl:

```
aws ec2 describe-reserved-instances-offerings
```

Ausgabe:

```
{
  "ReservedInstancesOfferings": [
    {
      "OfferingType": "Partial Upfront",
      "AvailabilityZone": "us-east-1b",
      "InstanceTenancy": "default",
      "PricingDetails": [],
      "ProductDescription": "Red Hat Enterprise Linux",
      "UsagePrice": 0.0,
      "RecurringCharges": [
        {
          "Amount": 0.088,
          "Frequency": "Hourly"
        }
      ],
      "Marketplace": false,
      "CurrencyCode": "USD",
      "FixedPrice": 631.0,
      "Duration": 94608000,
      "ReservedInstancesOfferingId": "9a06095a-bdc6-47fe-a94a-2a382f016040",
      "InstanceType": "c1.medium"
    },
  ],
}
```

```

    {
      "OfferingType": "PartialUpfront",
      "AvailabilityZone": "us-east-1b",
      "InstanceTenancy": "default",
      "PricingDetails": [],
      "ProductDescription": "Linux/UNIX",
      "UsagePrice": 0.0,
      "RecurringCharges": [
        {
          "Amount": 0.028,
          "Frequency": "Hourly"
        }
      ],
      "Marketplace": false,
      "CurrencyCode": "USD",
      "FixedPrice": 631.0,
      "Duration": 94608000,
      "ReservedInstancesOfferingId": "bfbefc6c-0d10-418d-b144-7258578d329d",
      "InstanceType": "c1.medium"
    },
    ...
  }

```

Um Ihre Reserved Instance-Angebote mithilfe von Optionen zu beschreiben

In diesem Beispiel werden Reserved Instances aufgeführt, die von AWS mit den folgenden Spezifikationen angeboten werden: t1.micro-Instance-Typen, Windows-Produkte (AmazonVPC) und Angebote für hohe Auslastung.

Befehl:

```

aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-
type "t1.micro" --product-description "Windows (Amazon VPC)" --offering-type "no
upfront"

```

Ausgabe:

```

{
  "ReservedInstancesOfferings": [
    {
      "OfferingType": "No Upfront",
      "AvailabilityZone": "us-east-1b",
      "InstanceTenancy": "default",

```

```

    "PricingDetails": [],
    "ProductDescription": "Windows",
    "UsagePrice": 0.0,
    "RecurringCharges": [
      {
        "Amount": 0.015,
        "Frequency": "Hourly"
      }
    ],
    "Marketplace": false,
    "CurrencyCode": "USD",
    "FixedPrice": 0.0,
    "Duration": 31536000,
    "ReservedInstancesOfferingId": "c48ab04c-fe69-4f94-8e39-a23842292823",
    "InstanceType": "t1.micro"
  },
  ...
  {
    "OfferingType": "No Upfront",
    "AvailabilityZone": "us-east-1d",
    "InstanceTenancy": "default",
    "PricingDetails": [],
    "ProductDescription": "Windows (Amazon VPC)",
    "UsagePrice": 0.0,
    "RecurringCharges": [
      {
        "Amount": 0.015,
        "Frequency": "Hourly"
      }
    ],
    "Marketplace": false,
    "CurrencyCode": "USD",
    "FixedPrice": 0.0,
    "Duration": 31536000,
    "ReservedInstancesOfferingId": "3a98bf7d-2123-42d4-b4f5-8dbec4b06dc6",
    "InstanceType": "t1.micro"
  }
]
}

```

- API-Einheiten finden Sie [DescribeReservedInstancesOfferings](#) in der AWS CLI Befehlsreferenz.

describe-reserved-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-instances`.

AWS CLI

Um Ihre Reserved Instances zu beschreiben

Dieser Beispielbefehl beschreibt die Reserved Instances, die Sie besitzen.

Befehl:

```
aws ec2 describe-reserved-instances
```

Ausgabe:

```
{
  "ReservedInstances": [
    {
      "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342fexample",
      "OfferingType": "No Upfront",
      "AvailabilityZone": "us-west-1c",
      "End": "2016-08-14T21:34:34.000Z",
      "ProductDescription": "Linux/UNIX",
      "UsagePrice": 0.00,
      "RecurringCharges": [
        {
          "Amount": 0.104,
          "Frequency": "Hourly"
        }
      ],
      "Start": "2015-08-15T21:34:35.086Z",
      "State": "active",
      "FixedPrice": 0.0,
      "CurrencyCode": "USD",
      "Duration": 31536000,
      "InstanceTenancy": "default",
      "InstanceType": "m3.medium",
      "InstanceCount": 2
    },
    ...
  ]
}
```

Um Ihre Reserved Instances mithilfe von Filtern zu beschreiben

In diesem Beispiel wird die Antwort so gefiltert, dass sie nur dreijährige t2.micro UNIX Linux/Reserved Instances in us-west-1c enthält.

Befehl:

```
aws ec2 describe-reserved-instances --  
filters Name=duration,Values=94608000 Name=instance-  
type,Values=t2.micro Name=product-description,Values=Linux/UNIX Name=availability-  
zone,Values=us-east-1e
```

Ausgabe:

```
{  
  "ReservedInstances": [  
    {  
      "ReservedInstancesId": "f127bd27-edb7-44c9-a0eb-0d7e09259af0",  
      "OfferingType": "All Upfront",  
      "AvailabilityZone": "us-east-1e",  
      "End": "2018-03-26T21:34:34.000Z",  
      "ProductDescription": "Linux/UNIX",  
      "UsagePrice": 0.00,  
      "RecurringCharges": [],  
      "Start": "2015-03-27T21:34:35.848Z",  
      "State": "active",  
      "FixedPrice": 151.0,  
      "CurrencyCode": "USD",  
      "Duration": 94608000,  
      "InstanceTenancy": "default",  
      "InstanceType": "t2.micro",  
      "InstanceCount": 1  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Using Amazon EC2 Instances](#) im AWS Command Line Interface User Guide.

- API-Einheiten finden Sie [DescribeReservedInstances](#) unter AWS CLI Befehlsreferenz.

describe-route-tables

Das folgende Codebeispiel zeigt die Verwendung `describe-route-tables`.

AWS CLI

Um Ihre Routentabellen zu beschreiben

Im folgenden `describe-route-tables` Beispiel werden die Details zu Ihren Routentabellen abgerufen

```
aws ec2 describe-route-tables
```

Ausgabe:

```
{
  "RouteTables": [
    {
      "Associations": [
        {
          "Main": true,
          "RouteTableAssociationId": "rtbassoc-0df3f54e06EXAMPLE",
          "RouteTableId": "rtb-09ba434c1bEXAMPLE"
        }
      ],
      "PropagatingVgws": [],
      "RouteTableId": "rtb-09ba434c1bEXAMPLE",
      "Routes": [
        {
          "DestinationCidrBlock": "10.0.0.0/16",
          "GatewayId": "local",
          "Origin": "CreateRouteTable",
          "State": "active"
        },
        {
          "DestinationCidrBlock": "0.0.0.0/0",
          "NatGatewayId": "nat-06c018cbd8EXAMPLE",
          "Origin": "CreateRoute",
          "State": "blackhole"
        }
      ],
      "Tags": [],
      "VpcId": "vpc-0065acced4EXAMPLE",
    }
  ]
}
```

```
    "OwnerId": "111122223333"
  },
  {
    "Associations": [
      {
        "Main": true,
        "RouteTableAssociationId": "rtbassoc-9EXAMPLE",
        "RouteTableId": "rtb-a1eec7de"
      }
    ],
    "PropagatingVgws": [],
    "RouteTableId": "rtb-a1eec7de",
    "Routes": [
      {
        "DestinationCidrBlock": "172.31.0.0/16",
        "GatewayId": "local",
        "Origin": "CreateRouteTable",
        "State": "active"
      },
      {
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": "igw-fEXAMPLE",
        "Origin": "CreateRoute",
        "State": "active"
      }
    ],
    "Tags": [],
    "VpcId": "vpc-3EXAMPLE",
    "OwnerId": "111122223333"
  },
  {
    "Associations": [
      {
        "Main": false,
        "RouteTableAssociationId": "rtbassoc-0b100c28b2EXAMPLE",
        "RouteTableId": "rtb-07a98f76e5EXAMPLE",
        "SubnetId": "subnet-0d3d002af8EXAMPLE"
      }
    ],
    "PropagatingVgws": [],
    "RouteTableId": "rtb-07a98f76e5EXAMPLE",
    "Routes": [
      {
        "DestinationCidrBlock": "10.0.0.0/16",
```

```

        "GatewayId": "local",
        "Origin": "CreateRouteTable",
        "State": "active"
    },
    {
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": "igw-06cf664d80EXAMPLE",
        "Origin": "CreateRoute",
        "State": "active"
    }
],
"Tags": [],
"VpcId": "vpc-0065acced4EXAMPLE",
"OwnerId": "111122223333"
}
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Routentabellen](#) im AWS VPCBenutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeRouteTables](#) unter AWS CLI Befehlsreferenz.

describe-scheduled-instance-availability

Das folgende Codebeispiel zeigt die Verwendung `describe-scheduled-instance-availability`.

AWS CLI

Um einen verfügbaren Zeitplan zu beschreiben

Dieses Beispiel beschreibt einen Zeitplan, der jede Woche am Sonntag beginnt und am angegebenen Datum beginnt.

Befehl:

```

aws ec2 describe-scheduled-instance-availability --
recurrence Frequency=Weekly,Interval=1,OccurrenceDays=[1] --first-slot-start-time-
range EarliestTime=2016-01-31T00:00:00Z,LatestTime=2016-01-31T04:00:00Z

```

Ausgabe:

```
{
  "ScheduledInstanceAvailabilitySet": [
    {
      "AvailabilityZone": "us-west-2b",
      "TotalScheduledInstanceHours": 1219,
      "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
      "MinTermDurationInDays": 366,
      "AvailableInstanceCount": 20,
      "Recurrence": {
        "OccurrenceDaySet": [
          1
        ],
        "Interval": 1,
        "Frequency": "Weekly",
        "OccurrenceRelativeToEnd": false
      },
      "Platform": "Linux/UNIX",
      "FirstSlotStartTime": "2016-01-31T00:00:00Z",
      "MaxTermDurationInDays": 366,
      "SlotDurationInHours": 23,
      "NetworkPlatform": "EC2-VPC",
      "InstanceType": "c4.large",
      "HourlyPrice": "0.095"
    },
    ...
  ]
}
```

Um die Ergebnisse einzugrenzen, können Sie Filter hinzufügen, die das Betriebssystem, das Netzwerk und den Instanztyp angeben.

Befehl:

```
--filters Name=Plattform, Werte=Linux/Name=Netzwerkplattform, Werte= - Name=Instanztyp,
Werte=C4.large UNIX EC2 VPC
```

- API-Einheiten finden Sie in der [DescribeScheduledInstanceAvailability AWS CLI-Befehlsreferenz](#).

describe-scheduled-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-scheduled-instances`.

AWS CLI

Um Ihre geplanten Instances zu beschreiben

Dieses Beispiel beschreibt die angegebene Scheduled Instance.

Befehl:

```
aws ec2 describe-scheduled-instances --scheduled-instance-ids sci-1234-1234-1234-1234-123456789012
```

Ausgabe:

```
{
  "ScheduledInstanceSet": [
    {
      "AvailabilityZone": "us-west-2b",
      "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
      "HourlyPrice": "0.095",
      "CreateDate": "2016-01-25T21:43:38.612Z",
      "Recurrence": {
        "OccurrenceDaySet": [
          1
        ],
        "Interval": 1,
        "Frequency": "Weekly",
        "OccurrenceRelativeToEnd": false,
        "OccurrenceUnit": ""
      },
      "Platform": "Linux/UNIX",
      "TermEndDate": "2017-01-31T09:00:00Z",
      "InstanceCount": 1,
      "SlotDurationInHours": 32,
      "TermStartDate": "2016-01-31T09:00:00Z",
      "NetworkPlatform": "EC2-VPC",
      "TotalScheduledInstanceHours": 1696,
      "NextSlotStartTime": "2016-01-31T09:00:00Z",
      "InstanceType": "c4.large"
    }
  ]
}
```

Dieses Beispiel beschreibt alle Ihre geplanten Instances.

Befehl:

```
aws ec2 describe-scheduled-instances
```

- API-Einheiten finden Sie [DescribeScheduledInstances](#) in der AWS CLI Befehlsreferenz.

describe-security-group-references

Das folgende Codebeispiel zeigt die Verwendung `describe-security-group-references`.

AWS CLI

Um Verweise auf Sicherheitsgruppen zu beschreiben

In diesem Beispiel werden die Sicherheitsgruppenreferenzen für `beschreibensg-bbbb2222`. Die Antwort weist darauf hin, dass eine Sicherheitsgruppe `sg-bbbb2222` in auf die Sicherheitsgruppe `vpc-aaaaaaaa` verweist.

Befehl:

```
aws ec2 describe-security-group-references --group-id sg-bbbbbb22222
```

Ausgabe:

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeSecurityGroupReferences](#) in der AWS CLI Befehlsreferenz.

describe-security-group-rules

Das folgende Codebeispiel zeigt die Verwendung `describe-security-group-rules`.

AWS CLI

Beispiel 1: Um die Sicherheitsgruppenregeln für eine Sicherheitsgruppe zu beschreiben

Das folgende `describe-security-group-rules` Beispiel beschreibt die Sicherheitsgruppenregeln einer angegebenen Sicherheitsgruppe. Verwenden Sie `filters` diese Option, um die Ergebnisse auf eine bestimmte Sicherheitsgruppe zu beschränken.

```
aws ec2 describe-security-group-rules \
  --filters Name="group-id",Values="sg-1234567890abcdef0"
```

Ausgabe:

```
{
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-abcdef01234567890",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "111122223333",
      "IsEgress": false,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "ReferencedGroupInfo": {
        "GroupId": "sg-1234567890abcdef0",
        "UserId": "111122223333"
      },
      "Tags": []
    },
    {
      "SecurityGroupRuleId": "sgr-bcdef01234567890a",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "111122223333",
      "IsEgress": true,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv6": "::/0",
      "Tags": []
    },
    {
      "SecurityGroupRuleId": "sgr-cdef01234567890ab",
      "GroupId": "sg-1234567890abcdef0",
```

```

    "GroupOwnerId": "111122223333",
    "IsEgress": true,
    "IpProtocol": "-1",
    "FromPort": -1,
    "ToPort": -1,
    "CidrIpv4": "0.0.0.0/0",
    "Tags": []
  }
]
}

```

Beispiel 2: Um eine Sicherheitsgruppenregel zu beschreiben

Das folgende `describe-security-group-rules` Beispiel beschreibt die angegebene Sicherheitsgruppenregel.

```

aws ec2 describe-security-group-rules \
  --security-group-rule-ids sgr-cdef01234567890ab

```

Ausgabe:

```

{
  "SecurityGroupRules": [
    {
      "SecurityGroupId": "sgr-cdef01234567890ab",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "111122223333",
      "IsEgress": true,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv4": "0.0.0.0/0",
      "Tags": []
    }
  ]
}

```

Weitere Informationen finden Sie unter [Regeln für Sicherheitsgruppen](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSecurityGroupRules](#) in der AWS CLI Befehlsreferenz.

describe-security-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-security-groups`.

AWS CLI

Beispiel 1: So beschreiben Sie eine Sicherheitsgruppe

Das folgende `describe-security-groups`-Beispiel beschreibt die angegebene Sicherheitsgruppe.

```
aws ec2 describe-security-groups \
  --group-ids sg-903004f8
```

Ausgabe:

```
{
  "SecurityGroups": [
    {
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "UserIdGroupPairs": [],
          "PrefixListIds": []
        }
      ],
      "Description": "My security group",
      "Tags": [
        {
          "Value": "SG1",
          "Key": "Name"
        }
      ],
      "IpPermissions": [
        {
          "IpProtocol": "-1",
          "IpRanges": [],
          "UserIdGroupPairs": [
```

```

        {
            "UserId": "123456789012",
            "GroupId": "sg-903004f8"
        }
    ],
    "PrefixListIds": []
},
{
    "PrefixListIds": [],
    "FromPort": 22,
    "IpRanges": [
        {
            "Description": "Access from NY office",
            "CidrIp": "203.0.113.0/24"
        }
    ],
    "ToPort": 22,
    "IpProtocol": "tcp",
    "UserIdGroupPairs": []
}
],
"GroupName": "MySecurityGroup",
"VpcId": "vpc-1a2b3c4d",
"OwnerId": "123456789012",
"GroupId": "sg-903004f8",
}
]
}

```

Beispiel 2: So beschreiben Sie Sicherheitsgruppen, die bestimmte Regeln haben

Im folgenden `describe-security-groups` Beispiel werden Filter verwendet, um die Ergebnisse auf Sicherheitsgruppen zu beschränken, die eine Regel haben, die SSH Datenverkehr zulässt (Port 22), und eine Regel, die Datenverkehr von allen Adressen zulässt (`0.0.0.0/0`). Im Beispiel wird der `--query`-Parameter verwendet, um nur die Namen der Sicherheitsgruppen anzuzeigen. Sicherheitsgruppen müssen mit allen Filtern übereinstimmen, die in den Ergebnissen zurückgegeben werden. Eine einzige Regel muss jedoch nicht mit allen Filtern übereinstimmen. Die Ausgabe gibt beispielsweise eine Sicherheitsgruppe mit einer Regel zurück, die SSH Datenverkehr von einer bestimmten IP-Adresse zulässt, und einer anderen Regel, die HTTP Datenverkehr von allen Adressen zulässt.

```
aws ec2 describe-security-groups \
```

```
--filters Name=ip-permission.from-port,Values=22 Name=ip-permission.to-  
port,Values=22 Name=ip-permission.cidr,Values='0.0.0.0/0' \  
--query "SecurityGroups[*].[GroupName]" \  
--output text
```

Ausgabe:

```
default  
my-security-group  
web-servers  
launch-wizard-1
```

Beispiel 3: So beschreiben Sie Sicherheitsgruppen anhand von Tags

Im folgenden `describe-security-groups`-Beispiel werden Filter verwendet, um die Ergebnisse auf Sicherheitsgruppen zu beschränken, die `test` im Namen der Sicherheitsgruppe enthalten und die das Tag `Test=To-delete` haben. Das Beispiel verwendet den `--query` Parameter, um nur die Namen und die IDs der Sicherheitsgruppen anzuzeigen.

```
aws ec2 describe-security-groups \  
--filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \  
--query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

Ausgabe:

```
[  
  {  
    "Name": "testfornewinstance",  
    "ID": "sg-33bb22aa"  
  },  
  {  
    "Name": "newgrouptest",  
    "ID": "sg-1a2b3c4d"  
  }  
]
```

Weitere Beispiele für die Verwendung von Tagfiltern finden Sie unter [Arbeiten mit Tags](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSecurityGroups](#) in der AWS CLI Befehlsreferenz.

describe-snapshot-attribute

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshot-attribute`.

AWS CLI

Um die Snapshot-Attribute für einen Snapshot zu beschreiben

Das folgende `describe-snapshot-attribute` Beispiel listet die Konten auf, mit denen ein Snapshot gemeinsam genutzt wird.

```
aws ec2 describe-snapshot-attribute \
  --snapshot-id snap-01234567890abcdef \
  --attribute createVolumePermission
```

Ausgabe:

```
{
  "SnapshotId": "snap-01234567890abcdef",
  "CreateVolumePermissions": [
    {
      "UserId": "123456789012"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Einen EBS Amazon-Snapshot teilen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSnapshotAttribute](#) in der AWS CLI Befehlsreferenz.

describe-snapshot-tier-status

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshot-tier-status`.

AWS CLI

Um Archivinformationen zu einem archivierten Snapshot anzuzeigen

Das folgende `describe-snapshot-tier-status` Beispiel enthält Archivinformationen zu einem archivierten Snapshot.

```
aws ec2 describe-snapshot-tier-status \
  --filters "Name=snapshot-id, Values=snap-01234567890abcdef"
```

Ausgabe:

```
{
  "SnapshotTierStatuses": [
    {
      "Status": "completed",
      "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
      "LastTieringProgress": 100,
      "Tags": [],
      "VolumeId": "vol-01234567890abcdef",
      "LastTieringOperationState": "archival-completed",
      "StorageTier": "archive",
      "OwnerId": "123456789012",
      "SnapshotId": "snap-01234567890abcdef",
      "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Archivierte Snapshots anzeigen](#) im Benutzerhandbuch für Amazon Elastic Compute Cloud.

- API-Einheiten finden Sie unter [DescribeSnapshotTierStatus AWS CLI Befehlsreferenz](#).

describe-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshots`.

AWS CLI

Beispiel 1: So beschreiben Sie einen Snapshot

Das folgende `describe-snapshots`-Beispiel beschreibt den angegebenen Snapshot.

```
aws ec2 describe-snapshots \
  --snapshot-ids snap-1234567890abcdef0
```

Ausgabe:

```
{
  "Snapshots": [
    {
      "Description": "This is my snapshot",
      "Encrypted": false,
      "VolumeId": "vol-049df61146c4d7901",
      "State": "completed",
      "VolumeSize": 8,
      "StartTime": "2019-02-28T21:28:32.000Z",
      "Progress": "100%",
      "OwnerId": "012345678910",
      "SnapshotId": "snap-01234567890abcdef",
      "Tags": [
        {
          "Key": "Stack",
          "Value": "test"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon EBS Snapshots](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: So beschreiben Sie Snapshots basierend auf Filtern

Im folgenden `describe-snapshots` Beispiel werden Filter verwendet, um die Ergebnisse auf Snapshots zu beschränken, die Ihrem AWS Konto gehören und sich im `pending` Bundesstaat befinden. Das Beispiel verwendet den `--query` Parameter, um nur den Snapshot IDs und die Uhrzeit anzuzeigen, zu der der Snapshot gestartet wurde.

```
aws ec2 describe-snapshots \
  --owner-ids self \
  --filters Name=status,Values=pending \
  --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"
```

Ausgabe:

```
[
  {
    "ID": "snap-1234567890abcdef0",
```



```

    "Time": "2019-08-04T12:48:18.000Z"
  },
  {
    "ID": "snap-066877671789bd71b",
    "Time": "2019-08-04T02:45:16.000Z"
  },
  ...
]

```

Im folgenden `describe-snapshots`-Beispiel werden Filter verwendet, um die Ergebnisse auf Snapshots zu beschränken, die aus dem angegebenen Volume erstellt wurden. Das Beispiel verwendet den `--query` Parameter, um nur den Snapshot anzuzeigenIDs.

```

aws ec2 describe-snapshots \
  --filters Name=volume-id,Values=049df61146c4d7901 \
  --query "Snapshots[*].[SnapshotId]" \
  --output text

```

Ausgabe:

```

snap-1234567890abcdef0
snap-08637175a712c3fb9
...

```

Weitere Beispiele für die Verwendung von Filtern finden Sie unter [Ressourcen auflisten und filtern](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 3: So beschreiben Sie Snapshots auf der Grundlage von Tags

Im folgenden `describe-snapshots`-Beispiel werden Tag-Filter verwendet, um die Ergebnisse auf Snapshots zu beschränken, die das Tag `Stack=Prod` enthalten.

```

aws ec2 describe-snapshots \
  --filters Name=tag:Stack,Values=prod

```

Ein Beispiel für die Ausgabe von `describe-snapshots` finden Sie in Beispiel 1.

Weitere Beispiele für die Verwendung von Tagfiltern finden Sie unter [Arbeiten mit Tags](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 4: So beschreiben Sie Snapshots anhand des Alters

Im folgenden `describe-snapshots` Beispiel werden JMESPath Ausdrücke verwendet, um alle Schnappschüsse zu beschreiben, die von Ihrem AWS Konto vor dem angegebenen Datum erstellt wurden. Es wird nur der Snapshot IDs angezeigt.

```
aws ec2 describe-snapshots \  
  --owner-ids 012345678910 \  
  --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

Weitere Beispiele für die Verwendung von Filtern finden Sie unter [Ressourcen auflisten und filtern](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 5: So zeigen Sie nur archivierte Snapshots an

Im folgenden `describe-snapshots`-Beispiel werden ausschließlich Snapshots aufgeführt, die auf der Archivstufe gespeichert sind.

```
aws ec2 describe-snapshots \  
  --filters "Name=storage-tier,Values=archive"
```

Ausgabe:

```
{  
  "Snapshots": [  
    {  
      "Description": "Snap A",  
      "Encrypted": false,  
      "VolumeId": "vol-01234567890aaaaaa",  
      "State": "completed",  
      "VolumeSize": 8,  
      "StartTime": "2021-09-07T21:00:00.000Z",  
      "Progress": "100%",  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-01234567890aaaaaa",  
      "StorageTier": "archive",  
      "Tags": []  
    },  
  ]  
}
```

Weitere Informationen finden Sie unter [Archivierte Snapshots anzeigen](#) im Benutzerhandbuch für Amazon Elastic Compute Cloud.

- API-Einheiten finden Sie [DescribeSnapshots](#) in der AWS CLI Befehlsreferenz.

describe-spot-datafeed-subscription

Das folgende Codebeispiel zeigt die Verwendung `describe-spot-datafeed-subscription`.

AWS CLI

Um das Spot-Instance-Datenfeed-Abonnement für ein Konto zu beschreiben

Dieser Beispielbefehl beschreibt den Datenfeed für das Konto.

Befehl:

```
aws ec2 describe-spot-datafeed-subscription
```

Ausgabe:

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "123456789012",
    "Prefix": "spotdata",
    "Bucket": "my-s3-bucket",
    "State": "Active"
  }
}
```

- API-Einheiten finden Sie [DescribeSpotDatafeedSubscription](#) in der AWS CLI Befehlsreferenz.

describe-spot-fleet-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-spot-fleet-instances`.

AWS CLI

Um die Spot-Instances zu beschreiben, die einer Spot-Flotte zugeordnet sind

Dieser Beispielbefehl listet die Spot-Instances auf, die der angegebenen Spot-Flotte zugeordnet sind.

Befehl:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Ausgabe:

```
{
  "ActiveInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "InstanceType": "m3.medium",
      "SpotInstanceRequestId": "sir-08b93456"
    },
    ...
  ],
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- API-Einheiten finden Sie [DescribeSpotFleetInstances](#) unter AWS CLI Befehlsreferenz.

describe-spot-fleet-request-history

Das folgende Codebeispiel zeigt die Verwendung `describe-spot-fleet-request-history`.

AWS CLI

Um die Geschichte der Spot-Flotte zu beschreiben

Dieser Beispielbefehl gibt den Verlauf für die angegebene Spot-Flotte ab dem angegebenen Zeitpunkt zurück.

Befehl:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

Die folgende Beispielausgabe zeigt die erfolgreichen Starts von zwei Spot-Instances für die Spot-Flotte.

Ausgabe:

```
{
  "HistoryRecords": [
```

```

    {
      "Timestamp": "2015-05-26T23:17:20.697Z",
      "EventInformation": {
        "EventSubType": "submitted"
      },
      "EventType": "fleetRequestChange"
    },
    {
      "Timestamp": "2015-05-26T23:17:20.873Z",
      "EventInformation": {
        "EventSubType": "active"
      },
      "EventType": "fleetRequestChange"
    },
    {
      "Timestamp": "2015-05-26T23:21:21.712Z",
      "EventInformation": {
        "InstanceId": "i-1234567890abcdef0",
        "EventSubType": "launched"
      },
      "EventType": "instanceChange"
    },
    {
      "Timestamp": "2015-05-26T23:21:21.816Z",
      "EventInformation": {
        "InstanceId": "i-1234567890abcdef1",
        "EventSubType": "launched"
      },
      "EventType": "instanceChange"
    }
  ],
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
  "NextToken": "CpHNsscimcV5oH7bSbub03CI2Qms5+ypNpNm
+53MNlR0YcXAKp0xFlfKf91yVxSExmbtma3awYxMFzNA663ZskT0AHTJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauVg
+P+TfG1WxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
  "StartTime": "2015-05-26T00:00:00Z"
}

```

- API-Einheiten finden Sie [DescribeSpotFleetRequestHistory](#) unter AWS CLI Befehlsreferenz.

describe-spot-fleet-requests

Das folgende Codebeispiel zeigt die Verwendung `describe-spot-fleet-requests`.

AWS CLI

Um Ihre Spot-Flottenanfragen zu beschreiben

In diesem Beispiel werden alle Ihre Spot-Flottenanfragen beschrieben.

Befehl:

```
aws ec2 describe-spot-fleet-requests
```

Ausgabe:

```
{
  "SpotFleetRequestConfigs": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "SpotFleetRequestConfig": {
        "TargetCapacity": 20,
        "LaunchSpecifications": [
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ],
            "InstanceType": "cc2.8xlarge",
            "ImageId": "ami-1a2b3c4d"
          },
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

        ],
        "InstanceType": "r3.8xlarge",
        "ImageId": "ami-1a2b3c4d"
    }
  ],
  "SpotPrice": "0.05",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
},
"SpotFleetRequestState": "active"
},
{
  "SpotFleetRequestId": "sfr-306341ed-9739-402e-881b-ce47bEXAMPLE",
  "SpotFleetRequestConfig": {
    "TargetCapacity": 20,
    "LaunchSpecifications": [
      {
        "EbsOptimized": false,
        "NetworkInterfaces": [
          {
            "SubnetId": "subnet-6e7f829e",
            "DeviceIndex": 0,
            "DeleteOnTermination": false,
            "AssociatePublicIpAddress": true,
            "SecondaryPrivateIpAddressCount": 0
          }
        ],
        "InstanceType": "m3.medium",
        "ImageId": "ami-1a2b3c4d"
      }
    ],
    "SpotPrice": "0.05",
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
  },
  "SpotFleetRequestState": "active"
}
]
}

```

Um eine Spot-Flottenanfrage zu beschreiben

Dieses Beispiel beschreibt die angegebene Spot-Flottenanfrage.

Befehl:

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Ausgabe:

```
{
  "SpotFleetRequestConfigs": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "SpotFleetRequestConfig": {
        "TargetCapacity": 20,
        "LaunchSpecifications": [
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ],
            "InstanceType": "cc2.8xlarge",
            "ImageId": "ami-1a2b3c4d"
          },
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ],
            "InstanceType": "r3.8xlarge",
            "ImageId": "ami-1a2b3c4d"
          }
        ],
        "SpotPrice": "0.05",
        "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
      }
    }
  ]
}
```



```
    },
    "SpotFleetRequestState": "active"
  }
]
}
```

- API-Einheiten finden Sie [DescribeSpotFleetRequests](#) in der AWS CLI Befehlsreferenz.

describe-spot-instance-requests

Das folgende Codebeispiel zeigt die Verwendung `describe-spot-instance-requests`.

AWS CLI

Beispiel 1: Um eine Spot-Instance-Anfrage zu beschreiben

Das folgende `describe-spot-instance-requests` Beispiel beschreibt die angegebene Spot-Instance-Anfrage.

```
aws ec2 describe-spot-instance-requests \
  --spot-instance-request-ids sir-08b93456
```

Ausgabe:

```
{
  "SpotInstanceRequests": [
    {
      "CreateTime": "2018-04-30T18:14:55.000Z",
      "InstanceId": "i-1234567890abcdef1",
      "LaunchSpecification": {
        "InstanceType": "t2.micro",
        "ImageId": "ami-003634241a8fcdec0",
        "KeyName": "my-key-pair",
        "SecurityGroups": [
          {
            "GroupName": "default",
            "GroupId": "sg-e38f24a7"
          }
        ],
        "BlockDeviceMappings": [
          {
```

```
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "DeleteOnTermination": true,
            "SnapshotId": "snap-0e54a519c999adbbd",
            "VolumeSize": 8,
            "VolumeType": "standard",
            "Encrypted": false
        }
    ],
    "NetworkInterfaces": [
        {
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "SubnetId": "subnet-049df61146c4d7901"
        }
    ],
    "Placement": {
        "AvailabilityZone": "us-east-2b",
        "Tenancy": "default"
    },
    "Monitoring": {
        "Enabled": false
    }
},
"LaunchedAvailabilityZone": "us-east-2b",
"ProductDescription": "Linux/UNIX",
"SpotInstanceRequestId": "sir-08b93456",
"SpotPrice": "0.010000"
"State": "active",
"Status": {
    "Code": "fulfilled",
    "Message": "Your Spot request is fulfilled.",
    "UpdateTime": "2018-04-30T18:16:21.000Z"
},
"Tags": [],
"Type": "one-time",
"InstanceInterruptionBehavior": "terminate"
}
]
```

Beispiel 2: Um Spot-Instance-Anfragen auf der Grundlage von Filtern zu beschreiben

Im folgenden `describe-spot-instance-requests` Beispiel werden Filter verwendet, um die Ergebnisse auf Spot-Instance-Anfragen mit dem angegebenen Instance-Typ in der angegebenen Availability Zone zu beschränken. Im Beispiel wird der `--query` Parameter verwendet, um nur die Instance anzuzeigenIDs.

```
aws ec2 describe-spot-instance-requests \
  --filters Name=launch.instance-type,Values=m3.medium Name=launched-availability-
zone,Values=us-east-2a \
  --query "SpotInstanceRequests[*].[InstanceId]" \
  --output text
```

Ausgabe:

```
i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...
```

Weitere Beispiele für die Verwendung von Filtern finden Sie unter [Auflisten und Filtern Ihrer Ressourcen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Beispiel 3: Zur Beschreibung von Spot-Instance-Anfragen auf der Grundlage von Tags

Im folgenden `describe-spot-instance-requests` Beispiel werden Tagfilter verwendet, um die Ergebnisse auf Spot-Instance-Anfragen zu beschränken, die das Tag `cost-center=cc123` enthalten.

```
aws ec2 describe-spot-instance-requests \
  --filters Name=tag:cost-center,Values=cc123
```

Ein Beispiel für die Ausgabe von `describe-spot-instance-requests` finden Sie in Beispiel 1.

Weitere Beispiele für die Verwendung von Tagfiltern finden Sie unter [Arbeiten mit Tags](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSpotInstanceRequests](#) in der AWS CLI Befehlsreferenz.

describe-spot-price-history

Das folgende Codebeispiel zeigt die Verwendung `describe-spot-price-history`.

AWS CLI

Um die Entwicklung der Spot-Preise zu beschreiben

Dieser Beispielbefehl gibt den Spot-Preisverlauf für m1.xlarge-Instances für einen bestimmten Tag im Januar zurück.

Befehl:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

Ausgabe:

```
{
  "SpotPriceHistory": [
    {
      "Timestamp": "2014-01-06T07:10:55.000Z",
      "ProductDescription": "SUSE Linux",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.087000",
      "AvailabilityZone": "us-west-1b"
    },
    {
      "Timestamp": "2014-01-06T07:10:55.000Z",
      "ProductDescription": "SUSE Linux",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.087000",
      "AvailabilityZone": "us-west-1c"
    },
    {
      "Timestamp": "2014-01-06T05:42:36.000Z",
      "ProductDescription": "SUSE Linux (Amazon VPC)",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.087000",
      "AvailabilityZone": "us-west-1a"
    },
    ...
  ]
}
```

Um die Spot-Preisentwicklung für Linux/Amazon UNIX zu beschreiben VPC

Dieser Beispielbefehl gibt den Spot-Preisverlauf für m1.xlarge-, UNIX VPC Linux/Amazon-Instances für einen bestimmten Tag im Januar zurück.

Befehl:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-  
description "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-  
time 2014-01-06T08:09:10
```

Ausgabe:

```
{  
  "SpotPriceHistory": [  
    {  
      "Timestamp": "2014-01-06T04:32:53.000Z",  
      "ProductDescription": "Linux/UNIX (Amazon VPC)",  
      "InstanceType": "m1.xlarge",  
      "SpotPrice": "0.080000",  
      "AvailabilityZone": "us-west-1a"  
    },  
    {  
      "Timestamp": "2014-01-05T11:28:26.000Z",  
      "ProductDescription": "Linux/UNIX (Amazon VPC)",  
      "InstanceType": "m1.xlarge",  
      "SpotPrice": "0.080000",  
      "AvailabilityZone": "us-west-1c"  
    }  
  ]  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeSpotPriceHistory](#) AWS CLI

describe-stale-security-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-stale-security-groups`.

AWS CLI

Um veraltete Sicherheitsgruppen zu beschreiben

In diesem Beispiel werden veraltete Sicherheitsgruppenregeln für `vpc-11223344` beschrieben. Die Antwort zeigt, dass `sg-5fa68d3a` in Ihrem Konto eine veraltete Eingangsregel hat, die auf den

Peer verweist, und dass es sg-279ab042 sg-fe6fba9a in Ihrem Konto eine veraltete SSH Ausgangsregel gibtVPC, die auf den Peer verweist. SSH sg-ef6fba8b VPC

Befehl:

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

Ausgabe:

```
{
  "StaleSecurityGroupSet": [
    {
      "VpcId": "vpc-11223344",
      "StaleIpPermissionsEgress": [
        {
          "ToPort": 22,
          "FromPort": 22,
          "UserIdGroupPairs": [
            {
              "VpcId": "vpc-7a20e51f",
              "GroupId": "sg-ef6fba8b",
              "VpcPeeringConnectionId": "pcx-b04deed9",
              "PeeringStatus": "active"
            }
          ],
          "IpProtocol": "tcp"
        }
      ],
      "GroupName": "MySG1",
      "StaleIpPermissions": [],
      "GroupId": "sg-fe6fba9a",
      "Description": "MySG1"
    },
    {
      "VpcId": "vpc-11223344",
      "StaleIpPermissionsEgress": [],
      "GroupName": "MySG2",
      "StaleIpPermissions": [
        {
          "ToPort": 22,
          "FromPort": 22,
          "UserIdGroupPairs": [
            {
```

```

        "VpcId": "vpc-7a20e51f",
        "GroupId": "sg-279ab042",
        "Description": "Access from pcx-b04deed9",
        "VpcPeeringConnectionId": "pcx-b04deed9",
        "PeeringStatus": "active"
    }
  ],
  "IpProtocol": "tcp"
}
],
"GroupId": "sg-5fa68d3a",
>Description": "MySG2"
}
]
}

```

- [DescribeStateSecurityGroups](#) Einzelheiten finden Sie AWS CLI in der Befehlsreferenz. API

describe-store-image-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-store-image-tasks`.

AWS CLI

Um den Fortschritt einer AMI Speicheraufgabe zu beschreiben

Das folgende `describe-store-image-tasks` Beispiel beschreibt den Fortschritt einer AMI Speicheraufgabe.

```
aws ec2 describe-store-image-tasks
```

Ausgabe:

```

{
  "StoreImageTaskResults": [
    {
      "AmiId": "ami-1234567890abcdef0",
      "Bucket": "my-ami-bucket",
      "ProgressPercentage": 17,
      "S3objectKey": "ami-1234567890abcdef0.bin",
      "StoreTaskState": "InProgress",
      "StoreTaskFailureReason": null,
    }
  ]
}

```

```

        "TaskStartTime": "2022-01-01T01:01:01.001Z"
      }
    ]
  }

```

Weitere Informationen zum Speichern und Wiederherstellen eines AMI mit S3 finden Sie unter [Speichern und Wiederherstellen eines AMI mit S3 <https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-store-restore.html>](https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-store-restore.html) im Amazon-Benutzerhandbuch. EC2

- Einzelheiten API finden Sie in der Befehlsreferenz. [DescribeStoreImageTasks](#) AWS CLI

describe-subnets

Das folgende Codebeispiel zeigt die Verwendung `describe-subnets`.

AWS CLI

Beispiel 1: So beschreiben Sie Ihre Subnetze

Im folgenden `describe-subnets`-Beispiel werden die Details zu Ihren Subnetzen angezeigt.

```
aws ec2 describe-subnets
```

Ausgabe:

```

{
  "Subnets": [
    {
      "AvailabilityZone": "us-east-1d",
      "AvailabilityZoneId": "use1-az2",
      "AvailableIpAddressCount": 4089,
      "CidrBlock": "172.31.80.0/20",
      "DefaultForAz": true,
      "MapPublicIpOnLaunch": false,
      "MapCustomerOwnedIpOnLaunch": true,
      "State": "available",
      "SubnetId": "subnet-0bb1c79de3EXAMPLE",
      "VpcId": "vpc-0ee975135dEXAMPLE",
      "OwnerId": "111122223333",
      "AssignIpv6AddressOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],
      "CustomerOwnedIpv4Pool": "pool-2EXAMPLE",
    }
  ]
}

```



```

        "SubnetArn": "arn:aws:ec2:us-east-2:111122223333:subnet/
subnet-0bb1c79de3EXAMPLE",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        }
    },
    {
        "AvailabilityZone": "us-east-1d",
        "AvailabilityZoneId": "use1-az2",
        "AvailableIpAddressCount": 4089,
        "CidrBlock": "172.31.80.0/20",
        "DefaultForAz": true,
        "MapPublicIpOnLaunch": true,
        "MapCustomerOwnedIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-8EXAMPLE",
        "VpcId": "vpc-3EXAMPLE",
        "OwnerId": "111122223333",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "Tags": [
            {
                "Key": "Name",
                "Value": "MySubnet"
            }
        ],
        "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit VPCs und Subnetzen](#) im AWS VPCBenutzerhandbuch.

Beispiel 2: Zur Beschreibung der Subnetze eines bestimmten VPC

Das folgende `describe-subnets` Beispiel verwendet einen Filter, um Details für die Subnetze der angegebenen Subnetze abzurufen. VPC

```
aws ec2 describe-subnets \  
  --filters "Name=vpc-id,Values=vpc-3EXAMPLE"
```

Ausgabe:

```
{  
  "Subnets": [  
    {  
      "AvailabilityZone": "us-east-1d",  
      "AvailabilityZoneId": "use1-az2",  
      "AvailableIpAddressCount": 4089,  
      "CidrBlock": "172.31.80.0/20",  
      "DefaultForAz": true,  
      "MapPublicIpOnLaunch": true,  
      "MapCustomerOwnedIpOnLaunch": false,  
      "State": "available",  
      "SubnetId": "subnet-8EXAMPLE",  
      "VpcId": "vpc-3EXAMPLE",  
      "OwnerId": "1111222233333",  
      "AssignIpv6AddressOnCreation": false,  
      "Ipv6CidrBlockAssociationSet": [],  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "MySubnet"  
        }  
      ],  
      "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/  
subnet-8EXAMPLE",  
      "EnableDns64": false,  
      "Ipv6Native": false,  
      "PrivateDnsNameOptionsOnLaunch": {  
        "HostnameType": "ip-name",  
        "EnableResourceNameDnsARecord": false,  
        "EnableResourceNameDnsAAAARecord": false  
      }  
    }  
  ]  
}
```

```
}  
  }  
] }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit VPCs und Subnetzen](#) im AWS VPCBenutzerhandbuch.

Beispiel 3: So beschreiben Sie die Subnetze mit einem bestimmten Tag

Das folgende `describe-subnets` Beispiel verwendet einen Filter, um die Details dieser Subnetze mit dem Tag `CostCenter=123` und dem `--query` Parameter abzurufen, um das Subnetz IDs der Subnetze mit diesem Tag anzuzeigen.

```
aws ec2 describe-subnets \  
  --filters "Name=tag:CostCenter,Values=123" \  
  --query "Subnets[*].SubnetId" \  
  --output text
```

Ausgabe:

```
subnet-0987a87c8b37348ef  
subnet-02a95061c45f372ee  
subnet-03f720e7de2788d73
```

Weitere Informationen finden Sie unter [Arbeiten mit VPCs und Subnetzen](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeSubnets](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Beispiel 1: Um alle Tags für eine einzelne Ressource zu beschreiben

Das folgende `describe-tags` Beispiel beschreibt die Tags für die angegebene Instanz.

```
aws ec2 describe-tags \  
  --filters "Name=resource-id,Values=i-1234567890abcdef8"
```

Ausgabe:

```
{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Beta Server",
      "Key": "Name"
    }
  ]
}
```

Beispiel 2: Um alle Tags für einen Ressourcentyp zu beschreiben

Das folgende describe-tags Beispiel beschreibt die Tags für Ihre Volumes.

```
aws ec2 describe-tags \
  --filters "Name=resource-type,Values=volume"
```

Ausgabe:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-1234567890abcdef0",
      "Value": "Project1",
      "Key": "Purpose"
    },
    {
      "ResourceType": "volume",
      "ResourceId": "vol-049df61146c4d7901",
      "Value": "Logs",
      "Key": "Purpose"
    }
  ]
}
```

```
}
```

Beispiel 3: Um all Ihre Tags zu beschreiben

Das folgende `describe-tags` Beispiel beschreibt die Tags für all Ihre Ressourcen.

```
aws ec2 describe-tags
```

Beispiel 4: Um die Tags für Ihre Ressourcen anhand eines Tag-Schlüssels zu beschreiben

Das folgende `describe-tags` Beispiel beschreibt die Tags für Ihre Ressourcen, die ein Tag mit dem Schlüssel `habenStack`.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Stack
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "ResourceType": "volume",  
      "ResourceId": "vol-027552a73f021f3b",  
      "Value": "Production",  
      "Key": "Stack"  
    },  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",  
      "Value": "Test",  
      "Key": "Stack"  
    }  
  ]  
}
```

Beispiel 5: Um die Tags für Ihre Ressourcen auf der Grundlage eines Tag-Schlüssels und eines Tag-Werts zu beschreiben

Das folgende `describe-tags` Beispiel beschreibt die Tags für Ihre Ressourcen, die das Tag `enthaltenStack=Test`.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Test
```

```
--filters Name=key,Values=Stack Name=value,Values=Test
```

Ausgabe:

```
{
  "Tags": [
    {
      "ResourceType": "image",
      "ResourceId": "ami-3ac336533f021f3bd",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    }
  ]
}
```

Im folgenden `describe-tags` Beispiel wird eine alternative Syntax verwendet, um Ressourcen mit dem Tag zu beschreiben `Stack=Test`.

```
aws ec2 describe-tags \  
--filters "Name=tag:Stack,Values=Test"
```

Das folgende `describe-tags` Beispiel beschreibt die Tags für all Ihre Instances, die ein Tag mit dem Schlüssel `Purpose` und ohne Wert haben.

```
aws ec2 describe-tags \  
--filters "Name=resource-  
type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="
```

Ausgabe:

```
{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef5",
      "Value": null,

```

```

    "Key": "Purpose"
  }
]
}

```

- API-Einheiten finden Sie [DescribeTags](#) unter AWS CLI Befehlsreferenz.

describe-traffic-mirror-filters

Das folgende Codebeispiel zeigt die Verwendung `describe-traffic-mirror-filters`.

AWS CLI

Um Ihre Traffic Mirror-Filter anzuzeigen

Im folgenden `describe-traffic-mirror-filters` Beispiel werden Details zu all Ihren Traffic Mirror-Filtern angezeigt.

```
aws ec2 describe-traffic-mirror-filters
```

Ausgabe:

```

{
  "TrafficMirrorFilters": [
    {
      "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
      "IngressFilterRules": [
        {
          "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
          "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
          "TrafficDirection": "ingress",
          "RuleNumber": 100,
          "RuleAction": "accept",
          "Protocol": 6,
          "DestinationCidrBlock": "10.0.0.0/24",
          "SourceCidrBlock": "10.0.0.0/24",
          "Description": "TCP Rule"
        }
      ],
      "EgressFilterRules": [],
      "NetworkServices": [],
      "Description": "Example filter",
    }
  ]
}

```

```
    "Tags": []
  }
]
}
```

Weitere Informationen finden Sie im [Traffic Mirroring Guide unter Ihre Traffic Mirroring-Filter anzeigen](#).

- API-Einheiten finden Sie [DescribeTrafficMirrorFilters](#) in der AWS CLI Befehlsreferenz.

describe-traffic-mirror-sessions

Das folgende Codebeispiel zeigt die Verwendung `describe-traffic-mirror-sessions`.

AWS CLI

Um eine Traffic Mirror-Sitzung zu beschreiben

Im folgenden `describe-traffic-mirror-sessions` Beispiel werden Details zu Ihren Traffic Mirror-Sitzungen angezeigt.

```
aws ec2 describe-traffic-mirror-sessions
```

Ausgabe:

```
{
  "TrafficMirrorSessions": [
    {
      "Tags": [],
      "VirtualNetworkId": 42,
      "OwnerId": "111122223333",
      "Description": "TCP Session",
      "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
      "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
      "TrafficMirrorFilterId": "tmf-083e18f985EXAMPLE",
      "PacketLength": 20,
      "SessionNumber": 1,
      "TrafficMirrorSessionId": "tms-0567a4c684EXAMPLE"
    },
    {
      "Tags": [
        {
          "Key": "Name",
```



```

        "Value": "tag test"
      }
    ],
    "VirtualNetworkId": 13314501,
    "OwnerId": "111122223333",
    "Description": "TCP Session",
    "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
    "TrafficMirrorTargetId": "tmt-03665551cbEXAMPLE",
    "TrafficMirrorFilterId": "tmf-06c787846cEXAMPLE",
    "SessionNumber": 2,
    "TrafficMirrorSessionId": "tms-0060101cf8EXAMPLE"
  }
]
}

```

Weitere Informationen finden Sie unter [Traffic Mirror-Sitzungsdetails anzeigen](#) im AWS Traffic Mirroring-Handbuch.

- API-Einheiten finden Sie unter [DescribeTrafficMirrorSessions AWS CLI Befehlsreferenz](#).

describe-traffic-mirror-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-traffic-mirror-targets`.

AWS CLI

Um ein Traffic Mirror-Ziel zu beschreiben

Im folgenden `describe-traffic-mirror-targets` Beispiel werden Informationen über das angegebene Traffic Mirror-Ziel angezeigt.

```
aws ec2 describe-traffic-mirror-targets \
  --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE
```

Ausgabe:

```
{
  "TrafficMirrorTargets": [
    {
      "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
      "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:111122223333:loadbalancer/net/NLB/7cdec873fEXAMPLE",
      "Type": "network-load-balancer",

```

```
    "Description": "Example Network Load Balancer target",
    "OwnerId": "111122223333",
    "Tags": []
  }
]
```

Weitere Informationen finden Sie unter [Traffic Mirroring Targets](#) im Amazon VPC Traffic Mirroring Guide.

- API-Einheiten finden Sie [DescribeTrafficMirrorTargets](#) in der AWS CLI Befehlsreferenz.

describe-transit-gateway-attachments

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-attachments`.

AWS CLI

Um Ihre Transit-Gateway-Anhänge anzuzeigen

Im folgenden `describe-transit-gateway-attachments` Beispiel werden Details zu Ihren Transit-Gateway-Anhängen angezeigt.

```
aws ec2 describe-transit-gateway-attachments
```

Ausgabe:

```
{
  "TransitGatewayAttachments": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
      "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
      "TransitGatewayOwnerId": "123456789012",
      "ResourceOwnerId": "123456789012",
      "ResourceType": "vpc",
      "ResourceId": "vpc-3EXAMPLE",
      "State": "available",
      "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "State": "associated"
      },
      "CreationTime": "2019-08-26T14:59:25.000Z",
      "Tags": [
```

```
        {
            "Key": "Name",
            "Value": "Example"
        }
    ]
},
{
    "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "TransitGatewayOwnerId": "123456789012",
    "ResourceOwnerId": "123456789012",
    "ResourceType": "vpc",
    "ResourceId": "vpc-0065acced4EXAMPLE",
    "State": "available",
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "State": "associated"
    },
    "CreationTime": "2019-08-07T17:03:07.000Z",
    "Tags": []
},
{
    "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "TransitGatewayOwnerId": "123456789012",
    "ResourceOwnerId": "123456789012",
    "ResourceType": "direct-connect-gateway",
    "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "State": "available",
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "State": "associated"
    },
    "CreationTime": "2019-08-14T20:27:44.000Z",
    "Tags": []
},
{
    "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "TransitGatewayOwnerId": "123456789012",
    "ResourceOwnerId": "123456789012",
    "ResourceType": "direct-connect-gateway",
    "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
    "State": "available",
```

```

    "Association": {
      "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
      "State": "associated"
    },
    "CreationTime": "2019-08-14T20:33:02.000Z",
    "Tags": []
  }
]
}

```

Weitere Informationen finden Sie im [Transit Gateways Guide unter Arbeiten mit Transit-Gateways](#).

- API-Einheiten finden Sie unter [DescribeTransitGatewayAttachments AWS CLI-Befehlsreferenz](#).

describe-transit-gateway-connect-peers

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-connect-peers`.

AWS CLI

Um einen Transit Gateway Connect-Peer zu beschreiben

Das folgende `describe-transit-gateway-connect-peers` Beispiel beschreibt den angegebenen Connect-Peer.

```

aws ec2 describe-transit-gateway-connect-peers \
  --transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE

```

Ausgabe:

```

{
  "TransitGatewayConnectPeers": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
      "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
      "State": "available",
      "CreationTime": "2021-10-13T03:35:17.000Z",
      "ConnectPeerConfiguration": {
        "TransitGatewayAddress": "10.0.0.234",
        "PeerAddress": "172.31.1.11",
        "InsideCidrBlocks": [

```

```

        "169.254.6.0/29"
    ],
    "Protocol": "gre",
    "BgpConfigurations": [
        {
            "TransitGatewayAsn": 64512,
            "PeerAsn": 64512,
            "TransitGatewayAddress": "169.254.6.2",
            "PeerAddress": "169.254.6.1",
            "BgpStatus": "down"
        },
        {
            "TransitGatewayAsn": 64512,
            "PeerAsn": 64512,
            "TransitGatewayAddress": "169.254.6.3",
            "PeerAddress": "169.254.6.1",
            "BgpStatus": "down"
        }
    ]
},
"Tags": []
}
]
}

```

Weitere Informationen finden Sie unter [Transit Gateway Connect-Anlagen und Transit Gateway Connect-Peers](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DescribeTransitGatewayConnectPeers AWS CLI Befehlsreferenz](#).

describe-transit-gateway-connects

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-connects`.

AWS CLI

Um einen Transit Gateway Connect-Anhang zu beschreiben

Das folgende `describe-transit-gateway-connects` Beispiel beschreibt den angegebenen Connect-Anhang.

```
aws ec2 describe-transit-gateway-connects \
```

```
--transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayConnects": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
      "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
      "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
      "State": "available",
      "CreationTime": "2021-03-09T19:59:17+00:00",
      "Options": {
        "Protocol": "gre"
      },
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Transit Gateway Connect-Anlagen und Transit Gateway Connect-Peers](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DescribeTransitGatewayConnects AWS CLI-Befehlsreferenz](#).

describe-transit-gateway-multicast-domains

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-multicast-domains`.

AWS CLI

Um Ihre Transit-Gateway-Multicast-Domänen zu beschreiben

Im folgenden `describe-transit-gateway-multicast-domains` Beispiel werden Details für alle Ihre Transit-Gateway-Multicast-Domänen angezeigt.

```
aws ec2 describe-transit-gateway-multicast-domains
```

Ausgabe:

```
{
```

```

"TransitGatewayMulticastDomains": [
  {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bfffefaEXAMPLE",
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
east-1:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
    "OwnerId": "123456789012",
    "Options": {
      "Icmpv2Support": "disable",
      "StaticSourcesSupport": "enable",
      "AutoAcceptSharedAssociations": "disable"
    },
    "State": "available",
    "CreationTime": "2019-12-10T18:32:50+00:00",
    "Tags": [
      {
        "Key": "Name",
        "Value": "mc1"
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Verwaltung von Multicast-Domänen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie [DescribeTransitGatewayMulticastDomains](#) in der AWS CLI Befehlsreferenz.

describe-transit-gateway-peering-attachments

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-peering-attachments`.

AWS CLI

Um Ihre Transit-Gateway-Peering-Anhänge zu beschreiben

Im folgenden `describe-transit-gateway-peering-attachments` Beispiel werden Details zu all Ihren Transit-Gateway-Peering-Anhängen angezeigt.

aws ec2 describe-transit-gateway-peering-attachments

Ausgabe:

```
{
  "TransitGatewayPeeringAttachments": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
      "RequesterTgwInfo": {
        "TransitGatewayId": "tgw-123abc05e04123abc",
        "OwnerId": "123456789012",
        "Region": "us-west-2"
      },
      "AccepterTgwInfo": {
        "TransitGatewayId": "tgw-11223344aabbcc112",
        "OwnerId": "123456789012",
        "Region": "us-east-2"
      },
      "State": "pendingAcceptance",
      "CreationTime": "2019-12-09T11:38:05.000Z",
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Transit Gateway Peering Attachments](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DescribeTransitGatewayPeeringAttachments AWS CLI Befehlsreferenz](#).

describe-transit-gateway-policy-tables

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-policy-tables`.

AWS CLI

Um eine Richtlinientabelle für ein Transit-Gateway zu beschreiben

Das folgende `describe-transit-gateway-policy-tables` Beispiel beschreibt die angegebene Richtlinientabelle für das Transit-Gateway.


```
aws ec2 describe-transit-gateway-policy-tables \
  --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

Ausgabe:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",
      "State": "available",
      "CreationTime": "2023-11-28T16:36:43+00:00",
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie in den [Richtlinientabellen für Transit Gateway](#) im Transit Gateway-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTransitGatewayPolicyTables](#) unter AWS CLI Befehlsreferenz.

describe-transit-gateway-route-tables

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-route-tables`.

AWS CLI

Um Ihre Transit-Gateway-Routentabellen zu beschreiben

Im folgenden `describe-transit-gateway-route-tables` Beispiel werden Details zu Ihren Transit-Gateway-Routentabellen angezeigt.

```
aws ec2 describe-transit-gateway-route-tables
```

Ausgabe:

```
{
  "TransitGatewayRouteTables": [
```

```
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0ca78a549EXAMPLE",
      "TransitGatewayId": "tgw-0bc994abffEXAMPLE",
      "State": "available",
      "DefaultAssociationRouteTable": true,
      "DefaultPropagationRouteTable": true,
      "CreationTime": "2018-11-28T14:24:49.000Z",
      "Tags": []
    },
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0e8f48f148EXAMPLE",
      "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
      "State": "available",
      "DefaultAssociationRouteTable": true,
      "DefaultPropagationRouteTable": true,
      "CreationTime": "2018-11-28T14:24:00.000Z",
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Routentabellen für Transit-Gateways anzeigen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [DescribeTransitGatewayRouteTables AWS CLI Befehlsreferenz](#).

describe-transit-gateway-vpc-attachments

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateway-vpc-attachments`.

AWS CLI

Um Ihre VPC Transit-Gateway-Anhänge zu beschreiben

Im folgenden `describe-transit-gateway-vpc-attachments` Beispiel werden Details zu Ihren VPC Transit-Gateway-Anhängen angezeigt.

```
aws ec2 describe-transit-gateway-vpc-attachments
```

Ausgabe:

```

{
  "TransitGatewayVpcAttachments": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-0a08e88308EXAMPLE",
      "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
      "VpcId": "vpc-0f501f7ee8EXAMPLE",
      "VpcOwnerId": "111122223333",
      "State": "available",
      "SubnetIds": [
        "subnet-045d586432EXAMPLE",
        "subnet-0a0ad478a6EXAMPLE"
      ],
      "CreationTime": "2019-02-13T11:04:02.000Z",
      "Options": {
        "DnsSupport": "enable",
        "Ipv6Support": "disable"
      },
      "Tags": [
        {
          "Key": "Name",
          "Value": "attachment name"
        }
      ]
    }
  ]
}

```

Weitere Informationen finden Sie im Transit Gateways Guide unter [Ihre VPC Anlagen anzeigen](#).

- API-Einheiten finden Sie unter [DescribeTransitGatewayVpcAttachments AWS CLI Befehlsreferenz](#).

describe-transit-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-transit-gateways`.

AWS CLI

Um Ihre Transit-Gateways zu beschreiben

Im folgenden `describe-transit-gateways` Beispiel werden Details zu Ihren Transit-Gateways abgerufen.

aws ec2 describe-transit-gateways

Ausgabe:

```
{
  "TransitGateways": [
    {
      "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
      "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0262a0e521EXAMPLE",
      "State": "available",
      "OwnerId": "111122223333",
      "Description": "MyTGW",
      "CreationTime": "2019-07-10T14:02:12.000Z",
      "Options": {
        "AmazonSideAsn": 64516,
        "AutoAcceptSharedAttachments": "enable",
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
      },
      "Tags": []
    },
    {
      "TransitGatewayId": "tgw-0fb8421e2dEXAMPLE",
      "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0fb8421e2da853bf3",
      "State": "available",
      "OwnerId": "111122223333",
      "CreationTime": "2019-03-15T22:57:33.000Z",
      "Options": {
        "AmazonSideAsn": 65412,
        "AutoAcceptSharedAttachments": "disable",
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
      },
    },
  ],
}
```

```

    "Tags": [
      {
        "Key": "Name",
        "Value": "TGW1"
      }
    ]
  }
]
}

```

- API-Einheiten finden Sie unter [Describe Transit Gateways AWS CLI Befehlsreferenz](#).

describe-verified-access-endpoints

Das folgende Codebeispiel zeigt die Verwendung von `describe-verified-access-endpoints`.

AWS CLI

Um einen Verified Access-Endpunkt zu beschreiben

Im folgenden `delete-verified-access-endpoints` Beispiel wird der angegebene Endpunkt für verifizierten Zugriff beschrieben.

```

aws ec2 describe-verified-access-endpoints \
  --verified-access-endpoint-ids vae-066fac616d4d546f2

```

Ausgabe:

```

{
  "VerifiedAccessEndpoints": [
    {
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
      "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
      "ApplicationDomain": "example.com",
      "EndpointType": "network-interface",
      "AttachmentType": "vpc",
      "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
      "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
    }
  ]
}

```

```
    "SecurityGroupIds": [
      "sg-004915970c4c8f13a"
    ],
    "NetworkInterfaceOptions": {
      "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
      "Protocol": "https",
      "Port": 443
    },
    "Status": {
      "Code": "active"
    },
    "Description": "",
    "CreationTime": "2023-08-25T20:54:43",
    "LastUpdatedTime": "2023-08-25T22:17:26",
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-va-endpoint"
      }
    ]
  }
]
```

Weitere Informationen finden Sie unter [Verified Access-Endpoints](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVerifiedAccessEndpoints](#) in der AWS CLI Befehlsreferenz.

describe-verified-access-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-verified-access-groups`.

AWS CLI

Um eine Verified Access-Gruppe zu beschreiben

Im folgenden `describe-verified-access-groups` Beispiel wird die angegebene Gruppe mit verifiziertem Zugriff beschrieben.

```
aws ec2 describe-verified-access-groups \
  --verified-access-group-ids vagr-0dbe967baf14b7235
```

Ausgabe:

```
{
  "VerifiedAccessGroups": [
    {
      "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "Description": "Testing Verified Access",
      "Owner": "123456789012",
      "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
      "CreationTime": "2023-08-25T19:55:19",
      "LastUpdatedTime": "2023-08-25T22:17:25",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-va-group"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verified Access-Gruppen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVerifiedAccessGroups](#) in der AWS CLI Befehlsreferenz.

describe-verified-access-instance-logging-configurations

Das folgende Codebeispiel zeigt die Verwendung `describe-verified-access-instance-logging-configurations`.

AWS CLI

Um die Protokollierungskonfiguration für eine Verified Access-Instanz zu beschreiben

Das folgende `describe-verified-access-instance-logging-configurations` Beispiel beschreibt die Protokollierungskonfiguration für die angegebene Verified Access-Instanz.

```
aws ec2 describe-verified-access-instance-logging-configurations \
  --verified-access-instance-ids vai-0ce000c0b7643abea
```

Ausgabe:

```
{
  "LoggingConfigurations": [
    {
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "AccessLogs": {
        "S3": {
          "Enabled": false
        },
        "CloudWatchLogs": {
          "Enabled": true,
          "DeliveryStatus": {
            "Code": "success"
          },
          "LogGroup": "my-log-group"
        },
        "KinesisDataFirehose": {
          "Enabled": false
        },
        "LogVersion": "ocsf-1.0.0-rc.2",
        "IncludeTrustContext": false
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verified Access-Logs](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVerifiedAccessInstanceLoggingConfigurations](#) in der AWS CLI Befehlsreferenz.

describe-verified-access-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-verified-access-instances`.

AWS CLI

Um eine Verified Access-Instanz zu beschreiben

Das folgende `describe-verified-access-instances` Beispiel beschreibt die angegebene Verified Access-Instanz.


```
aws ec2 describe-verified-access-instances \
  --verified-access-instance-ids vai-0ce000c0b7643abea
```

Ausgabe:

```
{
  "VerifiedAccessInstances": [
    {
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "Description": "Testing Verified Access",
      "VerifiedAccessTrustProviders": [
        {
          "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
          "TrustProviderType": "user",
          "UserTrustProviderType": "iam-identity-center"
        }
      ],
      "CreationTime": "2023-08-25T18:27:56",
      "LastUpdatedTime": "2023-08-25T19:03:32",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-ava-instance"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verified Access-Instanzen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVerifiedAccessInstances](#) in der AWS CLI Befehlsreferenz.

describe-verified-access-trust-providers

Das folgende Codebeispiel zeigt die Verwendung `describe-verified-access-trust-providers`.

AWS CLI

Um einen Vertrauensanbieter mit verifiziertem Zugriff zu beschreiben

Im folgenden `describe-verified-access-trust-providers` Beispiel wird der angegebene Verified Access-Vertrauensanbieter beschrieben.

```
aws ec2 describe-verified-access-trust-providers \
  --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

Ausgabe:

```
{
  "VerifiedAccessTrustProviders": [
    {
      "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
      "Description": "Testing Verified Access",
      "TrustProviderType": "user",
      "UserTrustProviderType": "iam-identity-center",
      "PolicyReferenceName": "idc",
      "CreationTime": "2023-08-25T19:00:38",
      "LastUpdatedTime": "2023-08-25T19:03:32",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-va-trust-provider"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Vertrauensanbietern für verifizierten Zugriff](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVerifiedAccessTrustProviders](#) in der AWS CLI Befehlsreferenz.

describe-volume-attribute

Das folgende Codebeispiel zeigt die Verwendung `describe-volume-attribute`.

AWS CLI

Um ein Volumenattribut zu beschreiben

Dieser Beispielbefehl beschreibt das `autoEnableIo` Attribut des Volumes mit der ID `vol-049df61146c4d7901`.

Befehl:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --  
attribute autoEnableIO
```

Ausgabe:

```
{  
  "AutoEnableIO": {  
    "Value": false  
  },  
  "VolumeId": "vol-049df61146c4d7901"  
}
```

- API-Einheiten finden Sie [DescribeVolumeAttribute](#) in der AWS CLI Befehlsreferenz.

describe-volume-status

Das folgende Codebeispiel zeigt die Verwendung `describe-volume-status`.

AWS CLI

Um den Status eines einzelnen Volumes zu beschreiben

Dieser Beispielbefehl beschreibt den Status des Volumes `vol-1234567890abcdef0`.

Befehl:

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

Ausgabe:

```
{  
  "VolumeStatuses": [  
    {  
      "VolumeStatus": {  
        "Status": "ok",  
        "Details": [  

```

```

        {
            "Status": "passed",
            "Name": "io-enabled"
        },
        {
            "Status": "not-applicable",
            "Name": "io-performance"
        }
    ]
},
"AvailabilityZone": "us-east-1a",
"VolumeId": "vol-1234567890abcdef0",
"Actions": [],
"Events": []
}
]
}

```

Um den Status von kaputten Volumes zu beschreiben

Dieser Beispielbefehl beschreibt den Status aller Datenträger, die beeinträchtigt sind. In dieser Beispielausgabe gibt es keine beeinträchtigten Volumes.

Befehl:

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

Ausgabe:

```

{
  "VolumeStatuses": []
}

```

Wenn bei Ihrem Volumen die Statusüberprüfung nicht bestanden wurde (der Status ist beeinträchtigt), finden Sie weitere Informationen unter Arbeiten mit einem beeinträchtigten Volumen im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeVolumeStatus](#) in der AWS CLI Befehlsreferenz.

describe-volumes-modifications

Das folgende Codebeispiel zeigt die Verwendung `describe-volumes-modifications`.

AWS CLI

Um den Änderungsstatus für ein Volume zu beschreiben

Das folgende `describe-volumes-modifications` Beispiel beschreibt den Status der Volumenänderung des angegebenen Volumes.

```
aws ec2 describe-volumes-modifications \  
  --volume-ids vol-1234567890abcdef0
```

Ausgabe:

```
{  
  "VolumeModification": {  
    "TargetSize": 150,  
    "TargetVolumeType": "io1",  
    "ModificationState": "optimizing",  
    "VolumeId": " vol-1234567890abcdef0",  
    "TargetIops": 100,  
    "StartTime": "2019-05-17T11:27:19.000Z",  
    "Progress": 70,  
    "OriginalVolumeType": "io1",  
    "OriginalIops": 100,  
    "OriginalSize": 100  
  }  
}
```

- API-Einheiten finden Sie [DescribeVolumesModifications](#) in der AWS CLI Befehlsreferenz.

describe-volumes

Das folgende Codebeispiel zeigt die Verwendung `describe-volumes`.

AWS CLI

Beispiel 1: Um ein Volumen zu beschreiben

Das folgende `describe-volumes` Beispiel beschreibt die angegebenen Volumes in der aktuellen Region.

```
aws ec2 describe-volumes \  
  --volume-ids vol-1234567890abcdef0
```

```
--volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

Ausgabe:

```
{
  "Volumes": [
    {
      "AvailabilityZone": "us-east-1a",
      "Attachments": [
        {
          "AttachTime": "2013-12-18T22:35:00.000Z",
          "InstanceId": "i-1234567890abcdef0",
          "VolumeId": "vol-049df61146c4d7901",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "Encrypted": true,
      "KmsKeyId": "arn:aws:kms:us-east-2a:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513eEXAMPLE",
      "VolumeType": "gp2",
      "VolumeId": "vol-049df61146c4d7901",
      "State": "in-use",
      "Iops": 100,
      "SnapshotId": "snap-1234567890abcdef0",
      "CreateTime": "2019-12-18T22:35:00.084Z",
      "Size": 8
    },
    {
      "AvailabilityZone": "us-east-1a",
      "Attachments": [],
      "Encrypted": false,
      "VolumeType": "gp2",
      "VolumeId": "vol-1234567890abcdef0",
      "State": "available",
      "Iops": 300,
      "SnapshotId": "",
      "CreateTime": "2020-02-27T00:02:41.791Z",
      "Size": 100
    }
  ]
}
```

Beispiel 2: Zur Beschreibung von Volumes, die an eine bestimmte Instanz angehängt sind

Im folgenden `describe-volumes` Beispiel werden alle Volumes beschrieben, die sowohl an die angegebene Instance angehängt als auch so eingestellt sind, dass sie gelöscht werden, wenn die Instance beendet wird.

```
aws ec2 describe-volumes \  
  --region us-east-1 \  
  --filters Name=attachment.instance-  
id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true
```

Ein Beispiel für die Ausgabe von `describe-volumes` finden Sie in Beispiel 1.

Beispiel 3: Um verfügbare Volumes in einer bestimmten Availability Zone zu beschreiben

Im folgenden `describe-volumes` Beispiel werden alle Volumes beschrieben, die den Status `available` haben und sich in der angegebenen Availability Zone befinden.

```
aws ec2 describe-volumes \  
  --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a
```

Ein Beispiel für die Ausgabe von `describe-volumes` finden Sie in Beispiel 1.

Beispiel 4: Um Volumes anhand von Tags zu beschreiben

Das folgende `describe-volumes` Beispiel beschreibt alle Volumes, die den Tag-Schlüssel `Name` und einen Wert haben, der mit `beginntest` beginnt. Die Ausgabe wird dann mit einer Abfrage gefiltert, bei der nur die Tags und IDs der Volumes angezeigt werden.

```
aws ec2 describe-volumes \  
  --filters Name=tag:Name,Values=Test* \  
  --query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

Ausgabe:

```
[  
  {  
    "Tag": [  
      {  
        "Value": "Test2",  
        "Key": "Name"      }  
    ]  
  }  
]
```

```
    }
  ],
  "ID": "vol-1234567890abcdef0"
},
{
  "Tag": [
    {
      "Value": "Test1",
      "Key": "Name"
    }
  ],
  "ID": "vol-049df61146c4d7901"
}
]
```

Weitere Beispiele für die Verwendung von Tagfiltern finden Sie unter [Arbeiten mit Tags](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeVolumes](#) in der AWS CLI Befehlsreferenz.

describe-vpc-attribute

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-attribute`.

AWS CLI

Um das `enableDnsSupport` Attribut zu beschreiben

Dieses Beispiel beschreibt das `enableDnsSupport` Attribut. Dieses Attribut gibt an, ob die DNS Auflösung für aktiviert ist VPC. Wenn dieses Attribut zutrifft `true`, löst der DNS Amazon-Server DNS Hostnamen für Ihre Instances in die entsprechenden IP-Adressen auf; andernfalls nicht.

Befehl:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

Ausgabe:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsSupport": {
    "Value": true
  }
}
```



```
}  
}
```

Um das Attribut zu beschreiben `enableDnsHostnames`

Dieses Beispiel beschreibt das `enableDnsHostnames` Attribut. Dieses Attribut gibt an, ob die Instances im VPC Get DNS Hostnames gestartet wurden. Wenn dieses Attribut zutrifft `true`, VPC erhalten die Instanzen in der DNS Liste Hostnamen, andernfalls nicht.

Befehl:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

Ausgabe:

```
{  
  "VpcId": "vpc-a01106c2",  
  "EnableDnsHostnames": {  
    "Value": true  
  }  
}
```

- API Einheiten finden Sie [DescribeVpcAttribute](#) in der AWS CLI Befehlsreferenz.

describe-vpc-classic-link-dns-support

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-classic-link-dns-support`.

AWS CLI

Um die ClassicLink DNS Unterstützung für Ihr zu beschreiben VPCs

Dieses Beispiel beschreibt den ClassicLink DNS Support-Status aller Ihrer VPCs.

Befehl:

```
aws ec2 describe-vpc-classic-link-dns-support
```

Ausgabe:

```
{
```

```
"Vpcs": [  
  {  
    "VpcId": "vpc-88888888",  
    "ClassicLinkDnsSupported": true  
  },  
  {  
    "VpcId": "vpc-1a2b3c4d",  
    "ClassicLinkDnsSupported": false  
  }  
]
```

- API-Einheiten finden Sie [DescribeVpcClassicLinkDnsSupport](#) in der AWS CLI Befehlsreferenz.

describe-vpc-classic-link

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-classic-link`.

AWS CLI

Um den ClassicLink Status Ihres zu beschreiben VPCs

Dieses Beispiel listet den ClassicLink Status von `vpc-88888888` auf.

Befehl:

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

Ausgabe:

```
{  
  "Vpcs": [  
    {  
      "ClassicLinkEnabled": true,  
      "VpcId": "vpc-88888888",  
      "Tags": [  
        {  
          "Value": "classiclinkvpc",  
          "Key": "Name"  
        }  
      ]  
    }  
  ]  
}
```

```
]
}
```

In diesem Beispiel werden nur diejenigen aufgeführt VPCs, die für Classiclink aktiviert sind (der Filterwert von `is-classic-link-enabled` ist auf `true` gesetzt).

Befehl:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-enabled,Values=true"
```

- API-Einheiten finden Sie unter [DescribeVpcClassicLink AWS CLI Befehlsreferenz](#).

describe-vpc-endpoint-connection-notifications

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-endpoint-connection-notifications`.

AWS CLI

Um Benachrichtigungen über Endpunktverbindungen zu beschreiben

Das folgende `describe-vpc-endpoint-connection-notifications` Beispiel beschreibt alle Ihre Benachrichtigungen über Endpunktverbindungen.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

Ausgabe:

```
{
  "ConnectionNotificationSet": [
    {
      "ConnectionNotificationState": "Enabled",
      "ConnectionNotificationType": "Topic",
      "ConnectionEvents": [
        "Accept",
        "Reject",
        "Delete",
        "Connect"
      ],
      "ConnectionNotificationId": "vpce-nfn-04bcb952bc8af7abc",
```

```

    "ConnectionNotificationArn": "arn:aws:sns:us-
east-1:123456789012:VpceNotification",
    "VpcEndpointId": "vpce-0324151a02f327123"
  }
]
}

```

- API-Einheiten finden Sie [DescribeVpcEndpointConnectionNotifications](#) in der AWS CLI Befehlsreferenz.

describe-vpc-endpoint-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-endpoint-connections`.

AWS CLI

Um VPC-Endpunktverbindungen zu beschreiben

In diesem Beispiel werden die Schnittstellenendpunktverbindungen zu Ihrem Endpunktdienst beschrieben und die Ergebnisse so gefiltert, dass Endpunkte angezeigt werden, die es sind `PendingAcceptance`.

Befehl:

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

Ausgabe:

```

{
  "VpcEndpointConnections": [
    {
      "VpcEndpointId": "vpce-0abed31004e618123",
      "ServiceId": "vpce-svc-0abced088d20def56",
      "CreationTimestamp": "2017-11-30T10:00:24.350Z",
      "VpcEndpointState": "pendingAcceptance",
      "VpcEndpointOwner": "123456789012"
    }
  ]
}

```

- API-Einheiten finden Sie unter [DescribeVpcEndpointConnections AWS CLI](#) Befehlsreferenz.

describe-vpc-endpoint-service-configurations

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-endpoint-service-configurations`.

AWS CLI

Um Konfigurationen von Endpunktdiensten zu beschreiben

Das folgende `describe-vpc-endpoint-service-configurations` Beispiel beschreibt Ihre Endpunkt-Servicekonfigurationen.

```
aws ec2 describe-vpc-endpoint-service-configurations
```

Ausgabe:

```
{
  "ServiceConfigurations": [
    {
      "ServiceType": [
        {
          "ServiceType": "GatewayLoadBalancer"
        }
      ],
      "ServiceId": "vpce-svc-012d33a1c4321cabc",
      "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-012d33a1c4321cabc",
      "ServiceState": "Available",
      "AvailabilityZones": [
        "us-east-1d"
      ],
      "AcceptanceRequired": false,
      "ManagesVpcEndpoints": false,
      "GatewayLoadBalancerArns": [
        "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
gwy/GWLBSvc/123210844e429123"
      ],
      "Tags": []
    },
    {
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "ServiceId": "vpce-svc-123cab125efa123",
  "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123cab125efa123",
  "ServiceState": "Available",
  "AvailabilityZones": [
    "us-east-1a"
  ],
  "AcceptanceRequired": true,
  "ManagesVpcEndpoints": false,
  "NetworkLoadBalancerArns": [
    "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
net/NLBforService/1238753950b25123"
  ],
  "BaseEndpointDnsNames": [
    "vpce-svc-123cab125efa123.us-east-1.vpce.amazonaws.com"
  ],
  "PrivateDnsName": "example.com",
  "PrivateDnsNameConfiguration": {
    "State": "failed",
    "Type": "TXT",
    "Value": "vpce:qUAth3FdeABCaPuiXabc",
    "Name": "_1d367jvbg34znqvyefrj"
  },
  "Tags": []
}
]
}

```

Weitere Informationen finden Sie unter [VPC Endpoint Services](#) im VPC Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeVpcEndpointServiceConfigurations](#) in der AWS CLI Befehlsreferenz.

describe-vpc-endpoint-service-permissions

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-endpoint-service-permissions`.

AWS CLI

Um die Berechtigungen für Endpunktdienste zu beschreiben

In diesem Beispiel werden die Berechtigungen für den angegebenen Endpunktdienst beschrieben.

Befehl:

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3
```

Ausgabe:

```
{  
  "AllowedPrincipals": [  
    {  
      "PrincipalType": "Account",  
      "Principal": "arn:aws:iam::123456789012:root"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribeVpcEndpointServicePermissions](#) in der AWS CLI Befehlsreferenz.

describe-vpc-endpoint-services

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-endpoint-services`.

AWS CLI

Beispiel 1: Um alle VPC Endpunktdienste zu beschreiben

Das folgende "describe-vpc-endpoint-services" Beispiel listet alle VPC Endpunktdienste für eine AWS Region auf.

```
aws ec2 describe-vpc-endpoint-services
```

Ausgabe:

```
{  
  "ServiceDetails": [  
    {  
      "ServiceType": [  

```

```
        {
            "ServiceType": "Gateway"
        }
    ],
    "AcceptanceRequired": false,
    "ServiceName": "com.amazonaws.us-east-1.dynamodb",
    "VpcEndpointPolicySupported": true,
    "Owner": "amazon",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ],
    "BaseEndpointDnsNames": [
        "dynamodb.us-east-1.amazonaws.com"
    ]
},
{
    "ServiceType": [
        {
            "ServiceType": "Interface"
        }
    ],
    "PrivateDnsName": "ec2.us-east-1.amazonaws.com",
    "ServiceName": "com.amazonaws.us-east-1.ec2",
    "VpcEndpointPolicySupported": false,
    "Owner": "amazon",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ],
    "AcceptanceRequired": false,
    "BaseEndpointDnsNames": [
        "ec2.us-east-1.vpce.amazonaws.com"
    ]
},
{
```



```

    "ServiceType": [
      {
        "ServiceType": "Interface"
      }
    ],
    "PrivateDnsName": "ssm.us-east-1.amazonaws.com",
    "ServiceName": "com.amazonaws.us-east-1.ssm",
    "VpcEndpointPolicySupported": true,
    "Owner": "amazon",
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1c",
      "us-east-1d",
      "us-east-1e"
    ],
    "AcceptanceRequired": false,
    "BaseEndpointDnsNames": [
      "ssm.us-east-1.vpce.amazonaws.com"
    ]
  }
],
"ServiceNames": [
  "com.amazonaws.us-east-1.dynamodb",
  "com.amazonaws.us-east-1.ec2",
  "com.amazonaws.us-east-1.ec2messages",
  "com.amazonaws.us-east-1.elasticloadbalancing",
  "com.amazonaws.us-east-1.kinesis-streams",
  "com.amazonaws.us-east-1.s3",
  "com.amazonaws.us-east-1.ssm"
]
}

```

Weitere Informationen finden Sie unter [Verfügbare AWS Dienstnamen anzeigen](#) im Benutzerhandbuch für AWS PrivateLink.

Beispiel 2: Um die Details zu einem Endpunktdienst zu beschreiben

Das folgende Beispiel `describe-vpc-endpoint-services` listet die Details des Amazon S3 S3-Schnittstellen-Endpunktdienstes auf

```
aws ec2 describe-vpc-endpoint-services \
```

```
--filter "Name=service-type,Values=Interface" Name=service-name,Values=com.amazonaws.us-east-1.s3
```

Ausgabe:

```
{
  "ServiceDetails": [
    {
      "ServiceName": "com.amazonaws.us-east-1.s3",
      "ServiceId": "vpce-svc-081d84efcdEXAMPLE",
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ],
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ],
      "Owner": "amazon",
      "BaseEndpointDnsNames": [
        "s3.us-east-1.vpce.amazonaws.com"
      ],
      "VpcEndpointPolicySupported": true,
      "AcceptanceRequired": false,
      "ManagesVpcEndpoints": false,
      "Tags": []
    }
  ],
  "ServiceNames": [
    "com.amazonaws.us-east-1.s3"
  ]
}
```

Weitere Informationen finden Sie unter [Verfügbare AWS Servicenamen anzeigen](#) im Benutzerhandbuch für AWS PrivateLink.

- API-Einheiten finden Sie [DescribeVpcEndpointServices](#) in der AWS CLI Befehlsreferenz.

describe-vpc-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-endpoints`.

AWS CLI

Um Ihre VPC Endpunkte zu beschreiben

Im folgenden `describe-vpc-endpoints` Beispiel werden Details zu all Ihren VPC Endpunkten angezeigt.

```
aws ec2 describe-vpc-endpoints
```

Ausgabe:

```
{
  "VpcEndpoints": [
    {
      "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}",
      "VpcId": "vpc-aabb1122",
      "NetworkInterfaceIds": [],
      "SubnetIds": [],
      "PrivateDnsEnabled": true,
      "State": "available",
      "ServiceName": "com.amazonaws.us-east-1.dynamodb",
      "RouteTableIds": [
        "rtb-3d560345"
      ],
      "Groups": [],
      "VpcEndpointId": "vpce-032a826a",
      "VpcEndpointType": "Gateway",
      "CreationTimestamp": "2017-09-05T20:41:28Z",
      "DnsEntries": [],
      "OwnerId": "123456789012"
    },
    {
      "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*\n\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource\n\": \"*\"\n    }\n  ]\n}",
      "VpcId": "vpc-1a2b3c4d",
      "NetworkInterfaceIds": [
        "eni-2ec2b084",

```

```
        "eni-1b4a65cf"
    ],
    "SubnetIds": [
        "subnet-d6fcaa8d",
        "subnet-7b16de0c"
    ],
    "PrivateDnsEnabled": false,
    "State": "available",
    "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
    "RouteTableIds": [],
    "Groups": [
        {
            "GroupName": "default",
            "GroupId": "sg-54e8bf31"
        }
    ],
    "VpcEndpointId": "vpce-0f89a33420c1931d7",
    "VpcEndpointType": "Interface",
    "CreationTimestamp": "2017-09-05T17:55:27.583Z",
    "DnsEntries": [
        {
            "HostedZoneId": "Z7HUB22UULQXV",
            "DnsName": "vpce-0f89a33420c1931d7-
bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
        },
        {
            "HostedZoneId": "Z7HUB22UULQXV",
            "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1b.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
        },
        {
            "HostedZoneId": "Z7HUB22UULQXV",
            "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
        }
    ],
    "OwnerId": "123456789012"
},
{
    "VpcEndpointId": "vpce-aabbaabbaabbaabba",
    "VpcEndpointType": "GatewayLoadBalancer",
    "VpcId": "vpc-111122223333aabbc",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-123123a1c43abc123",
```

```
    "State": "available",
    "SubnetIds": [
      "subnet-0011aabbcc2233445"
    ],
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
      "eni-01010120203030405"
    ],
    "CreationTimestamp": "2020-11-11T08:06:03.522Z",
    "Tags": [],
    "OwnerId": "123456789012"
  }
]
}
```

Weitere Informationen finden Sie unter [VPCEndpoints](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeVpcEndpoints](#) in der AWS CLI Befehlsreferenz.

describe-vpc-peering-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-vpc-peering-connections`.

AWS CLI

Um Ihre VPC Peering-Verbindungen zu beschreiben

In diesem Beispiel werden alle Ihre VPC Peering-Verbindungen beschrieben.

Befehl:

```
aws ec2 describe-vpc-peering-connections
```

Ausgabe:

```
{
  "VpcPeeringConnections": [
    {
      "Status": {
        "Message": "Active",
        "Code": "active"
      },
      "Tags": [
```

```
        {
            "Value": "Peering-1",
            "Key": "Name"
        }
    ],
    "AcceptorVpcInfo": {
        "OwnerId": "111122223333",
        "VpcId": "vpc-1a2b3c4d",
        "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-11122233",
    "RequesterVpcInfo": {
        "PeeringOptions": {
            "AllowEgressFromLocalVpcToRemoteClassicLink": false,
            "AllowEgressFromLocalClassicLinkToRemoteVpc": false
        },
        "OwnerId": "444455556666",
        "VpcId": "vpc-123abc45",
        "CidrBlock": "192.168.0.0/16"
    }
},
{
    "Status": {
        "Message": "Pending Acceptance by 444455556666",
        "Code": "pending-acceptance"
    },
    "Tags": [],
    "RequesterVpcInfo": {
        "PeeringOptions": {
            "AllowEgressFromLocalVpcToRemoteClassicLink": false,
            "AllowEgressFromLocalClassicLinkToRemoteVpc": false
        },
        "OwnerId": "444455556666",
        "VpcId": "vpc-11aa22bb",
        "CidrBlock": "10.0.0.0/28"
    },
    "VpcPeeringConnectionId": "pcx-abababab",
    "ExpirationTime": "2014-04-03T09:12:43.000Z",
    "AcceptorVpcInfo": {
        "OwnerId": "444455556666",
        "VpcId": "vpc-33cc44dd"
    }
}
]
```

```
}
```

Um bestimmte VPC Peering-Verbindungen zu beschreiben

In diesem Beispiel werden all Ihre VPC Peering-Verbindungen beschrieben, die sich im Status „Ausstehende Annahme“ befinden.

Befehl:

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

In diesem Beispiel werden alle Ihre VPC Peering-Verbindungen beschrieben, die das Tag Owner=Finance haben.

Befehl:

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

Dieses Beispiel beschreibt alle VPC Peering-Verbindungen, die Sie für den angegebenen vpc-1a2b3c4d angefordert haben. VPC

Befehl:

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- Einzelheiten API finden Sie unter Befehlsreferenz. [DescribeVpcPeeringConnections](#)AWS CLI

describe-vpcs

Das folgende Codebeispiel zeigt die Verwendung `describe-vpcs`.

AWS CLI

Beispiel 1: Um all deine zu beschreiben VPCs

Im folgenden `describe-vpcs` Beispiel werden Details zu Ihrem VPCs abgerufen.

```
aws ec2 describe-vpcs
```

Ausgabe:

```
{
  "Vpcs": [
    {
      "CidrBlock": "30.1.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-0e9801d129EXAMPLE",
      "OwnerId": "111122223333",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-062c64cfafEXAMPLE",
          "CidrBlock": "30.1.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": false,
      "Tags": [
        {
          "Key": "Name",
          "Value": "Not Shared"
        }
      ]
    },
    {
      "CidrBlock": "10.0.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-06e4ab6c6cEXAMPLE",
      "OwnerId": "222222222222",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
          "CidrBlock": "10.0.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ]
    }
  ],
}
```



```

        "IsDefault": false,
        "Tags": [
            {
                "Key": "Name",
                "Value": "Shared VPC"
            }
        ]
    }
]
}

```

Beispiel 2: Um ein bestimmtes zu beschreiben VPC

Im folgenden `describe-vpcs` Beispiel werden Details für das angegebene VPC Objekt abgerufen.

```

aws ec2 describe-vpcs \
  --vpc-ids vpc-06e4ab6c6cEXAMPLE

```

Ausgabe:

```

{
  "Vpcs": [
    {
      "CidrBlock": "10.0.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-06e4ab6c6cEXAMPLE",
      "OwnerId": "111122223333",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
          "CidrBlock": "10.0.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": false,
      "Tags": [
        {

```

```

        "Key": "Name",
        "Value": "Shared VPC"
      }
    ]
  }
}

```

- API-Einheiten finden Sie unter [DescribeVpcs AWS CLI-Befehlsreferenz](#).

describe-vpn-connections

Das folgende Codebeispiel zeigt die Verwendung `describe-vpn-connections`.

AWS CLI

Beispiel 1: Um Ihre VPN-Verbindungen zu beschreiben

Das folgende `describe-vpn-connections` Beispiel beschreibt alle Ihre Site-to-Site VPN-Verbindungen.

```
aws ec2 describe-vpn-connections
```

Ausgabe:

```

{
  "VpnConnections": [
    {
      "CustomerGatewayConfiguration": "...configuration information...",
      "CustomerGatewayId": "cgw-01234567abcde1234",
      "Category": "VPN",
      "State": "available",
      "Type": "ipsec.1",
      "VpnConnectionId": "vpn-1122334455aabbccd",
      "TransitGatewayId": "tgw-00112233445566aab",
      "Options": {
        "EnableAcceleration": false,
        "StaticRoutesOnly": true,
        "LocalIpv4NetworkCidr": "0.0.0.0/0",
        "RemoteIpv4NetworkCidr": "0.0.0.0/0",
        "TunnelInsideIpVersion": "ipv4"
      }
    }
  ]
}

```

```

    },
    "Routes": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "CanadaVPN"
      }
    ],
    "VgwTelemetry": [
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2020-07-29T10:35:11.000Z",
        "OutsideIpAddress": "203.0.113.3",
        "Status": "DOWN",
        "StatusMessage": ""
      },
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2020-09-02T09:09:33.000Z",
        "OutsideIpAddress": "203.0.113.5",
        "Status": "UP",
        "StatusMessage": ""
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [So AWS Site-to-Site VPN funktioniert](#) es.

Beispiel 2: Um Ihre verfügbaren VPN Verbindungen zu beschreiben

Das folgende `describe-vpn-connections` Beispiel beschreibt Ihre Site-to-Site VPN Verbindungen mit einem Status von `available`.

```

aws ec2 describe-vpn-connections \
  --filters "Name=state,Values=available"

```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [So AWS Site-to-Site VPN funktioniert](#) es.

- API Einheiten finden Sie [DescribeVpnConnections](#) in der AWS CLI Befehlsreferenz.

describe-vpn-gateways

Das folgende Codebeispiel zeigt die Verwendung `describe-vpn-gateways`.

AWS CLI

Um Ihre virtuellen privaten Gateways zu beschreiben

Dieses Beispiel beschreibt Ihre virtuellen privaten Gateways.

Befehl:

```
aws ec2 describe-vpn-gateways
```

Ausgabe:

```
{
  "VpnGateways": [
    {
      "State": "available",
      "Type": "ipsec.1",
      "VpnGatewayId": "vgw-f211f09b",
      "VpcAttachments": [
        {
          "State": "attached",
          "VpcId": "vpc-98eb5ef5"
        }
      ]
    },
    {
      "State": "available",
      "Type": "ipsec.1",
      "VpnGatewayId": "vgw-9a4cacf3",
      "VpcAttachments": [
        {
          "State": "attaching",
          "VpcId": "vpc-a01106c2"
        }
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeVpnGateways](#) in der AWS CLI Befehlsreferenz.

detach-classic-link-vpc

Das folgende Codebeispiel zeigt die Verwendung `detach-classic-link-vpc`.

AWS CLI

Um die Verknüpfung einer EC2 -Classic-Instanz von einer aufzuheben (zu trennen) VPC

In diesem Beispiel wird die Verknüpfung der Instanz `i-0598c7d356eba48d7` mit `vpc-88888888` aufgehoben. VPC

Befehl:

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-id vpc-88888888
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden [DetachClassicLinkVpc](#) Sie unter Befehlsreferenz. AWS CLI

detach-internet-gateway

Das folgende Codebeispiel zeigt die Verwendung `detach-internet-gateway`.

AWS CLI

Um ein Internet-Gateway von Ihrem zu trennen VPC

Im folgenden `detach-internet-gateway` Beispiel wird das angegebene Internet-Gateway vom spezifischen getrennt. VPC

```
aws ec2 detach-internet-gateway \
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Internet-Gateways](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DetachInternetGateway](#) in der AWS CLI Befehlsreferenz.

detach-network-interface

Das folgende Codebeispiel zeigt die Verwendung `detach-network-interface`.

AWS CLI

Um eine Netzwerkschnittstelle von Ihrer Instance zu trennen

In diesem Beispiel wird die angegebene Netzwerkschnittstelle von der angegebenen Instanz getrennt. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- APIEinzelheiten finden Sie [DetachNetworkInterface](#) in der AWS CLI Befehlsreferenz.

detach-verified-access-trust-provider

Das folgende Codebeispiel zeigt die Verwendung `detach-verified-access-trust-provider`.

AWS CLI

Um einen Trust Provider von einer Instance zu trennen

Im folgenden `detach-verified-access-trust-provider` Beispiel wird der angegebene Verified Access-Vertrauensanbieter von der angegebenen Verified Access-Instanz getrennt.

```
aws ec2 detach-verified-access-trust-provider \  
  --verified-access-instance-id vai-0ce000c0b7643abea \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Ausgabe:

```
{  
  "VerifiedAccessTrustProvider": {
```

```
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
    "Description": "Testing Verified Access",
    "TrustProviderType": "user",
    "UserTrustProviderType": "iam-identity-center",
    "PolicyReferenceName": "idc",
    "CreationTime": "2023-08-25T19:00:38",
    "LastUpdatedTime": "2023-08-25T19:00:38"
  },
  "VerifiedAccessInstance": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "VerifiedAccessTrustProviders": [],
    "CreationTime": "2023-08-25T18:27:56",
    "LastUpdatedTime": "2023-08-25T18:27:56"
  }
}
```

Weitere Informationen finden Sie unter [Verified Access-Instanzen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [DetachVerifiedAccessTrustProvider](#) in der AWS CLI Befehlsreferenz.

detach-volume

Das folgende Codebeispiel zeigt die Verwendung `detach-volume`.

AWS CLI

Um ein Volume von einer Instance zu trennen

Dieser Beispielbefehl trennt das Volume (`vol-049df61146c4d7901`) von der Instance, an die es angehängt ist.

Befehl:

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

Ausgabe:

```
{
  "AttachTime": "2014-02-27T19:23:06.000Z",
  "InstanceId": "i-1234567890abcdef0",
```

```
"VolumeId": "vol-049df61146c4d7901",  
"State": "detaching",  
"Device": "/dev/sdb"  
}
```

- API-Einheiten finden Sie [DetachVolume](#) in der AWS CLI Befehlsreferenz.

detach-vpn-gateway

Das folgende Codebeispiel zeigt die Verwendung `detach-vpn-gateway`.

AWS CLI

Um ein virtuelles privates Gateway von Ihrem zu trennen VPC

In diesem Beispiel wird das angegebene virtuelle private Gateway vom angegebenen getrennt. VPC Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- API-Einheiten finden Sie unter [DetachVpnGateway AWS CLI](#) Befehlsreferenz.

disable-address-transfer

Das folgende Codebeispiel zeigt die Verwendung `disable-address-transfer`.

AWS CLI

Um eine Elastic IP-Adressübertragung zu deaktivieren

Im folgenden `disable-address-transfer` Beispiel wird die Elastic IP-Adressübertragung für die angegebene Elastic IP-Adresse deaktiviert.

```
aws ec2 disable-address-transfer \  
--allocation-id eipalloc-09ad461b0d03f6aaf
```

Ausgabe:

```
{
```



```
"AddressTransfer": {
  "PublicIp": "100.21.184.216",
  "AllocationId": "eipalloc-09ad461b0d03f6aaf",
  "AddressTransferStatus": "disabled"
}
```

Weitere Informationen finden Sie unter [Transfer Elastic IP-Adressen](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DisableAddressTransfer](#) in der AWS CLI Befehlsreferenz.

disable-aws-network-performance-metric-subscription

Das folgende Codebeispiel zeigt die Verwendung `disable-aws-network-performance-metric-subscription`.

AWS CLI

Um ein Metrik-Abonnement zu deaktivieren

Im folgenden `disable-aws-network-performance-metric-subscription` Beispiel wird die Überwachung der aggregierten Netzwerklatenz zwischen den angegebenen Quell- und Zielregionen deaktiviert.

```
aws ec2 disable-aws-network-performance-metric-subscription \
  --source us-east-1 \
  --destination eu-west-1 \
  --metric aggregate-latency \
  --statistic p50
```

Ausgabe:

```
{
  "Output": true
}
```

Weitere Informationen finden Sie unter [Abonnements verwalten](#) im Infrastructure Performance User Guide.

- APIEinzelheiten finden Sie [DisableAwsNetworkPerformanceMetricSubscription](#) in der AWS CLI Befehlsreferenz.

disable-ebs-encryption-by-default

Das folgende Codebeispiel zeigt die Verwendung `disable-ebs-encryption-by-default`.

AWS CLI

Um die EBS Verschlüsselung standardmäßig zu deaktivieren

Im folgenden `disable-ebs-encryption-by-default` Beispiel wird die EBS Verschlüsselung standardmäßig für Ihr AWS Konto in der aktuellen Region deaktiviert.

```
aws ec2 disable-ebs-encryption-by-default
```

Ausgabe:

```
{
  "EbsEncryptionByDefault": false
}
```

- API-Einheiten finden Sie [DisableEbsEncryptionByDefault](#) in der AWS CLI Befehlsreferenz.

disable-fast-launch

Das folgende Codebeispiel zeigt die Verwendung `disable-fast-launch`.

AWS CLI

Um den Schnellstart für ein Image abzubrechen

Im folgenden `disable-fast-launch` Beispiel wird der Schnellstart auf den angegebenen Geräten eingestellt und vorhandene AMI, vorab bereitgestellte Snapshots bereinigt.

```
aws ec2 disable-fast-launch \
  --image-id ami-01234567890abcdef
```

Ausgabe:

```
{
  "ImageId": "ami-01234567890abcdef",
  "ResourceType": "snapshot",
  "SnapshotConfiguration": {},
}
```

```

    "LaunchTemplate": {
      "LaunchTemplateId": "lt-01234567890abcdef",
      "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
      "Version": "1"
    },
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "disabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"
  }
}

```

Weitere Informationen zur Konfiguration eines Windows AMI für einen schnelleren Start finden [Sie unter Konfiguration AMI für einen schnelleren Start](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DisableFastLaunch](#) in der AWS CLI Befehlsreferenz.

disable-fast-snapshot-restores

Das folgende Codebeispiel zeigt die Verwendung `disable-fast-snapshot-restores`.

AWS CLI

Um die schnelle Snapshot-Wiederherstellung zu deaktivieren

Im folgenden `disable-fast-snapshot-restores` Beispiel wird die schnelle Snapshot-Wiederherstellung für den angegebenen Snapshot in der angegebenen Availability Zone deaktiviert.

```

aws ec2 disable-fast-snapshot-restores \
  --availability-zones us-east-2a \
  --source-snapshot-ids snap-1234567890abcdef0

```

Ausgabe:

```

{
  "Successful": [
    {
      "SnapshotId": "snap-1234567890abcdef0"
      "AvailabilityZone": "us-east-2a",
      "State": "disabling",
    }
  ]
}

```

```

        "StateTransitionReason": "Client.UserInitiated",
        "OwnerId": "123456789012",
        "EnablingTime": "2020-01-25T23:57:49.602Z"
    }
],
"Unsuccessful": []
}

```

- API-Einheiten finden Sie unter [DisableFastSnapshotRestores AWS CLI Befehlsreferenz](#).

disable-image-block-public-access

Das folgende Codebeispiel zeigt die Verwendung `disable-image-block-public-access`.

AWS CLI

Um den öffentlichen Zugriff AMIs in der angegebenen Region zu deaktivieren

Im folgenden `disable-image-block-public-access` Beispiel wird die Sperrung des öffentlichen Zugriffs AMIs auf Kontoebene in der angegebenen Region deaktiviert.

```

aws ec2 disable-image-block-public-access \
  --region us-east-1

```

Ausgabe:

```

{
  "ImageBlockPublicAccessState": "unblocked"
}

```

Weitere Informationen finden Sie unter [Sperren des öffentlichen Zugriffs auf Ihr AMIs](#) im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DisableImageBlockPublicAccess](#) in der AWS CLI Befehlsreferenz.

disable-image-deprecation

Das folgende Codebeispiel zeigt die Verwendung `disable-image-deprecation`.

AWS CLI

Um die Veralterung eines aufzuheben AMI

Im folgenden `disable-image-deprecation` Beispiel wird die Veralterung von an aufgehobenAMI, wodurch das Feld aus der Ausgabe entfernt wird. `DeprecationTime describe-images` Sie müssen der AMI Besitzer sein, um dieses Verfahren ausführen zu können.

```
aws ec2 disable-image-deprecation \  
  --image-id ami-1234567890abcdef0
```

Ausgabe:

```
{  
  "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",  
  "Return": "true"  
}
```

Weitere Informationen finden Sie unter [Deprecate an AMI <https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html>](https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html) im Amazon-Benutzerhandbuch. EC2

- APIEinzelheiten finden Sie in der Befehlsreferenz [DisableImageDeprecation](#).AWS CLI

disable-image

Das folgende Codebeispiel zeigt die Verwendung `disable-image`.

AWS CLI

Um ein zu deaktivieren AMI

Das folgende `disable-image` Beispiel deaktiviert das angegebeneAMI.

```
aws ec2 disable-image \  
  --image-id ami-1234567890abcdef0
```

Ausgabe:

```
{  
  "Return": "true"  
}
```

Weitere Informationen finden [Sie unter Deaktivieren](#) von AMI im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DisableImage](#) in der AWS CLI Befehlsreferenz.

disable-ipam-organization-admin-account

Das folgende Codebeispiel zeigt die Verwendung `disable-ipam-organization-admin-account`.

AWS CLI

Um den delegierten Administrator IPAM zu deaktivieren

In bestimmten Szenarien werden Sie eine Integration IPAM mit AWS Organizations durchführen. Wenn Sie das tun, delegiert das AWS Organisationsverwaltungskonto ein AWS Organisations-Mitgliedskonto als IPAM Administrator.

In diesem Beispiel sind Sie das Verwaltungskonto der AWS Organizations, das das IPAM Administratorkonto delegiert hat, und Sie möchten dieses Konto als IPAM Administrator deaktivieren.

Sie können eine beliebige AWS Region für verwenden `--region`, wenn Sie diese Anfrage stellen. Sie müssen nicht die Region verwenden, in der Sie den Administrator ursprünglich delegiert haben, in der der erstellt IPAM wurde, oder eine Region, in der der Administrator IPAM aktiv ist. Wenn Sie das delegierte Administratorkonto deaktivieren, können Sie es jederzeit wieder aktivieren oder ein neues Konto als Administrator delegieren. IPAM

Im folgenden `disable-ipam-organization-admin-account` Beispiel wird der delegierte IPAM Administrator in Ihrem Konto deaktiviert. AWS

```
aws ec2 disable-ipam-organization-admin-account \
  --delegated-admin-account-id 320805250157 \
  --region ap-south-1
```

Ausgabe:

```
{
  "Success": true
}
```

Weitere Informationen finden Sie unter [Integration IPAM mit Konten in einer AWS Organisation](#) im VPCIPAM Amazon-Benutzerhandbuch.

- API Einheiten finden Sie [DisableIpamOrganizationAdminAccount](#) in der AWS CLI Befehlsreferenz.

disable-serial-console-access

Das folgende Codebeispiel zeigt die Verwendung `disable-serial-console-access`.

AWS CLI

Um den Zugriff auf die EC2 serielle Konsole für Ihr Konto zu deaktivieren

Im folgenden `disable-serial-console-access` Beispiel wird der Kontozugriff auf die serielle Konsole deaktiviert.

```
aws ec2 disable-serial-console-access
```

Ausgabe:

```
{
  "SerialConsoleAccessEnabled": false
}
```

Weitere Informationen finden Sie unter [EC2Serial Console](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DisableSerialConsoleAccess](#) in der AWS CLI Befehlsreferenz.

disable-snapshot-block-public-access

Das folgende Codebeispiel zeigt die Verwendung `disable-snapshot-block-public-access`.

AWS CLI

Um den öffentlichen Zugriff für Snapshots zu deaktivieren

Im folgenden `disable-snapshot-block-public-access` Beispiel wird „Öffentlichen Zugriff blockieren“ für Schnappschüsse deaktiviert, damit Sie Ihre Schnappschüsse öffentlich teilen können.

```
aws ec2 disable-snapshot-block-public-access
```

Ausgabe:

```
{
  "State": "unblocked"
}
```

```
}
```

Weitere Informationen finden Sie unter [Sperren des öffentlichen Zugriffs für Snapshots](#) im EBSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DisableSnapshotBlockPublicAccess](#) in der AWS CLI Befehlsreferenz.

disable-transit-gateway-route-table-propagation

Das folgende Codebeispiel zeigt die Verwendung `disable-transit-gateway-route-table-propagation`.

AWS CLI

Um eine Transit-Gateway-Verbindung zu deaktivieren, um Routen an die angegebene Propagierungsroutentabelle weiterzuleiten

Im folgenden `disable-transit-gateway-route-table-propagation` Beispiel wird die angegebene Anlage deaktiviert, um Routen an die angegebene Propagierungsroutentabelle weiterzugeben.

```
aws ec2 disable-transit-gateway-route-table-propagation \  
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Ausgabe:

```
{  
  "Propagation": {  
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
    "ResourceId": "vpc-4d7de228",  
    "ResourceType": "vpc",  
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",  
    "State": "disabled"  
  }  
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- APIEinzelheiten finden Sie unter [DisableTransitGatewayRouteTablePropagation AWS CLIBefehlsreferenz](#).

disable-vgw-route-propagation

Das folgende Codebeispiel zeigt die Verwendung `disable-vgw-route-propagation`.

AWS CLI

Um die Route-Propagierung zu deaktivieren

In diesem Beispiel wird verhindert, dass das angegebene Virtual Private Gateway statische Routen an die angegebene Routentabelle weitergibt. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- API-Einheiten finden Sie unter [DisableVgwRoutePropagation AWS CLI Befehlsreferenz](#).

disable-vpc-classic-link-dns-support

Das folgende Codebeispiel zeigt die Verwendung `disable-vpc-classic-link-dns-support`.

AWS CLI

Um die ClassicLink DNS Unterstützung für ein zu deaktivieren VPC

In diesem Beispiel wird die ClassicLink DNS Unterstützung für `vpc-88888888` deaktiviert.

Befehl:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden Sie [DisableVpcClassicLinkDnsSupport](#) in der AWS CLI Befehlsreferenz.

disable-vpc-classic-link

Das folgende Codebeispiel zeigt die Verwendung `disable-vpc-classic-link`.

AWS CLI

Zum Deaktivieren ClassicLink für ein VPC

Dieses Beispiel deaktiviert ClassicLink für `vpc-88888888`.

Befehl:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

Ausgabe:

```
{  
  "Return": true  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz. [DisableVpcClassicLink](#) AWS CLI

disassociate-address

Das folgende Codebeispiel zeigt die Verwendung `disassociate-address`.

AWS CLI

Um die Zuordnung einer Elastic IP-Adresse in EC2 -Classic aufzuheben

In diesem Beispiel wird die Zuordnung einer Elastic IP-Adresse zu einer Instanz in -Classic getrennt. EC2 Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

Um die Zuordnung einer Elastic IP-Adresse in - zu trennen EC2 VPC

In diesem Beispiel wird die Zuordnung einer Elastic IP-Adresse zu einer Instance in a getrennt. VPC Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- API-Einheiten finden Sie [DisassociateAddress](#) in der AWS CLI Befehlsreferenz.

disassociate-client-vpn-target-network

Das folgende Codebeispiel zeigt die Verwendung `disassociate-client-vpn-target-network`.

AWS CLI

Um ein Netzwerk von einem VPN Client-Endpunkt zu trennen

Im folgenden `disassociate-client-vpn-target-network` Beispiel wird die Zuordnung des Zielnetzwerks aufgehoben, das der `cvpn-assoc-12312312312312312` Zuordnungs-ID für den angegebenen VPN Client-Endpunkt zugeordnet ist.

```
aws ec2 disassociate-client-vpn-target-network \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --association-id cvpn-assoc-12312312312312312
```

Ausgabe:

```
{  
  "AssociationId": "cvpn-assoc-12312312312312312",  
  "Status": {  
    "Code": "disassociating"  
  }  
}
```

Weitere Informationen finden Sie unter [Zielnetzwerke](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [DisassociateClientVpnTargetNetwork](#) in der AWS CLI Befehlsreferenz.

disassociate-iam-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `disassociate-iam-instance-profile`.

AWS CLI

Um die Zuordnung eines IAM Instanzprofils aufzuheben

In diesem Beispiel wird die Zuordnung eines IAM Instanzprofils zur Zuordnungs-ID aufgehoben.

`iip-assoc-05020b59952902f5f`

Befehl:

```
aws ec2 disassociate-iam-instance-profile --association-id iip-  
assoc-05020b59952902f5f
```

Ausgabe:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-123456789abcde123",  
    "State": "disassociating",  
    "AssociationId": "iip-assoc-05020b59952902f5f",  
    "IamInstanceProfile": {  
      "Id": "AIPAI5IVIHMFYY2DKV5Y",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"  
    }  
  }  
}
```

- API-Einheiten finden Sie unter [DisassociateIamInstanceProfile AWS CLI Befehlsreferenz](#).

disassociate-instance-event-window

Das folgende Codebeispiel zeigt die Verwendung `disassociate-instance-event-window`.

AWS CLI

Beispiel 1: Um die Zuordnung einer oder mehrerer Instanzen zu einem Ereignisfenster zu trennen

Im folgenden `disassociate-instance-event-window` Beispiel wird die Zuordnung einer oder mehrerer Instanzen zu einem Ereignisfenster getrennt. Geben Sie den `instance-event-window-id` Parameter zur Angabe des Ereignisfensters an. Um die Zuordnung von Instanzen aufzuheben, geben Sie den `association-target` Parameter an, und geben Sie für die Parameterwerte eine oder mehrere Instanzen IDs an.

```
aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Ausgabe:

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 2: So trennen Sie die Zuordnung von Instance-Tags zu einem Ereignisfenster

Im folgenden `disassociate-instance-event-window` Beispiel wird die Zuordnung von Instanz-Tags zu einem Ereignisfenster aufgehoben. Geben Sie den `instance-event-window-id` Parameter an, um das Ereignisfenster zu spezifizieren. Um die Zuordnung der Instance-Tags (Markierungen) aufzuheben, geben Sie den `association-target`-Parameter und für die Parameterwerte ein oder mehrere Tags (Markierungen) an.

```
aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"
```

Ausgabe:

```
{
  "InstanceEventWindow": {
```

```

    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 3: So trennen Sie die Zuordnung eines Dedicated Hosts zu einem Eventfenster

Im folgenden `disassociate-instance-event-window` Beispiel wird die Zuordnung eines Dedicated Hosts zu einem Ereignisfenster getrennt. Geben Sie den `instance-event-window-id` Parameter zur Angabe des Ereignisfensters an. Um die Zuordnung eines Dedicated Hosts aufzuheben, geben Sie den `association-target` Parameter und für die Parameterwerte einen oder mehrere Dedicated Host IDs an.

```

aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target DedicatedHostIds=h-029fa35a02b99801d

```

Ausgabe:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

```
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

- API-Einheiten finden Sie [DisassociateInstanceEventWindow](#) in der AWS CLI Befehlsreferenz.

disassociate-ipam-resource-discovery

Das folgende Codebeispiel zeigt die Verwendung `disassociate-ipam-resource-discovery`.

AWS CLI

Um eine Ressourcenerkennung von einem zu trennen IPAM

In diesem Beispiel sind Sie ein IPAM delegiertes Administratorkonto und möchten eine IPAM Ressourcenerkennung von Ihrem Konto trennen. IPAM Sie haben den Befehl `describe` ausgeführt und festgestellt, dass der `"ResourceDiscoveryStatus": "not-found"` und Sie ihn von Ihrem trennen möchten, um Platz für andere Zuordnungen IPAM zu schaffen.

Im folgenden `disassociate-ipam-resource-discovery` Beispiel wird die Zuordnung einer IPAM Ressourcensuche in Ihrem AWS Konto aufgehoben.

```
aws ec2 disassociate-ipam-resource-discovery \
  --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82 \
  --region us-east-1
```

Ausgabe:

```
{
  "IpamResourceDiscoveryAssociation": {
    "OwnerId": "320805250157",
    "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
    "IpamResourceDiscoveryAssociationArn":
"arn:aws:ec2::320805250157:ipam-resource-discovery-association/ipam-res-disco-
assoc-04382a6346357cf82",
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
    "IpamId": "ipam-005f921c17ebd5107",
    "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
    "IpamRegion": "us-east-1",
```

```
    "IsDefault": false,  
    "ResourceDiscoveryStatus": "not-found",  
    "State": "disassociate-in-progress"  
  }  
}
```

Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

- API-Einheiten finden Sie [DisassociateIpamResourceDiscovery](#) in der AWS CLI Befehlsreferenz.

disassociate-nat-gateway-address

Das folgende Codebeispiel zeigt die Verwendung `disassociate-nat-gateway-address`.

AWS CLI

Um eine Elastic IP-Adresse von einem öffentlichen NAT Gateway zu trennen

Im folgenden `disassociate-nat-gateway-address` Beispiel wird die Verbindung zwischen der angegebenen Elastic IP-Adresse und dem angegebenen öffentlichen NAT Gateway getrennt.

```
aws ec2 disassociate-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --association-ids eipassoc-0f96bdca17EXAMPLE
```

Ausgabe:

```
{  
  "NatGatewayId": "nat-1234567890abcdef0",  
  "NatGatewayAddresses": [  
    {  
      "AllocationId": "eipalloc-0be6ecac95EXAMPLE",  
      "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
      "PrivateIp": "10.0.0.74",  
      "PublicIp": "3.211.231.218",  
      "AssociationId": "eipassoc-0f96bdca17EXAMPLE",  
      "IsPrimary": false,  
      "Status": "disassociating"  
    }  
  ]  
}
```



```
}
```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DisassociateNatGatewayAddress](#) in der AWS CLI Befehlsreferenz.

disassociate-route-table

Das folgende Codebeispiel zeigt die Verwendung `disassociate-route-table`.

AWS CLI

Um die Zuordnung einer Routentabelle aufzuheben

In diesem Beispiel wird die Verbindung zwischen der angegebenen Routing-Tabelle und dem angegebenen Subnetz getrennt. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- APIEinzelheiten finden Sie unter [DisassociateRouteTable AWS CLI](#) Befehlsreferenz.

disassociate-subnet-cidr-block

Das folgende Codebeispiel zeigt die Verwendung `disassociate-subnet-cidr-block`.

AWS CLI

Um einen IPv6 CIDR Block von einem Subnetz zu trennen

In diesem Beispiel wird die Zuordnung eines IPv6 CIDR Blocks zu einem Subnetz mithilfe der Zuordnungs-ID für den Block getrennt. CIDR

Befehl:

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

Ausgabe:

```
{
```

```

    "SubnetId": "subnet-5f46ec3b",
    "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
      "Ipv6CidrBlockState": {
        "State": "disassociating"
      }
    }
  }
}

```

- API-Einheiten finden Sie unter [DisassociateSubnetCidrBlock AWS CLI](#) Befehlsreferenz.

disassociate-transit-gateway-multicast-domain

Das folgende Codebeispiel zeigt die Verwendung `disassociate-transit-gateway-multicast-domain`.

AWS CLI

Um Subnetze von einer Multicast-Domäne zu trennen

Im folgenden `disassociate-transit-gateway-multicast-domain` Beispiel wird die Zuordnung eines Subnetzes zur angegebenen Multicast-Domäne aufgehoben.

```

aws ec2 disassociate-transit-gateway-multicast-domain \
  --transit-gateway-attachment-id tgw-attach-070e571cd1EXAMPLE \
  --subnet-id subnet-000de86e3bEXAMPLE \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE

```

Ausgabe:

```

{
  "Associations": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
    "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
    "ResourceId": "vpc-7EXAMPLE",
    "ResourceType": "vpc",
    "Subnets": [
      {
        "SubnetId": "subnet-000de86e3bEXAMPLE",
        "State": "disassociating"
      }
    ]
  }
}

```

```

    ]
  }
}

```

Weitere Informationen finden Sie unter [Working with Multicast](#) im Transit Gateways Guide '.

- API-Einheiten finden Sie unter [Disassociate Transit Gateway Multicast Domain AWS CLI](#) Befehlsreferenz.

disassociate-transit-gateway-route-table

Das folgende Codebeispiel zeigt die Verwendung `disassociate-transit-gateway-route-table`.

AWS CLI

Um die Zuordnung einer Transit-Gateway-Routentabelle zu einem Ressourcenanhang zu trennen

Im folgenden `disassociate-transit-gateway-route-table` Beispiel wird die Zuordnung des angegebenen Anhangs zur Routentabelle des Transit-Gateways aufgehoben.

```

aws ec2 disassociate-transit-gateway-route-table \
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
  --transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE

```

Ausgabe:

```

{
  "Association": {
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
    "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
    "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "ResourceType": "direct-connect-gateway",
    "State": "disassociating"
  }
}

```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Disassociate Transit Gateway Route Table AWS CLI](#) Befehlsreferenz.

disassociate-vpc-cidr-block

Das folgende Codebeispiel zeigt die Verwendung `disassociate-vpc-cidr-block`.

AWS CLI

Um einen IPv6 CIDR Block von einem zu trennen VPC

In diesem Beispiel wird die Zuordnung eines IPv6 CIDR Blocks von einem getrennt, VPC wobei die Zuordnungs-ID für den CIDR Block verwendet wird.

Befehl:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

Ausgabe:

```
{
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
    "AssociationId": "vpc-cidr-assoc-eca54085",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  },
  "VpcId": "vpc-a034d6c4"
}
```

Um die Zuordnung eines IPv4 CIDR Blocks zu einem zu trennen VPC

In diesem Beispiel wird die Zuordnung eines IPv4 CIDR Blocks von einem getrennt. VPC

Befehl:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

Ausgabe:

```
{
  "CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-0287ac6b",
  }
}
```

```

    "CidrBlock": "172.18.0.0/16",
    "CidrBlockState": {
        "State": "disassociating"
    }
  },
  "VpcId": "vpc-27621243"
}

```

- API-Einheiten finden Sie unter [DisassociateVpcCidrBlock AWS CLI Befehlsreferenz](#).

enable-address-transfer

Das folgende Codebeispiel zeigt die Verwendung von `enable-address-transfer`.

AWS CLI

Um eine Elastic IP-Adressübertragung zu aktivieren

Das folgende `enable-address-transfer` Beispiel aktiviert die Elastic IP-Adressübertragung für die angegebene Elastic IP-Adresse an das angegebene Konto.

```

aws ec2 enable-address-transfer \
  --allocation-id eipalloc-09ad461b0d03f6aaf \
  --transfer-account-id 123456789012

```

Ausgabe:

```

{
  "AddressTransfer": {
    "PublicIp": "100.21.184.216",
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",
    "TransferAccountId": "123456789012",
    "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",
    "AddressTransferStatus": "pending"
  }
}

```

Weitere Informationen finden Sie unter [Transfer Elastic IP-Adressen](#) im VPC Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableAddressTransfer](#) in der AWS CLI Befehlsreferenz.

enable-aws-network-performance-metric-subscription

Das folgende Codebeispiel zeigt die Verwendung von `enable-aws-network-performance-metric-subscription`.

AWS CLI

Um ein Metrik-Abonnement zu aktivieren

Das folgende `enable-aws-network-performance-metric-subscription` Beispiel ermöglicht die Überwachung der aggregierten Netzwerklatenz zwischen den angegebenen Quell- und Zielregionen.

```
aws ec2 enable-aws-network-performance-metric-subscription \
  --source us-east-1 \
  --destination eu-west-1 \
  --metric aggregate-latency \
  --statistic p50
```

Ausgabe:

```
{
  "Output": true
}
```

Weitere Informationen finden Sie unter [Abonnements verwalten](#) im Infrastructure Performance User Guide.

- API-Einheiten finden Sie [EnableAwsNetworkPerformanceMetricSubscription](#) in der AWS CLI Befehlsreferenz.

enable-ebs-encryption-by-default

Das folgende Codebeispiel zeigt die Verwendung von `enable-ebs-encryption-by-default`.

AWS CLI

Um die EBS Verschlüsselung standardmäßig zu aktivieren

Das folgende `enable-ebs-encryption-by-default` Beispiel aktiviert standardmäßig die EBS Verschlüsselung für Ihr AWS Konto in der aktuellen Region.

```
aws ec2 enable-ebs-encryption-by-default
```

Ausgabe:

```
{
  "EbsEncryptionByDefault": true
}
```

- API-Einheiten finden Sie [EnableEbsEncryptionByDefault](#) in der AWS CLI Befehlsreferenz.

enable-fast-launch

Das folgende Codebeispiel zeigt die Verwendung von `enable-fast-launch`.

AWS CLI

Um mit dem Schnellstart für ein Bild zu beginnen

Im folgenden `enable-fast-launch` Beispiel wird der Schnellstart auf den angegebenen Werten gestartet. AMI und die maximale Anzahl parallel gestarteter Instances, die gestartet werden sollen, auf 6 festgelegt. Der Ressourcentyp, der für die Vorabbereitstellung verwendet werden soll, ist `snapshot`, AMI ist auf eingestellt, was auch der Standardwert ist.

```
aws ec2 enable-fast-launch \
  --image-id ami-01234567890abcdef \
  --max-parallel-launches 6 \
  --resource-type snapshot
```

Ausgabe:

```
{
  "ImageId": "ami-01234567890abcdef",
  "ResourceType": "snapshot",
  "SnapshotConfiguration": {
    "TargetResourceCount": 10
  },
  "LaunchTemplate": {},
  "MaxParallelLaunches": 6,
  "OwnerId": "0123456789123",
  "State": "enabling",
}
```

```

    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
  }

```

Weitere Informationen zur Konfiguration eines Windows AMI für einen schnelleren Start finden [Sie unter Konfiguration AMI für einen schnelleren Start](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [EnableFastLaunch](#) in der AWS CLI Befehlsreferenz.

enable-fast-snapshot-restores

Das folgende Codebeispiel zeigt die Verwendung `enable-fast-snapshot-restores`.

AWS CLI

Um eine schnelle Snapshot-Wiederherstellung zu ermöglichen

Das folgende `enable-fast-snapshot-restores` Beispiel aktiviert die schnelle Snapshot-Wiederherstellung für den angegebenen Snapshot in den angegebenen Availability Zones.

```

aws ec2 enable-fast-snapshot-restores \
  --availability-zones us-east-2a us-east-2b \
  --source-snapshot-ids snap-1234567890abcdef0

```

Ausgabe:

```

{
  "Successful": [
    {
      "SnapshotId": "snap-1234567890abcdef0"
      "AvailabilityZone": "us-east-2a",
      "State": "enabling",
      "StateTransitionReason": "Client.UserInitiated",
      "OwnerId": "123456789012",
      "EnablingTime": "2020-01-25T23:57:49.602Z"
    },
    {
      "SnapshotId": "snap-1234567890abcdef0"
      "AvailabilityZone": "us-east-2b",
      "State": "enabling",
      "StateTransitionReason": "Client.UserInitiated",
      "OwnerId": "123456789012",
    }
  ]
}

```



```
        "EnablingTime": "2020-01-25T23:57:49.596Z"
      }
    ],
    "Unsuccessful": []
  }
```

- API-Einheiten finden Sie [EnableFastSnapshotRestores](#) unter AWS CLI Befehlsreferenz.

enable-image-block-public-access

Das folgende Codebeispiel zeigt die Verwendung von `enable-image-block-public-access`.

AWS CLI

Um den öffentlichen Zugriff blockieren für AMIs die angegebene Region zu aktivieren

Im folgenden `enable-image-block-public-access` Beispiel wird die Sperrung des öffentlichen Zugriffs AMIs auf Kontoebene in der angegebenen Region aktiviert.

```
aws ec2 enable-image-block-public-access \
  --region us-east-1 \
  --image-block-public-access-state block-new-sharing
```

Ausgabe:

```
{
  "ImageBlockPublicAccessState": "block-new-sharing"
}
```

Weitere Informationen finden Sie unter [Sperrung des öffentlichen Zugriffs auf Ihre AMIs](#) im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableImageBlockPublicAccess](#) in der AWS CLI Befehlsreferenz.

enable-image-deprecation

Das folgende Codebeispiel zeigt die Verwendung von `enable-image-deprecation`.

AWS CLI

Beispiel 1: Um ein zu verwerfen AMI

Im folgenden `enable-image-deprecation` Beispiel wird ein an einem bestimmten Datum und zu einer bestimmten Uhrzeit als AMI veraltet eingestuft. Wenn Sie einen Wert für Sekunden angeben, EC2 rundet Amazon die Sekunden auf die nächste Minute ab. Sie müssen der AMI Eigentümer sein, um dieses Verfahren durchführen zu können.

```
aws ec2 enable-image-deprecation \  
  --image-id ami-1234567890abcdef0 \  
  --deprecate-at "2022-10-15T13:17:12.000Z"
```

Ausgabe:

```
{  
  "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",  
  "Return": "true"  
}
```

Weitere Informationen finden Sie unter [Deprecate an AMI <https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html>](https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html) im Amazon-Benutzerhandbuch. EC2

- API-Einheiten finden Sie in der Befehlsreferenz [EnableImageDeprecation](#).AWS CLI

enable-image

Das folgende Codebeispiel zeigt die Verwendung von `enable-image`.

AWS CLI

Um eine zu aktivieren AMI

Das folgende `enable-image` Beispiel aktiviert das angegebene AMI.

```
aws ec2 enable-image \  
  --image-id ami-1234567890abcdef0
```

Ausgabe:

```
{  
  "Return": "true"  
}
```

Weitere Informationen finden Sie unter [Deaktivieren](#) von AMI im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableImage](#) in der AWS CLI Befehlsreferenz.

enable-ipam-organization-admin-account

Das folgende Codebeispiel zeigt die Verwendung `enable-ipam-organization-admin-account`.

AWS CLI

Um sich mit AWS Organizations zu integrieren und ein Mitgliedskonto als Konto zu delegieren
IPAM

Das folgende `enable-ipam-organization-admin-account` Beispiel integriert IPAM sich in AWS Organizations und delegiert ein Mitgliedskonto als IPAM Konto.

```
aws ec2 enable-ipam-organization-admin-account \
  --delegated-admin-account-id 320805250157
```

Ausgabe:

```
{
  "Success": true
}
```

Weitere Informationen finden Sie unter [Integration IPAM mit AWS Organizations](#) im VPC/IPAM/Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableIpamOrganizationAdminAccount](#) in der AWS CLI Befehlsreferenz.

enable-reachability-analyzer-organization-sharing

Das folgende Codebeispiel zeigt die Verwendung `enable-reachability-analyzer-organization-sharing`.

AWS CLI

So aktivieren Sie den vertrauenswürdigen Zugriff für Reachability Analyzer

Das folgende `enable-reachability-analyzer-organization-sharing` Beispiel aktiviert den vertrauenswürdigen Zugriff für Reachability Analyzer.

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontoübergreifende Analysen](#) im Reachability Analyzer-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [EnableReachabilityAnalyzerOrganizationSharing](#).AWS CLI

enable-serial-console-access

Das folgende Codebeispiel zeigt die Verwendung von `enable-serial-console-access`.

AWS CLI

Um den Zugriff auf die serielle Konsole für Ihr Konto zu aktivieren

Das folgende `enable-serial-console-access` Beispiel ermöglicht den Kontozugriff auf die serielle Konsole.

```
aws ec2 enable-serial-console-access
```

Ausgabe:

```
{
  "SerialConsoleAccessEnabled": true
}
```

Weitere Informationen finden Sie unter [EC2 Serial Console](#) im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableSerialConsoleAccess](#) in der AWS CLI Befehlsreferenz.

enable-snapshot-block-public-access

Das folgende Codebeispiel zeigt die Verwendung von `enable-snapshot-block-public-access`.

AWS CLI

Um den öffentlichen Zugriff auf Snapshots zu blockieren

Das folgende `enable-snapshot-block-public-access` Beispiel blockiert das gesamte öffentliche Teilen Ihrer Schnappschüsse.

```
aws ec2 enable-snapshot-block-public-access \  
  --state block-all-sharing
```

Ausgabe:

```
{  
  "State": "block-all-sharing"  
}
```

Weitere Informationen finden Sie unter [Sperren des öffentlichen Zugriffs für Snapshots](#) im EBSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [EnableSnapshotBlockPublicAccess](#) in der AWS CLI Befehlsreferenz.

enable-transit-gateway-route-table-propagation

Das folgende Codebeispiel zeigt die Verwendung `enable-transit-gateway-route-table-propagation`.

AWS CLI

Um einem Transit-Gateway-Anhang die Weitergabe von Routen an die angegebene Propagierungsroutentabelle zu ermöglichen

Das folgende `enable-transit-gateway-route-table-propagation` Beispiel ermöglicht es der angegebenen Anlage, Routen an die angegebene Propagierungsroutentabelle weiterzugeben.

```
aws ec2 enable-transit-gateway-route-table-propagation \  
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Ausgabe:

```
{  
  "Propagation": {  
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
    "ResourceId": "vpc-4d7de228",
```

```
    "ResourceType": "vpc",
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
    "State": "disabled"
  }
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Enable Transit Gateway Route Table Propagation AWS CLI](#) Befehlsreferenz.

enable-vgw-route-propagation

Das folgende Codebeispiel zeigt die Verwendung von `enable-vgw-route-propagation`.

AWS CLI

Um die Route-Propagierung zu aktivieren

In diesem Beispiel kann das angegebene Virtual Private Gateway statische Routen an die angegebene Routentabelle weitergeben. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- API-Einheiten finden Sie unter [EnableVgwRoutePropagation AWS CLI](#) Befehlsreferenz.

enable-volume-io

Das folgende Codebeispiel zeigt die Verwendung von `enable-volume-io`.

AWS CLI

Um I/O für ein Volume zu aktivieren

In diesem Beispiel wird I/O auf dem Volume `vol-1234567890abcdef0` aktiviert.

Befehl:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden Sie [EnableVolumeIo](#) in der AWS CLI Befehlsreferenz.

enable-vpc-classic-link-dns-support

Das folgende Codebeispiel zeigt die Verwendung von `enable-vpc-classic-link-dns-support`.

AWS CLI

Um die ClassicLink DNS Unterstützung für ein zu aktivieren VPC

Dieses Beispiel aktiviert die ClassicLink DNS Unterstützung für `vpc-88888888`.

Befehl:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden Sie [EnableVpcClassicLinkDnsSupport](#) in der AWS CLI Befehlsreferenz.

enable-vpc-classic-link

Das folgende Codebeispiel zeigt die Verwendung von `enable-vpc-classic-link`.

AWS CLI

Um ein VPC für zu aktivieren ClassicLink

In diesem Beispiel wird `vpc-88888888` für aktiviert. ClassicLink

Befehl:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

Ausgabe:

```
{  
  "Return": true  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [EnableVpcClassicLink](#).AWS CLI

export-client-vpn-client-certificate-revocation-list

Das folgende Codebeispiel zeigt die Verwendung `export-client-vpn-client-certificate-revocation-list`.

AWS CLI

Um eine Sperrliste für Client-Zertifikate zu exportieren

Im folgenden `export-client-vpn-client-certificate-revocation-list` Beispiel wird die Sperrliste der Client-Zertifikate für den angegebenen VPN Client-Endpoint exportiert. In diesem Beispiel wird die Ausgabe im Textformat zurückgegeben, um das Lesen zu erleichtern.

```
aws ec2 export-client-vpn-client-certificate-revocation-list \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --output text
```

Ausgabe:

```
-----BEGIN X509 CRL-----  
MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMC  
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAwTC0lBTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd  
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI1MjA0NTIxWjCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC0lBTSBDb25z  
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft  
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
```



```
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END X509 CRL-----
STATUS      pending
```

Weitere Informationen finden Sie unter [Sperrlisten für Client-Zertifikate](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [ExportClientVpnClientCertificateRevocationList](#) in der AWS CLI Befehlsreferenz.

export-client-vpn-client-configuration

Das folgende Codebeispiel zeigt die Verwendung `export-client-vpn-client-configuration`.

AWS CLI

Um die Client-Konfiguration zu exportieren

Das folgende `export-client-vpn-client-configuration` Beispiel exportiert die Client-Konfiguration für den angegebenen VPN Client-Endpunkt. In diesem Beispiel wird die Ausgabe im Textformat zurückgegeben, um das Lesen zu erleichtern.

```
aws ec2 export-client-vpn-client-configuration \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --output text
```

Ausgabe:

```
client
dev tun
proto udp
remote cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-south-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
```

```

cipher AES-256-GCM
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAsTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAsTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEA4nU
hVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----
</ca>
reneg-sec 0

```

Weitere Informationen finden Sie unter [VPNClient-Endpoints](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [ExportClientVpnClientConfiguration](#) in der AWS CLI Befehlsreferenz.

export-image

Das folgende Codebeispiel zeigt die Verwendung `export-image`.

AWS CLI

Um eine VM aus einem zu exportieren AMI

Im folgenden `export-image` Beispiel wird der angegebene Wert AMI im angegebenen Format in den angegebenen Bucket exportiert.

```

aws ec2 export-image \
  --image-id ami-1234567890abcdef0 \
  --disk-image-format VMDK \
  --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/

```

Ausgabe:

```
{
  "DiskImageFormat": "vmdk",
  "ExportImageTaskId": "export-ami-1234567890abcdef0"
  "ImageId": "ami-1234567890abcdef0",
  "RoleName": "vmimport",
  "Progress": "0",
  "S3ExportLocation": {
    "S3Bucket": "my-export-bucket",
    "S3Prefix": "exports/"
  },
  "Status": "active",
  "StatusMessage": "validating"
}
```

- API-Einheiten finden Sie [ExportImage](#) unter AWS CLI Befehlsreferenz.

get-associated-ipv6-pool-cidrs

Das folgende Codebeispiel zeigt die Verwendung `get-associated-ipv6-pool-cidrs`.

AWS CLI

Um die Verknüpfungen für einen IPv6 Adresspool abzurufen

Im folgenden `get-associated-ipv6-pool-cidrs` Beispiel werden die Verknüpfungen für den angegebenen IPv6 Adresspool abgerufen.

```
aws ec2 get-associated-ipv6-pool-cidrs \
  --pool-id ipv6pool-ec2-012345abc12345abc
```

Ausgabe:

```
{
  "Ipv6CidrAssociations": [
    {
      "Ipv6Cidr": "2001:db8:1234:1a00::/56",
      "AssociatedResource": "vpc-111111222222333ab"
    }
  ]
}
```

- API-Einheiten finden Sie unter [GetAssociatedIpv6 PoolCidrs](#) in der AWS CLI Befehlsreferenz.

get-aws-network-performance-data

Das folgende Codebeispiel zeigt die Verwendung get-aws-network-performance-data.

AWS CLI

Um Netzwerkleistungsdaten abzurufen

Im folgenden get-aws-network-performance-data Beispiel werden Daten zur Netzwerkleistung zwischen den angegebenen Regionen im angegebenen Zeitraum abgerufen.

```
aws ec2 get-aws-network-performance-data \
  --start-time 2022-10-26T12:00:00.000Z \
  --end-time 2022-10-26T12:30:00.000Z \
  --data-queries Id=my-query,Source=us-east-1,Destination=eu-
west-1,Metric=aggregate-latency,Statistic=p50,Period=five-minutes
```

Ausgabe:

```
{
  "DataResponses": [
    {
      "Id": "my-query",
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes",
      "MetricPoints": [
        {
          "StartDate": "2022-10-26T12:00:00+00:00",
          "EndDate": "2022-10-26T12:05:00+00:00",
          "Value": 62.44349,
          "Status": "OK"
        },
        {
          "StartDate": "2022-10-26T12:05:00+00:00",
          "EndDate": "2022-10-26T12:10:00+00:00",
          "Value": 62.483498,
          "Status": "OK"
        }
      ]
    }
  ]
}
```

```
{
  "StartDate": "2022-10-26T12:10:00+00:00",
  "EndDate": "2022-10-26T12:15:00+00:00",
  "Value": 62.51248,
  "Status": "OK"
},
{
  "StartDate": "2022-10-26T12:15:00+00:00",
  "EndDate": "2022-10-26T12:20:00+00:00",
  "Value": 62.635475,
  "Status": "OK"
},
{
  "StartDate": "2022-10-26T12:20:00+00:00",
  "EndDate": "2022-10-26T12:25:00+00:00",
  "Value": 62.733974,
  "Status": "OK"
},
{
  "StartDate": "2022-10-26T12:25:00+00:00",
  "EndDate": "2022-10-26T12:30:00+00:00",
  "Value": 62.773975,
  "Status": "OK"
},
{
  "StartDate": "2022-10-26T12:30:00+00:00",
  "EndDate": "2022-10-26T12:35:00+00:00",
  "Value": 62.75349,
  "Status": "OK"
}
]
}
```

Weitere Informationen finden Sie unter [Überwachen der Netzwerkleistung](#) im Infrastructure Performance User Guide.

- API-Einheiten finden Sie [GetAwsNetworkPerformanceData](#) in der AWS CLI Befehlsreferenz.

get-capacity-reservation-usage

Das folgende Codebeispiel zeigt die Verwendung `get-capacity-reservation-usage`.

AWS CLI

Um die Nutzung der Kapazitätsreservierungen für mehrere AWS Konten anzuzeigen

Im folgenden `get-capacity-reservation-usage` Beispiel werden Nutzungsinformationen für die angegebene Kapazitätsreservierung angezeigt.

```
aws ec2 get-capacity-reservation-usage \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Ausgabe:

```
{
  "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
  "InstanceUsages": [
    {
      "UsedInstanceCount": 1,
      "AccountId": "123456789012"
    }
  ],
  "AvailableInstanceCount": 4,
  "TotalInstanceCount": 5,
  "State": "active",
  "InstanceType": "t2.medium"
}
```

Weitere Informationen finden Sie unter [Nutzung von Shared Capacity Reservation Usage](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [GetCapacityReservationUsage](#) unter AWS CLI Befehlsreferenz.

get-coip-pool-usage

Das folgende Codebeispiel zeigt die Verwendung `get-coip-pool-usage`.

AWS CLI

Um die Nutzung des kundeneigenen IP-Adresspools zu erhalten

Im folgenden `get-coip-pool-usage` Beispiel werden die Nutzungsdetails für den angegebenen kundeneigenen IP-Adresspool abgerufen.

```
aws ec2 get-coip-pool-usage \  
  --pool-id ipv4pool-coip-123a45678bEXAMPLE
```

Ausgabe:

```
{  
  "CoipPoolId": "ipv4pool-coip-123a45678bEXAMPLE",  
  "CoipAddressUsages": [  
    {  
      "CoIp": "0.0.0.0"  
    },  
    {  
      "AllocationId": "eipalloc-123ab45c6dEXAMPLE",  
      "AwsAccountId": "123456789012",  
      "CoIp": "0.0.0.0"  
    },  
    {  
      "AllocationId": "eipalloc-123ab45c6dEXAMPLE",  
      "AwsAccountId": "123456789111",  
      "CoIp": "0.0.0.0"  
    }  
  ],  
  "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Kundeneigene IP-Adressen](#) im AWS -Outposts-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetCoipPoolUsage AWS CLI Befehlsreferenz](#).

get-console-output

Das folgende Codebeispiel zeigt die Verwendung `get-console-output`.

AWS CLI

Beispiel 1: Um die Konsolenausgabe zu erhalten

Im folgenden `get-console-output` Beispiel wird die Konsolenausgabe für die angegebene Linux-Instanz abgerufen.

```
aws ec2 get-console-output \  
  --instance-id i-12345678901234567890
```

```
--instance-id i-1234567890abcdef0
```

Ausgabe:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-07-25T21:23:53.000Z",
  "Output": "..."}
}
```

Weitere Informationen finden Sie unter [Ausgabe der Instance-Konsole](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Um die neueste Konsolenausgabe zu erhalten

Im folgenden `get-console-output` Beispiel wird die neueste Konsolenausgabe für die angegebene Linux-Instance abgerufen.

```
aws ec2 get-console-output \
  --instance-id i-1234567890abcdef0 \
  --latest \
  --output text
```

Ausgabe:

```
i-1234567890abcdef0 [ 0.000000] Command line: root=LABEL=/ console=tty1
console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point
registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
...
Cloud-init v. 0.7.6 finished at Wed, 09 May 2018 19:01:13 +0000. Datasource
DataSourceEc2. Up 21.50 seconds
Amazon Linux AMI release 2018.03
Kernel 4.14.26-46.32.amzn1.x
```

Weitere Informationen finden Sie unter [Ausgabe der Instance-Konsole](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetConsoleOutput](#) in der AWS CLI Befehlsreferenz.

get-console-screenshot

Das folgende Codebeispiel zeigt die Verwendung `get-console-screenshot`.

AWS CLI

Um einen Screenshot einer laufenden Instanz abzurufen

Im folgenden `get-console-screenshot` Beispiel wird ein Screenshot der angegebenen Instanz im JPG-Format abgerufen. Der Screenshot wird als Base64-kodierte Zeichenfolge zurückgegeben.

```
aws ec2 get-console-screenshot \  
  --instance-id i-1234567890abcdef0
```

Ausgabe:

```
{  
  "ImageData": "997987/8kgj49ikjhewkww0008084EXAMPLE",  
  "InstanceId": "i-1234567890abcdef0"  
}
```

- API-Einheiten finden Sie [GetConsoleScreenshot](#) in AWS CLI der Befehlsreferenz.

get-default-credit-specification

Das folgende Codebeispiel zeigt die Verwendung `get-default-credit-specification`.

AWS CLI

Um die Standard-Kreditoption zu beschreiben

Im folgenden `get-default-credit-specification` Beispiel wird die Standard-Kreditoption für T2-Instances beschrieben.

```
aws ec2 get-default-credit-specification \  
  --instance-family t2
```

Ausgabe:

```
{
```

```
"InstanceFamilyCreditSpecification": {
  "InstanceFamily": "t2",
  "CpuCredits": "standard"
}
```

- API-Einheiten finden Sie [GetDefaultCreditSpecification](#) unter AWS CLI Befehlsreferenz.

get-ebs-default-kms-key-id

Das folgende Codebeispiel zeigt die Verwendung `get-ebs-default-kms-key-id`.

AWS CLI

Um Ihre EBS Standardverschlüsselung CMK zu beschreiben

Das folgende `get-ebs-default-kms-key-id` Beispiel beschreibt die EBS Standardverschlüsselung CMK für Ihr AWS Konto.

```
aws ec2 get-ebs-default-kms-key-id
```

Die Ausgabe zeigt die Standardeinstellung CMK für EBS Verschlüsselung, bei der es sich um eine CMK mit dem Alias `aws/alias/aws/ebs` handelt.

```
{
  "KmsKeyId": "alias/aws/ebs"
}
```

Die folgende Ausgabe zeigt eine benutzerdefinierte CMK EBS Verschlüsselung.

```
{
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- API-Einheiten finden Sie [GetEbsDefaultKmsKeyId](#) in der AWS CLI Befehlsreferenz.

get-ebs-encryption-by-default

Das folgende Codebeispiel zeigt die Verwendung `get-ebs-encryption-by-default`.

AWS CLI

Um zu beschreiben, ob die EBS Verschlüsselung standardmäßig aktiviert ist

Das folgende `get-ebs-encryption-by-default` Beispiel zeigt, ob die EBS Verschlüsselung für Ihr AWS Konto in der aktuellen Region standardmäßig aktiviert ist.

```
aws ec2 get-ebs-encryption-by-default
```

Die folgende Ausgabe zeigt, dass die EBS Verschlüsselung standardmäßig deaktiviert ist.

```
{
  "EbsEncryptionByDefault": false
}
```

Die folgende Ausgabe gibt an, dass die EBS Verschlüsselung standardmäßig aktiviert ist.

```
{
  "EbsEncryptionByDefault": true
}
```

- API-Einheiten finden Sie [GetEbsEncryptionByDefault](#) in der AWS CLI Befehlsreferenz.

get-flow-logs-integration-template

Das folgende Codebeispiel zeigt die Verwendung `get-flow-logs-integration-template`.

AWS CLI

Um eine CloudFormation Vorlage zur Automatisierung der Integration von VPC Flow-Protokollen mit Amazon Athena zu erstellen

In den folgenden `get-flow-logs-integration-template` Beispielen wird eine CloudFormation Vorlage erstellt, um die Integration von VPC Flow-Protokollen mit Amazon Athena zu automatisieren.

Linux:

```
aws ec2 get-flow-logs-integration-template \
  --flow-log-id fl-1234567890abcdef0 \
  --config-delivery-s3-destination-arn arn:aws:s3:::DOC-EXAMPLE-BUCKET \
```

--integrate-services

```
AthenaIntegrations='[{IntegrationResultS3DestinationArn=arn:aws:s3:::DOC-EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
}{IntegrationResultS3DestinationArn=arn:aws:s3:::DOC-EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

Windows:

```
aws ec2 get-flow-logs-integration-template ^
--flow-log-id fl-1234567890abcdef0 ^
--config-delivery-s3-destination-arn arn:aws:s3:::DOC-EXAMPLE-BUCKET ^
--integrate-
services AthenaIntegrations=[{IntegrationResultS3DestinationArn=arn:aws:s3:::DOC-
EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
}{IntegrationResultS3DestinationArn=arn:aws:s3:::DOC-EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

Ausgabe:

```
{
  "Result": "https://DOC-EXAMPLE-BUCKET.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_fl-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
}
```

Informationen zur Verwendung von CloudFormation Vorlagen finden Sie unter [Arbeiten mit AWS CloudFormation Vorlagen](#) im AWS CloudFormation Benutzerhandbuch.

Informationen zur Verwendung von Amazon Athena und Flow-Protokollen finden Sie unter [Abfragen von Flow-Protokollen mit Amazon Athena](#) im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [GetFlowLogsIntegrationTemplate](#) in der AWS CLI Befehlsreferenz.

get-groups-for-capacity-reservation

Das folgende Codebeispiel zeigt die Verwendung `get-groups-for-capacity-reservation`.

AWS CLI

Um die Ressourcengruppen mit einer Kapazitätsreservierung aufzulisten

Das folgende `get-groups-for-capacity-reservation` Beispiel listet die Ressourcengruppen auf, zu denen die angegebene Kapazitätsreservierung hinzugefügt wurde.

```
aws ec2 get-groups-for-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Ausgabe:

```
{
  "CapacityReservationsGroup": [
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
      "OwnerId": "123456789012"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Kapazitätsreservierungen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [GetGroupsForCapacityReservation](#) unter AWS CLI Befehlsreferenz.

get-host-reservation-purchase-preview

Das folgende Codebeispiel zeigt die Verwendung von `get-host-reservation-purchase-preview`.

AWS CLI

Um eine Kaufvorschau für eine Reservierung mit einem Dedicated Host zu erhalten

Dieses Beispiel bietet eine Vorschau der Kosten für eine bestimmte Dedicated Host-Reservierung für den angegebenen Dedicated Host in Ihrem Konto.

Befehl:

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --
host-id-set h-013abcd2a00cbd123
```

Ausgabe:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
    {
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "HostIdSet": [
        "h-013abcd2a00cbd123"
      ],
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- API-Einheiten finden Sie [GetHostReservationPurchasePreview](#) in der AWS CLI Befehlsreferenz.

get-image-block-public-access-state

Das folgende Codebeispiel zeigt die Verwendung `get-image-block-public-access-state`.

AWS CLI

Um den Status des Blockierens des öffentlichen Zugriffs AMIs in der angegebenen Region abzurufen

Im folgenden `get-image-block-public-access-state` Beispiel wird der Status „Öffentlicher Zugriff gesperrt“ AMIs auf Kontoebene in der angegebenen Region abgerufen.

```
aws ec2 get-image-block-public-access-state \
  --region us-east-1
```

Ausgabe:

```
{
  "ImageBlockPublicAccessState": "block-new-sharing"
}
```

Weitere Informationen finden Sie unter [Sperren des öffentlichen Zugriffs auf Ihre AMIs](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetImageBlockPublicAccessState](#) in der AWS CLI Befehlsreferenz.

get-instance-types-from-instance-requirements

Das folgende Codebeispiel zeigt die Verwendung `get-instance-types-from-instance-requirements`.

AWS CLI

Um eine Vorschau der Instanztypen anzuzeigen, die den angegebenen Attributen entsprechen

Im folgenden `get-instance-types-from-instance-requirements` Beispiel wird zunächst eine Liste aller möglichen Attribute generiert, die mithilfe des `--generate-cli-skeleton` Parameters angegeben werden können, und die Liste wird in einer JSON Datei gespeichert. Anschließend wird die JSON Datei verwendet, um die Attribute anzupassen, für die eine Vorschau der übereinstimmenden Instanztypen angezeigt werden soll.

Verwenden Sie den folgenden Befehl, um alle möglichen Attribute zu generieren und die Ausgabe direkt in einer JSON Datei zu speichern.

```
aws ec2 get-instance-types-from-instance-requirements \
  --region us-east-1 \
  --generate-cli-skeleton input > attributes.json
```

Ausgabe:

```
{
  "DryRun": true,
  "ArchitectureTypes": [
    "x86_64_mac"
  ],
  "VirtualizationTypes": [
    "paravirtual"
  ],
  "InstanceRequirements": {
    "VCpuCount": {
      "Min": 0,
      "Max": 0
    }
  },
}
```

```
"MemoryMiB": {
  "Min": 0,
  "Max": 0
},
"CpuManufacturers": [
  "intel"
],
"MemoryGiBPerVCpu": {
  "Min": 0.0,
  "Max": 0.0
},
"ExcludedInstanceTypes": [
  ""
],
"InstanceGenerations": [
  "current"
],
"SpotMaxPricePercentageOverLowestPrice": 0,
"OnDemandMaxPricePercentageOverLowestPrice": 0,
"BareMetal": "included",
"BurstablePerformance": "excluded",
"RequireHibernateSupport": true,
"NetworkInterfaceCount": {
  "Min": 0,
  "Max": 0
},
"LocalStorage": "required",
"LocalStorageTypes": [
  "hdd"
],
"TotalLocalStorageGB": {
  "Min": 0.0,
  "Max": 0.0
},
"BaselineEbsBandwidthMbps": {
  "Min": 0,
  "Max": 0
},
"AcceleratorTypes": [
  "inference"
],
"AcceleratorCount": {
  "Min": 0,
  "Max": 0
}
```



```

    },
    "AcceleratorManufacturers": [
        "xilinx"
    ],
    "AcceleratorNames": [
        "t4"
    ],
    "AcceleratorTotalMemoryMiB": {
        "Min": 0,
        "Max": 0
    }
},
"MaxResults": 0,
"NextToken": ""
}

```

Konfigurieren Sie die JSON-Datei. Sie müssen Werte für `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` und `MemoryMiB` angeben. Sie können die anderen Attribute weglassen. Wenn sie weggelassen werden, werden Standardwerte verwendet. Eine Beschreibung der einzelnen Attribute und ihrer Standardwerte finden Sie unter `get-instance-types-from-instance-requirements <- -requirements.html>` <https://docs.aws.amazon.com/cli/latest/reference/ec2/get-instance-types-from-instance-requirements>

Sehen Sie sich eine Vorschau der Instanztypen an, deren Attribute angegeben sind. `attributes.json` Geben Sie mithilfe des `--cli-input-json` Parameters den Namen und den Pfad zu Ihrer JSON Datei an. In der folgenden Anforderung wird die Ausgabe als Tabelle formatiert.

```

aws ec2 get-instance-types-from-instance-requirements \
  --cli-input-json file://attributes.json \
  --output table

```

Inhalt der `attributes.json` Datei:

```

{
  "ArchitectureTypes": [
    "x86_64"
  ],
  "VirtualizationTypes": [
    "hvm"
  ]
}

```

```

    ],
    "InstanceRequirements": {
      "VCpuCount": {
        "Min": 4,
        "Max": 6
      },
      "MemoryMiB": {
        "Min": 2048
      },
      "InstanceGenerations": [
        "current"
      ]
    }
  }
}

```

Ausgabe:

```

-----
|GetInstanceTypesFromInstanceRequirements|
+-----+
||           InstanceTypes           ||
|+-----+|
||           InstanceType           ||
|+-----+|
||  c4.xlarge                        ||
||  c5.xlarge                        ||
||  c5a.xlarge                       ||
||  c5ad.xlarge                      ||
||  c5d.xlarge                       ||
||  c5n.xlarge                       ||
||  d2.xlarge                        ||
...

```

Weitere Informationen zur attributbasierten Instance-Typ-Auswahl finden Sie unter [So funktioniert die attributbasierte Instance-Typ-Auswahl](#) im Amazon-Benutzerhandbuch. EC2

- API-Einheiten finden Sie in der Befehlsreferenz.
[GetInstanceTypesFromInstanceRequirements](#) AWS CLI

get-instance-uefi-data

Das folgende Codebeispiel zeigt die Verwendung `get-instance-uefi-data`.

AWS CLI

Um UEFI Daten von einer Instanz abzurufen

Im folgenden `get-instance-uefi-data` Beispiel werden UEFI Daten aus einer Instanz abgerufen. Wenn die Ausgabe leer ist, enthält die Instanz keine UEFI Daten.

```
aws ec2 get-instance-uefi-data \
  --instance-id i-0123456789example
```

Ausgabe:

```
{
  "InstanceId": "i-0123456789example",
  "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.
  <snipped>
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/A0Dshho="
}
```

Weitere Informationen finden Sie unter [UEFI Secure Boot](#) im EC2 Amazon-Benutzerhandbuch.

- API Einzelheiten finden Sie [GetInstanceUefiData](#) in der AWS CLI Befehlsreferenz.

get-ipam-address-history

Das folgende Codebeispiel zeigt die Verwendung `get-ipam-address-history`.

AWS CLI

Um die Geschichte eines zu ermitteln CIDR

Das folgende `get-ipam-address-history` Beispiel ruft den Verlauf von a ab CIDR.

(Linux):

```
aws ec2 get-ipam-address-history \
  --cidr 10.0.0.0/16 \
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
  --start-time 2021-12-08T01:00:00.000Z \
  --end-time 2021-12-10T01:00:00.000Z
```

(Windows):

```
aws ec2 get-ipam-address-history ^  
--cidr 10.0.0.0/16 ^  
--ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
--start-time 2021-12-08T01:00:00.000Z ^  
--end-time 2021-12-10T01:00:00.000Z
```

Ausgabe:

```
{  
  "HistoryRecords": [  
    {  
      "ResourceOwnerId": "123456789012",  
      "ResourceRegion": "us-west-1",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-06cbefa9ee907e1c0",  
      "ResourceCidr": "10.0.0.0/16",  
      "ResourceName": "Demo",  
      "ResourceComplianceStatus": "unmanaged",  
      "ResourceOverlapStatus": "overlapping",  
      "VpcId": "vpc-06cbefa9ee907e1c0",  
      "SampledStartTime": "2021-12-08T19:54:57.675000+00:00"  
    },  
    {  
      "ResourceOwnerId": "123456789012",  
      "ResourceRegion": "us-east-2",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-042702f474812c9ad",  
      "ResourceCidr": "10.0.0.0/16",  
      "ResourceName": "test",  
      "ResourceComplianceStatus": "unmanaged",  
      "ResourceOverlapStatus": "overlapping",  
      "VpcId": "vpc-042702f474812c9ad",  
      "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"  
    },  
    {  
      "ResourceOwnerId": "123456789012",  
      "ResourceRegion": "us-east-2",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-042b8a44f64267d67",  
      "ResourceCidr": "10.0.0.0/16",  
      "ResourceName": "tester",  
      "ResourceComplianceStatus": "unmanaged",  
      "ResourceOverlapStatus": "overlapping",
```

```

        "VpcId": "vpc-042b8a44f64267d67",
        "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Anzeigen des Verlaufs von IP-Adressen](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetIpamAddressHistory](#) in der AWS CLI Befehlsreferenz.

get-ipam-discovered-accounts

Das folgende Codebeispiel zeigt die Verwendung `get-ipam-discovered-accounts`.

AWS CLI

Um die Konten anzuzeigen, die von einem entdeckten IPAM

In diesem Szenario sind Sie ein IPAM delegierter Administrator, der die AWS Konten einsehen möchte, denen die Ressourcen gehören, die IPAM er entdeckt.

Das `--discovery-region` ist die IPAM Betriebsregion, in der Sie den Status der überwachten Konten einsehen möchten. Wenn Sie beispielsweise über drei IPAM Betriebsregionen verfügen, möchten Sie diese Anfrage möglicherweise dreimal stellen, um die für die Erkennung spezifischen Zeitstempel in jeder dieser Regionen einzusehen.

Das folgende `get-ipam-discovered-accounts` Beispiel listet die AWS Konten auf, denen die Ressourcen gehören, die IPAM der entdeckt.

```

aws ec2 get-ipam-discovered-accounts \
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
  --discovery-region us-east-1

```

Ausgabe:

```

{
  "IpamDiscoveredAccounts": [
    {
      "AccountId": "149977607591",
      "DiscoveryRegion": "us-east-1",

```

```

        "LastAttemptedDiscoveryTime": "2024-02-09T19:04:31.379000+00:00",
        "LastSuccessfulDiscoveryTime": "2024-02-09T19:04:31.379000+00:00"
    }
]
}

```

Weitere Informationen finden Sie im VPCIPAMAmazon-Benutzerhandbuch unter [Integration IPAM mit Konten außerhalb Ihrer Organisation](#).

- APIEinzelheiten finden Sie [GetIpamDiscoveredAccounts](#) in der AWS CLI Befehlsreferenz.

get-ipam-discovered-public-addresses

Das folgende Codebeispiel zeigt die Verwendung `get-ipam-discovered-public-addresses`.

AWS CLI

Um entdeckte öffentliche IP-Adressen anzuzeigen

In diesem Beispiel sind Sie ein IPAM delegierter Administrator und möchten die IP-Adressen der Ressourcen einsehen, die von IPAM entdeckt wurden. Sie können die ID für die Ressourcenerkennung mit [describe-ipam-resource-discoveries](#) abrufen.

Das folgende `get-ipam-discovered-public-addresses` Beispiel zeigt die erkannten öffentlichen IP-Adressen für eine Ressourcenerkennung.

```

aws ec2 get-ipam-discovered-public-addresses \
  --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
  --address-region us-east-1 \
  --region us-east-1

```

Ausgabe:

```

{
  "IpamDiscoveredPublicAddresses": [
    {
      "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
      "AddressRegion": "us-east-1",
      "Address": "54.208.155.7",
      "AddressOwnerId": "320805250157",
      "AssociationStatus": "associated",
      "AddressType": "ec2-public-ip",
    }
  ]
}

```

```

    "VpcId": "vpc-073b294916198ce49",
    "SubnetId": "subnet-0b6c8a8839e9a4f15",
    "NetworkInterfaceId": "eni-081c446b5284a5e06",
    "NetworkInterfaceDescription": "",
    "InstanceId": "i-07459a6fca5b35823",
    "Tags": {},
    "NetworkBorderGroup": "us-east-1c",
    "SecurityGroups": [
      {
        "GroupName": "launch-wizard-2",
        "GroupId": "sg-0a489dd6a65c244ce"
      }
    ],
    "SampleTime": "2024-04-05T15:13:59.228000+00:00"
  },
  {
    "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
    "AddressRegion": "us-east-1",
    "Address": "44.201.251.218",
    "AddressOwnerId": "470889052923",
    "AssociationStatus": "associated",
    "AddressType": "ec2-public-ip",
    "VpcId": "vpc-6c31a611",
    "SubnetId": "subnet-062f47608b99834b1",
    "NetworkInterfaceId": "eni-024845359c2c3ae9b",
    "NetworkInterfaceDescription": "",
    "InstanceId": "i-04ef786d9c4e03f41",
    "Tags": {},
    "NetworkBorderGroup": "us-east-1a",
    "SecurityGroups": [
      {
        "GroupName": "launch-wizard-32",
        "GroupId": "sg-0ed1a426e96a68374"
      }
    ],
    "SampleTime": "2024-04-05T15:13:59.145000+00:00"
  }
}

```

Weitere Informationen finden Sie unter [Public IP Insights anzeigen](#) im VPCIPAM Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetIpamDiscoveredPublicAddresses](#) in der AWS CLI Befehlsreferenz.

get-ipam-discovered-resource-cidrs

Das folgende Codebeispiel zeigt die Verwendung `get-ipam-discovered-resource-cidrs`.

AWS CLI

Um die IP-Adresse anzuzeigen, die von einem CIDRs entdeckt wurde IPAM

In diesem Beispiel sind Sie ein IPAM delegierter Administrator, der Details CIDRs zur IP-Adresse der Ressourcen einsehen möchte, die IPAM er entdeckt.

Um diese Anfrage abzuschließen:

Die von Ihnen gewählte Ressourcensuche muss mit der IPAM AWS Region verknüpft sein, in der die Ressource erstellt wurde. `--resource-region`

Das folgende `get-ipam-discovered-resource-cidrs` Beispiel listet die IP-Adressen der Ressourcen auf, die der IPAM ermittelt.

```
aws ec2 get-ipam-discovered-resource-cidrs \  
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \  
  --resource-region us-east-1
```

Ausgabe:

```
{  
  {  
    "IpamDiscoveredResourceCidrs": [  
      {  
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",  
        "ResourceRegion": "us-east-1",  
        "ResourceId": "vpc-0c974c95ca7ceef4a",  
        "ResourceOwnerId": "149977607591",  
        "ResourceCidr": "172.31.0.0/16",  
        "ResourceType": "vpc",  
        "ResourceTags": [],  
        "IpUsage": 0.375,  
        "VpcId": "vpc-0c974c95ca7ceef4a",  
        "SampleTime": "2024-02-09T19:15:16.529000+00:00"  
      },  
      {  
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",  
        "ResourceRegion": "us-east-1",
```



```

    "ResourceId": "subnet-07fe028119082a8c1",
    "ResourceOwnerId": "149977607591",
    "ResourceCidr": "172.31.0.0/20",
    "ResourceType": "subnet",
    "ResourceTags": [],
    "IpUsage": 0.0012,
    "VpcId": "vpc-0c974c95ca7ceef4a",
    "SampleTime": "2024-02-09T19:15:16.529000+00:00"
  },
  {
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
    "ResourceRegion": "us-east-1",
    "ResourceId": "subnet-0a96893763984cc4e",
    "ResourceOwnerId": "149977607591",
    "ResourceCidr": "172.31.64.0/20",
    "ResourceType": "subnet",
    "ResourceTags": [],
    "IpUsage": 0.0012,
    "VpcId": "vpc-0c974c95ca7ceef4a",
    "SampleTime": "2024-02-09T19:15:16.529000+00:00"
  }
}

```

Weitere Informationen finden Sie unter [Überwachen der CIDR Nutzung nach Ressourcen](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [GetIpamDiscoveredResourceCidrs](#) in der AWS CLI Befehlsreferenz.

get-ipam-pool-allocations

Das folgende Codebeispiel zeigt die Verwendung `get-ipam-pool-allocations`.

AWS CLI

Um die CIDRs zugewiesenen Daten aus einem IPAM Pool abzurufen

Im folgenden `get-ipam-pool-allocations` Beispiel wird der CIDRs zugewiesene Wert aus einem IPAM Pool abgerufen.

(Linux):

```
aws ec2 get-ipam-pool-allocations \
```

```
--ipam-pool-id ipam-pool-0533048da7d823723 \  
--filters Name=ipam-pool-allocation-id,Values=ipam-pool-alloc-0e6186d73999e47389266a5d6991e6220
```

(Windows):

```
aws ec2 get-ipam-pool-allocations ^  
--ipam-pool-id ipam-pool-0533048da7d823723 ^  
--filters Name=ipam-pool-allocation-id,Values=ipam-pool-alloc-0e6186d73999e47389266a5d6991e6220
```

Ausgabe:

```
{  
  "IpamPoolAllocations": [  
    {  
      "Cidr": "10.0.0.0/16",  
      "IpamPoolAllocationId": "ipam-pool-alloc-0e6186d73999e47389266a5d6991e6220",  
      "ResourceType": "custom",  
      "ResourceOwner": "123456789012"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetIpamPoolAllocations](#) in der AWS CLI Befehlsreferenz.

get-ipam-pool-cidrs

Das folgende Codebeispiel zeigt die Verwendung von `get-ipam-pool-cidrs`.

AWS CLI

Um die CIDRs, die in einem IPAM-Pool provisioniert sind, zu erhalten

Im folgenden `get-ipam-pool-cidrs`-Beispiel wird das CIDR-Provisioning für einen IPAM-Pool abgerufen.

(Linux):

```
aws ec2 get-ipam-pool-cidrs \  

```

```
--ipam-pool-id ipam-pool-0533048da7d823723 \  
--filters 'Name=cidr,Values=10.*'
```

(Windows):

```
aws ec2 get-ipam-pool-cidrs ^  
--ipam-pool-id ipam-pool-0533048da7d823723 ^  
--filters Name=cidr,Values=10.*
```

Ausgabe:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "10.0.0.0/24",  
    "State": "provisioned"  
  }  
}
```

- API-Einheiten finden Sie [GetIpamPoolCidrs](#) in der AWS CLI Befehlsreferenz.

get-ipam-resource-cidrs

Das folgende Codebeispiel zeigt die Verwendung `get-ipam-resource-cidrs`.

AWS CLI

Um das einer Ressource CIDRs zugewiesene zu erhalten

Im folgenden `get-ipam-resource-cidrs` Beispiel wird der einer Ressource CIDRs zugewiesene Wert abgerufen.

(Linux):

```
aws ec2 get-ipam-resource-cidrs \  
--ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \  
--filters Name=management-state,Values=unmanaged
```

(Windows):

```
aws ec2 get-ipam-resource-cidrs ^
```

```
--ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
--filters Name=management-state,Values=unmanaged
```

Ausgabe:

```
{  
  "IpamResourceCidrs": [  
    {  
      "IpamId": "ipam-08440e7a3acde3908",  
      "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",  
      "ResourceRegion": "us-east-2",  
      "ResourceOwnerId": "123456789012",  
      "ResourceId": "vpc-621b8709",  
      "ResourceName": "Default AWS VPC",  
      "ResourceCidr": "172.33.0.0/16",  
      "ResourceType": "vpc",  
      "ResourceTags": [  
        {  
          "Key": "Environment",  
          "Value": "Test"  
        },  
        {  
          "Key": "Name",  
          "Value": "Default AWS VPC"  
        }  
      ],  
      "IpUsage": 0.0039,  
      "ComplianceStatus": "unmanaged",  
      "ManagementState": "unmanaged",  
      "OverlapStatus": "nonoverlapping",  
      "VpcId": "vpc-621b8709"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Überwachen der CIDR Nutzung nach Ressourcen](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [GetIpamResourceCidrs](#) in der AWS CLI Befehlsreferenz.

get-launch-template-data

Das folgende Codebeispiel zeigt die Verwendung `get-launch-template-data`.

AWS CLI

Um Instanzdaten für eine Startvorlage abzurufen

In diesem Beispiel werden Daten über die angegebene Instanz abgerufen und die `--query` Option verwendet, um den Inhalt zurückzugeben `LaunchTemplateData`. Sie können die Ausgabe als Basis zum Erstellen einer neuen Startvorlage oder einer neuen Version der Startvorlage verwenden.

Befehl:

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query  
'LaunchTemplateData'
```

Ausgabe:

```
{  
  "Monitoring": {},  
  "ImageId": "ami-8c1be5f6",  
  "BlockDeviceMappings": [  
    {  
      "DeviceName": "/dev/xvda",  
      "Ebs": {  
        "DeleteOnTermination": true  
      }  
    }  
  ],  
  "EbsOptimized": false,  
  "Placement": {  
    "Tenancy": "default",  
    "GroupName": "",  
    "AvailabilityZone": "us-east-1a"  
  },  
  "InstanceType": "t2.micro",  
  "NetworkInterfaces": [  
    {  
      "Description": "",  
      "NetworkInterfaceId": "eni-35306abc",  
      "PrivateIpAddresses": [  
        {  
          "Primary": true,  
          "PrivateIpAddress": "10.0.0.72"  
        }  
      ]  
    }  
  ]  
}
```

```

    ],
    "SubnetId": "subnet-7b16de0c",
    "Groups": [
      "sg-7c227019"
    ],
    ],
    "Ipv6Addresses": [
      {
        "Ipv6Address": "2001:db8:1234:1a00::123"
      }
    ],
    ],
    "PrivateIpAddress": "10.0.0.72"
  }
]
}

```

- API-Einheiten finden Sie [GetLaunchTemplateData](#) unter AWS CLI Befehlsreferenz.

get-managed-prefix-list-associations

Das folgende Codebeispiel zeigt die Verwendung `get-managed-prefix-list-associations`.

AWS CLI

Um Präfixlistenzuordnungen abzurufen

Im folgenden `get-managed-prefix-list-associations` Beispiel werden die Ressourcen abgerufen, die der angegebenen Präfixliste zugeordnet sind.

```

aws ec2 get-managed-prefix-list-associations \
  --prefix-list-id pl-0123456abcabc1

```

Ausgabe:

```

{
  "PrefixListAssociations": [
    {
      "ResourceId": "sg-0abc123456abc12345",
      "ResourceOwner": "123456789012"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetManagedPrefixListAssociations](#) in der AWS CLI Befehlsreferenz.

get-managed-prefix-list-entries

Das folgende Codebeispiel zeigt die Verwendung von `get-managed-prefix-list-entries`.

AWS CLI

Um die Einträge für eine Präfixliste abzurufen

Im Folgenden werden `get-managed-prefix-list-entries` die Einträge für die angegebene Präfixliste abgerufen.

```
aws ec2 get-managed-prefix-list-entries \  
  --prefix-list-id p1-0123456abcabc1
```

Ausgabe:

```
{  
  "Entries": [  
    {  
      "Cidr": "10.0.0.0/16",  
      "Description": "vpc-a"  
    },  
    {  
      "Cidr": "10.2.0.0/16",  
      "Description": "vpc-b"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetManagedPrefixListEntries](#) in der AWS CLI Befehlsreferenz.

get-network-insights-access-scope-analysis-findings

Das folgende Codebeispiel zeigt die Verwendung `get-network-insights-access-scope-analysis-findings`.

AWS CLI

Um die Ergebnisse von Network Insights zu erhalten, greifen Sie auf die Umfangsanalyse zu

Im folgenden `get-network-insights-access-scope-analysis-findings` Beispiel werden die ausgewählten Ergebnisse der Umfangsanalyse in Ihrem AWS Konto abgerufen.

```
aws ec2 get-network-insights-access-scope-analysis-findings \
  --region us-east-1 \
  --network-insights-access-scope-analysis-id nisa \
  --nis-123456789111
```

Ausgabe:

```
{
  "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
  "AnalysisFindings": [
    {
      "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
      "NetworkInsightsAccessScopeId": "nis-123456789111",
      "FindingComponents": [
        {
          "SequenceNumber": 1,
          "Component": {
            "Id": "eni-02e3d42d5cceca67d",
            "Arn": "arn:aws:ec2:us-east-1:936459623503:network-
interface/eni-02e3d32d9cceca17d"
          },
          "OutboundHeader": {
            "DestinationAddresses": [
              "0.0.0.0/5",
              "11.0.0.0/8",
              "12.0.0.0/6",
              "128.0.0.0/3",
              "16.0.0.0/4",
              "160.0.0.0/5",
              "168.0.0.0/6",
```



```
aws ec2 get-network-insights-access-scope-content \  
  --region us-east-1 \  
  --network-insights-access-scope-id nis-123456789222
```

Ausgabe:

```
{  
  "NetworkInsightsAccessScopeContent": {  
    "NetworkInsightsAccessScopeId": "nis-123456789222",  
    "MatchPaths": [  
      {  
        "Source": {  
          "ResourceStatement": {  
            "ResourceTypes": [  
              "AWS::EC2::NetworkInterface"  
            ]  
          }  
        },  
        "Destination": {  
          "ResourceStatement": {  
            "ResourceTypes": [  
              "AWS::EC2::InternetGateway"  
            ]  
          }  
        }  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer unter Verwendung von AWS CLI im Network Access Analyzer-Handbuch](#).

- API-Einheiten finden Sie [GetNetworkInsightsAccessScopeContent](#) unter AWS CLI Befehlsreferenz.

get-password-data

Das folgende Codebeispiel zeigt die Verwendung von `get-password-data`.

AWS CLI

Um das verschlüsselte Passwort zu erhalten

In diesem Beispiel wird das verschlüsselte Passwort abgerufen.

Befehl:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

Ausgabe:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-07T22:18:38.000Z",
  "PasswordData": "gSlJFq+VpcZXqy+iktXMF6NyxQ4qCrT4+ga0uN0enX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTYp7WmU3TUnhsuBd+p6LVk7T2lKUm160Xbk6WPW1VYYm/TRPB1
e1DQ7PY4an/DgZT4mwcprFfigzhniQgDDe01InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVe1Vz/
53TkDtxbNoU606M1gK9zUWSxqEgwbvV2j8c5rP0WCuaMWSF14ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yL1f3etP2L4ZR6CvG7K1hx7VK0QVN32Dajw=="
}
```

Um das entschlüsselte Passwort zu erhalten

In diesem Beispiel wird das entschlüsselte Passwort abgerufen.

Befehl:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0 --priv-launch-key C:
\Keys\MyKeyPair.pem
```

Ausgabe:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-30T23:18:05.000Z",
  "PasswordData": "&ViJ652e*u"
}
```

- API-Einheiten finden Sie [GetPasswordData](#) in der AWS CLI Befehlsreferenz.

get-reserved-instances-exchange-quote

Das folgende Codebeispiel zeigt die Verwendung `get-reserved-instances-exchange-quote`.

AWS CLI

Um ein Angebot für den Austausch einer Convertible Reserved Instance zu erhalten

In diesem Beispiel werden die Austauschinformationen für die angegebenen Convertible Reserved Instances abgerufen.

Befehl:

```
aws ec2 get-reserved-instances-exchange-quote --reserved-  
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-  
configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

Ausgabe:

```
{  
  "CurrencyCode": "USD",  
  "ReservedInstanceValueSet": [  
    {  
      "ReservedInstanceId": "7b8750c3-397e-4da4-bbcb-a45ebexample",  
      "ReservationValue": {  
        "RemainingUpfrontValue": "0.000000",  
        "HourlyPrice": "0.027800",  
        "RemainingTotalValue": "730.556200"  
      }  
    }  
  ],  
  "PaymentDue": "424.983828",  
  "TargetConfigurationValueSet": [  
    {  
      "TargetConfiguration": {  
        "InstanceCount": 5,  
        "OfferingId": "6fea5434-b379-434c-b07b-a7abexample"  
      },  
      "ReservationValue": {  
        "RemainingUpfrontValue": "424.983828",  
        "HourlyPrice": "0.016000",  
        "RemainingTotalValue": "845.447828"  
      }  
    }  
  ]  
}
```

```

],
  "IsValidExchange": true,
  "OutputReservedInstancesWillExpireAt": "2020-10-01T13:03:39Z",
  "ReservedInstanceValueRollup": {
    "RemainingUpfrontValue": "0.000000",
    "HourlyPrice": "0.027800",
    "RemainingTotalValue": "730.556200"
  },
  "TargetConfigurationValueRollup": {
    "RemainingUpfrontValue": "424.983828",
    "HourlyPrice": "0.016000",
    "RemainingTotalValue": "845.447828"
  }
}
}

```

- API-Einheiten finden Sie [GetReservedInstancesExchangeQuote](#) unter AWS CLI Befehlsreferenz.

get-security-groups-for-vpc

Das folgende Codebeispiel zeigt die Verwendung `get-security-groups-for-vpc`.

AWS CLI

Um Sicherheitsgruppen anzuzeigen, die Netzwerkschnittstellen in einer bestimmten Umgebung zugeordnet werden können VPC.

Das folgende `get-security-groups-for-vpc` Beispiel zeigt die Sicherheitsgruppen, die Netzwerkschnittstellen in der zugeordnet werden können VPC.

```

aws ec2 get-security-groups-for-vpc \
  --vpc-id vpc-6c31a611 \
  --region us-east-1

```

Ausgabe:

```

{
  "SecurityGroupForVpcs": [
    {
      "Description": "launch-wizard-36 created 2022-08-29T15:59:35.338Z",
      "GroupName": "launch-wizard-36",
      "OwnerId": "470889052923",

```

```
    "GroupId": "sg-007e0c3027ee885f5",
    "Tags": [],
    "PrimaryVpcId": "vpc-6c31a611"
  },
  {
    "Description": "launch-wizard-18 created 2024-01-19T20:22:27.527Z",
    "GroupName": "launch-wizard-18",
    "OwnerId": "470889052923",
    "GroupId": "sg-0147193bef51c9eef",
    "Tags": [],
    "PrimaryVpcId": "vpc-6c31a611"
  }
}
```

- API-Einheiten finden Sie [GetSecurityGroupsForVpc](#) in der AWS CLI Befehlsreferenz.

get-serial-console-access-status

Das folgende Codebeispiel zeigt die Verwendung `get-serial-console-access-status`.

AWS CLI

Um den Status des Kontozugriffs auf die serielle Konsole anzuzeigen

Im folgenden `get-serial-console-access-status` Beispiel wird ermittelt, ob der serielle Konsolenzugriff für Ihr Konto aktiviert ist.

```
aws ec2 get-serial-console-access-status
```

Ausgabe:

```
{
  "SerialConsoleAccessEnabled": true
}
```

Weitere Informationen finden Sie unter [EC2Serial Console](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSerialConsoleAccessStatus](#) in der AWS CLI Befehlsreferenz.

get-snapshot-block-public-access-state

Das folgende Codebeispiel zeigt die Verwendung `get-snapshot-block-public-access-state`.

AWS CLI

Um den aktuellen Status von Block Public Access for Snapshots abzurufen

Im folgenden `get-snapshot-block-public-access-state` Beispiel wird der aktuelle Status von Block Public Access für Snapshots abgerufen.

```
aws ec2 get-snapshot-block-public-access-state
```

Ausgabe:

```
{
  "State": "block-all-sharing"
}
```

Weitere Informationen finden Sie unter [Sperren des öffentlichen Zugriffs für Snapshots](#) im EBSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSnapshotBlockPublicAccessState](#) in der AWS CLI Befehlsreferenz.

get-spot-placement-scores

Das folgende Codebeispiel zeigt die Verwendung `get-spot-placement-scores`.

AWS CLI

Um den Spot-Platzierungswert für bestimmte Anforderungen zu berechnen

Im folgenden `get-spot-placement-scores` Beispiel wird zunächst eine Liste aller möglichen Parameter generiert, die mithilfe des `--generate-cli-skeleton` Parameters für die Konfiguration des Spot-Platzierungswerts angegeben werden können, und die Liste wird in einer JSON-Datei gespeichert. Anschließend wird die JSON-Datei verwendet, um die Anforderungen für die Berechnung des Spot-Platzierungswerts zu konfigurieren.

Generiert alle möglichen Parameter, die für die Konfiguration des Spot-Platzierungswerts angegeben werden können, und speichert die Ausgabe direkt in einer JSON-Datei.

```
aws ec2 get-spot-placement-scores \
  --region us-east-1 \
  --generate-cli-skeleton input > attributes.json
```

Ausgabe:

```
{
  "InstanceTypes": [
    ""
  ],
  "TargetCapacity": 0,
  "TargetCapacityUnitType": "vcpu",
  "SingleAvailabilityZone": true,
  "RegionNames": [
    ""
  ],
  "InstanceRequirementsWithMetadata": {
    "ArchitectureTypes": [
      "x86_64_mac"
    ],
    "VirtualizationTypes": [
      "hvm"
    ],
    "InstanceRequirements": {
      "VCpuCount": {
        "Min": 0,
        "Max": 0
      },
      "MemoryMiB": {
        "Min": 0,
        "Max": 0
      },
      "CpuManufacturers": [
        "amd"
      ],
      "MemoryGiBPerVCpu": {
        "Min": 0.0,
        "Max": 0.0
      },
      "ExcludedInstanceTypes": [
        ""
      ],
      "InstanceGenerations": [
        "previous"
      ],
      "SpotMaxPricePercentageOverLowestPrice": 0,
      "OnDemandMaxPricePercentageOverLowestPrice": 0,
      "BareMetal": "excluded",
    }
  }
}
```



```
    "BurstablePerformance": "excluded",
    "RequireHibernateSupport": true,
    "NetworkInterfaceCount": {
        "Min": 0,
        "Max": 0
    },
    "LocalStorage": "included",
    "LocalStorageTypes": [
        "hdd"
    ],
    "TotalLocalStorageGB": {
        "Min": 0.0,
        "Max": 0.0
    },
    "BaselineEbsBandwidthMbps": {
        "Min": 0,
        "Max": 0
    },
    "AcceleratorTypes": [
        "fpga"
    ],
    "AcceleratorCount": {
        "Min": 0,
        "Max": 0
    },
    "AcceleratorManufacturers": [
        "amd"
    ],
    "AcceleratorNames": [
        "vu9p"
    ],
    "AcceleratorTotalMemoryMiB": {
        "Min": 0,
        "Max": 0
    }
}
},
"DryRun": true,
"MaxResults": 0,
"NextToken": ""
}
```

Konfigurieren Sie die JSON-Datei. Sie müssen einen Wert für TargetCapacity angeben. Eine Beschreibung der einzelnen Parameter und ihrer Standardwerte finden Sie unter Calculate the Spot Placement Score (AWS CLI) <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/spotcalculate-sps-cli-placement-score.html#>.

Berechnen Sie den Spot-Platzierungswert für die in angegebenen Anforderungen.
attributes.json Geben Sie mithilfe des --cli-input-json Parameters den Namen und den Pfad zu Ihrer JSON Datei an.

```
aws ec2 get-spot-placement-scores \  
  --region us-east-1 \  
  --cli-input-json file://attributes.json
```

SingleAvailabilityZone Gibt aus, ob auf gesetzt false oder weggelassen wird (wenn es weggelassen wird, wird standardmäßig auf false) ausgegeben. Es wird eine Liste mit bewerteten Regionen zurückgegeben.

```
"Recommendation": [  
  {  
    "Region": "us-east-1",  
    "Score": 7  
  },  
  {  
    "Region": "us-west-1",  
    "Score": 5  
  },  
  ...
```

Ausgabe, wenn auf eingestellt SingleAvailabilityZone ist true. Es wird eine Liste von SingleAvailability Zonen mit Punktzahlen zurückgegeben.

```
"Recommendation": [  
  {  
    "Region": "us-east-1",  
    "AvailabilityZoneId": "use1-az1"  
    "Score": 8  
  },  
  {  
    "Region": "us-east-1",  
    "AvailabilityZoneId": "usw2-az3"  
    "Score": 6
```

```
},  
...
```

Weitere Informationen zur Berechnung eines Spot-Platzierungswerts und beispielsweise zu Konfigurationen finden [Sie unter Berechnung eines Spot-Platzierungswerts](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [GetSpotPlacementScores](#) in der AWS CLI Befehlsreferenz.

get-subnet-cidr-reservations

Das folgende Codebeispiel zeigt die Verwendung `get-subnet-cidr-reservations`.

AWS CLI

Um Informationen über eine CIDR Subnetzreservierung zu erhalten

Im folgenden `get-subnet-cidr-reservations` Beispiel werden Informationen zur angegebenen CIDR Subnetzreservierung angezeigt.

```
aws ec2 get-subnet-cidr-reservations \  
--subnet-id subnet-03c51e2e6cEXAMPLE
```

Ausgabe:

```
{  
  "SubnetIpv4CidrReservations": [  
    {  
      "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",  
      "SubnetId": "subnet-03c51e2e6cEXAMPLE",  
      "Cidr": "10.1.0.16/28",  
      "ReservationType": "prefix",  
      "OwnerId": "123456789012"  
    }  
  ],  
  "SubnetIpv6CidrReservations": []  
}
```

Weitere Informationen finden Sie unter [CIDRSubnetzreservierungen](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [GetSubnetCidrReservations](#) in der AWS CLI Befehlsreferenz.

get-transit-gateway-attachment-propagations

Das folgende Codebeispiel zeigt die Verwendung `get-transit-gateway-attachment-propagations`.

AWS CLI

Um die Routentabellen aufzulisten, an die der angegebene Ressourcenanhang Routen weitergibt

Im folgenden `get-transit-gateway-attachment-propagations` Beispiel wird die Routentabelle aufgeführt, an die der angegebene Ressourcenanhang Routen weitergibt.

```
aws ec2 get-transit-gateway-attachment-propagations \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayAttachmentPropagations": [
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
      "State": "enabled"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [GetTransitGatewayAttachmentPropagations AWS CLI-Befehlsreferenz](#).

get-transit-gateway-multicast-domain-associations

Das folgende Codebeispiel zeigt die Verwendung `get-transit-gateway-multicast-domain-associations`.

AWS CLI

Um die Informationen zu den Multicast-Domänenzuordnungen des Transit-Gateways anzuzeigen

Im folgenden `get-transit-gateway-multicast-domain-associations` Beispiel werden die Zuordnungen für die angegebene Multicast-Domäne zurückgegeben.

```
aws ec2 get-transit-gateway-multicast-domain-associations \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Ausgabe:

```
{  
  "MulticastDomainAssociations": [  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8EXAMPLE",  
      "ResourceId": "vpc-01128d2c24EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-000de86e3bEXAMPLE",  
        "State": "associated"  
      }  
    },  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
      "ResourceId": "vpc-7EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-4EXAMPLE",  
        "State": "associated"  
      }  
    },  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
      "ResourceId": "vpc-7EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-5EXAMPLE",  
        "State": "associated"  
      }  
    },  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
      "ResourceId": "vpc-7EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-aEXAMPLE",  
        "State": "associated"  
      }  
    }  
  ]  
}
```

```

        "State": "associated"
      }
    },
    {
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
      "ResourceId": "vpc-7EXAMPLE",
      "ResourceType": "vpc",
      "Subnet": {
        "SubnetId": "subnet-fEXAMPLE",
        "State": "associated"
      }
    }
  ]
}

```

Weitere Informationen finden Sie im Transit Gateways Guide unter [Managing Multicast-Domains](#).

- API-Einheiten finden Sie [GetTransitGatewayMulticastDomainAssociations](#) in der AWS CLI Befehlsreferenz.

get-transit-gateway-prefix-list-references

Das folgende Codebeispiel zeigt die Verwendung `get-transit-gateway-prefix-list-references`.

AWS CLI

Um Verweise auf Präfixlisten in einer Transit-Gateway-Routentabelle abzurufen

Im folgenden `get-transit-gateway-prefix-list-references` Beispiel werden die Verweise auf die Präfixliste für die angegebene Transit-Gateway-Routentabelle abgerufen und nach der ID einer bestimmten Präfixliste gefiltert.

```

aws ec2 get-transit-gateway-prefix-list-references \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --filters Name=prefix-list-id,Values=pl-111112222222333

```

Ausgabe:

```

{
  "TransitGatewayPrefixListReferences": [
    {

```

```

    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-1111112222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "available",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Referenzen zur Präfixliste](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [GetTransitGatewayPrefixListReferences AWS CLI-Befehlsreferenz](#).

get-transit-gateway-route-table-associations

Das folgende Codebeispiel zeigt die Verwendung `get-transit-gateway-route-table-associations`.

AWS CLI

Um Informationen über die Verknüpfungen für die angegebene Transit-Gateway-Routentabelle abzurufen

Im folgenden `get-transit-gateway-route-table-associations` Beispiel werden Informationen zu den Zuordnungen für die angegebene Transit-Gateway-Routentabelle angezeigt.

```

aws ec2 get-transit-gateway-route-table-associations \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE

```

Ausgabe:

```

{
  "Associations": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",

```

```

        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "State": "associating"
    }
]
}

```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [GetTransitGatewayRouteTableAssociations AWS CLI-Befehlsreferenz](#).

get-transit-gateway-route-table-propagations

Das folgende Codebeispiel zeigt die Verwendung `get-transit-gateway-route-table-propagations`.

AWS CLI

Um Informationen zu den Routentabellen-Propagationen für die angegebene Transit-Gateway-Routentabelle anzuzeigen

Im folgenden `get-transit-gateway-route-table-propagations` Beispiel werden die Routentabellenpropagierungen für die angegebene Routentabelle zurückgegeben.

```

aws ec2 get-transit-gateway-route-table-propagations \
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE

```

Ausgabe:

```

{
  "TransitGatewayRouteTablePropagations": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
      "ResourceId": "vpc-3EXAMPLE",
      "ResourceType": "vpc",
      "State": "enabled"
    },
    {
      "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",

```



```

        "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "ResourceType": "direct-connect-gateway",
        "State": "enabled"
    },
    {
        "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
        "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
        "ResourceType": "direct-connect-gateway",
        "State": "enabled"
    }
]
}

```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [GetTransitGatewayRouteTablePropagations AWS CLI-Befehlsreferenz](#).

get-verified-access-endpoint-policy

Das folgende Codebeispiel zeigt die Verwendung `get-verified-access-endpoint-policy`.

AWS CLI

Um die Verified Access-Richtlinie eines Endpunkts abzurufen

Im folgenden `get-verified-access-endpoint-policy` Beispiel wird die Verified Access-Richtlinie des angegebenen Endpunkts abgerufen.

```
aws ec2 get-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2
```

Ausgabe:

```

{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}

```

Weitere Informationen finden Sie unter [Richtlinien für verifizierten Zugriff](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [GetVerifiedAccessEndpointPolicy](#) in der AWS CLI Befehlsreferenz.

get-verified-access-group-policy

Das folgende Codebeispiel zeigt die Verwendung `get-verified-access-group-policy`.

AWS CLI

Um die Verified Access-Richtlinie einer Gruppe abzurufen

Im folgenden `get-verified-access-group-policy` Beispiel wird die Verified Access-Richtlinie der angegebenen Gruppe abgerufen.

```
aws ec2 get-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235
```

Ausgabe:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
```

Weitere Informationen finden Sie unter [Verified Access-Gruppen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [GetVerifiedAccessGroupPolicy](#) in der AWS CLI Befehlsreferenz.

get-vpn-connection-device-sample-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-vpn-connection-device-sample-configuration`.

AWS CLI

Um eine Beispielkonfigurationsdatei herunterzuladen

Im folgenden `get-vpn-connection-device-sample-configuration` Beispiel wird die angegebene Beispielkonfigurationsdatei heruntergeladen. Rufen Sie den `get-vpn-connection-device-types` Befehl auf, um die Gateway-Geräte mit einer Beispielkonfigurationsdatei aufzulisten.

```
aws ec2 get-vpn-connection-device-sample-configuration \  
  --vpn-connection-id vpn-123456789abc01234 \  
  --vpn-connection-device-type-id 5fb390ba
```

Ausgabe:

```
{  
  "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-  
file"  
}
```

Weitere Informationen finden [Sie im AWS Site-to-Site VPN Benutzerhandbuch unter Herunterladen der Konfigurationsdatei](#).

- API-Einheiten finden Sie [GetVpnConnectionDeviceSampleConfiguration](#) in der AWS CLI Befehlsreferenz.

get-vpn-connection-device-types

Das folgende Codebeispiel zeigt die Verwendung von `get-vpn-connection-device-types`.

AWS CLI

Um Gateway-Geräte mit einer Beispielkonfigurationsdatei aufzulisten

Das folgende `get-vpn-connection-device-types` Beispiel listet die Gateway-Geräte von Palo Alto Networks auf, die über Beispielkonfigurationsdateien verfügen.

```
aws ec2 get-vpn-connection-device-types \  
  --query "VpnConnectionDeviceTypes[?Vendor=='Palo Alto Networks']"
```

Ausgabe:

```
[  
  {
```

```
    "VpnConnectionDeviceTypeId": "754a6372",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 4.1.2+"
  },
  {
    "VpnConnectionDeviceTypeId": "9612cbed",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 4.1.2+ (GUI)"
  },
  {
    "VpnConnectionDeviceTypeId": "5fb390ba",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 7.0+"
  }
]
```

Weitere Informationen finden [Sie im AWS Site-to-Site VPN Benutzerhandbuch unter Herunterladen der Konfigurationsdatei.](#)

- API-Einheiten finden Sie [GetVpnConnectionDeviceTypes](#) in der AWS CLI Befehlsreferenz.

import-client-vpn-client-certificate-revocation-list

Das folgende Codebeispiel zeigt die Verwendung `import-client-vpn-client-certificate-revocation-list`.

AWS CLI

Um eine Sperrliste für Client-Zertifikate zu importieren

Im folgenden `import-client-vpn-client-certificate-revocation-list` Beispiel wird eine Sperrliste für Client-Zertifikate auf den VPN Client-Endpoint importiert, indem der Speicherort der Datei auf dem lokalen Computer angegeben wird.

```
aws ec2 import-client-vpn-client-certificate-revocation-list \
  --certificate-revocation-list file:///path/to/crl.pem \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie im [VPNClient-Administratorhandbuch unter Widerrufslisten für AWS Client-Zertifikate](#).

- API-Einheiten finden Sie [ImportClientVpnClientCertificateRevocationList](#) in der AWS CLI Befehlsreferenz.

import-image

Das folgende Codebeispiel zeigt die Verwendung `import-image`.

AWS CLI

Um eine VM-Imagedatei zu importieren als AMI

Im folgenden `import-image` Beispiel wird die angegebene Datei importiert OVA.

```
aws ec2 import-image \
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-
server-vm.ova}"
```

Ausgabe:

```
{
  "ImportTaskId": "import-ami-1234567890abcdef0",
  "Progress": "2",
  "SnapshotDetails": [
    {
      "DiskImageSize": 0.0,
      "Format": "ova",
      "UserBucket": {
        "S3Bucket": "my-import-bucket",
        "S3Key": "vms/my-server-vm.ova"
      }
    }
  ],
  "Status": "active",
  "StatusMessage": "pending"
}
```

```
}
```

- API-Einheiten finden Sie [ImportImage](#) in der AWS CLI Befehlsreferenz.

import-key-pair

Das folgende Codebeispiel zeigt die Verwendung `import-key-pair`.

AWS CLI

Um einen öffentlichen Schlüssel zu importieren

Generieren Sie zunächst ein key pair mit dem Tool Ihrer Wahl. Verwenden Sie beispielsweise diesen Befehl `ssh-keygen`:

Befehl:

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

Ausgabe:

```
Generating public/private rsa key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ec2-user/.ssh/my-key.  
Your public key has been saved in /home/ec2-user/.ssh/my-key.pub.  
...
```

Dieser Beispielbefehl importiert den angegebenen öffentlichen Schlüssel.

Befehl:

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

Ausgabe:

```
{  
  "KeyName": "my-key",
```

```
"KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- API-Einheiten finden Sie [ImportKeyPair](#) in der AWS CLI Befehlsreferenz.

import-snapshot

Das folgende Codebeispiel zeigt die Verwendung `import-snapshot`.

AWS CLI

Um einen Snapshot zu importieren

Im folgenden `import-snapshot` Beispiel wird die angegebene Festplatte als Snapshot importiert.

```
aws ec2 import-snapshot \
  --description "My server VMDK" \
  --disk-container Format=VMDK,UserBucket={S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.vmdk}
```

Ausgabe:

```
{
  "Description": "My server VMDK",
  "ImportTaskId": "import-snap-1234567890abcdef0",
  "SnapshotTaskDetail": {
    "Description": "My server VMDK",
    "DiskImageSize": "0.0",
    "Format": "VMDK",
    "Progress": "3",
    "Status": "active",
    "StatusMessage": "pending",
    "UserBucket": {
      "S3Bucket": "my-import-bucket",
      "S3Key": "vms/my-server-vm.vmdk"
    }
  }
}
```

- API-Einheiten finden Sie [ImportSnapshot](#) in der AWS CLI Befehlsreferenz.

list-images-in-recycle-bin

Das folgende Codebeispiel zeigt die Verwendung `list-images-in-recycle-bin`.

AWS CLI

Um die Bilder im Papierkorb aufzulisten

Das folgende `list-images-in-recycle-bin` Beispiel listet alle Bilder auf, die sich derzeit im Papierkorb befinden.

```
aws ec2 list-images-in-recycle-bin
```

Ausgabe:

```
{
  "Images": [
    {
      "RecycleBinEnterTime": "2022-03-14T15:35:08.000Z",
      "Description": "Monthly AMI One",
      "RecycleBinExitTime": "2022-03-15T15:35:08.000Z",
      "Name": "AMI_01",
      "ImageId": "ami-0111222333444abcd"
    }
  ]
}
```

Weitere Informationen finden Sie unter [AMIs aus dem Papierkorb wiederherstellen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [ListImagesInRecycleBin](#) in der AWS CLI Befehlsreferenz.

list-snapshots-in-recycle-bin

Das folgende Codebeispiel zeigt die Verwendung `list-snapshots-in-recycle-bin`.

AWS CLI

Um Schnappschüsse im Papierkorb anzuzeigen

Im folgenden `list-snapshots-in-recycle-bin` Beispiel werden Informationen zu Snapshots im Papierkorb aufgeführt, darunter die Snapshot-ID, eine Beschreibung des

Snapshots, die ID des Volumes, von dem der Snapshot erstellt wurde, das Datum und die Uhrzeit, an dem der Snapshot gelöscht und in den Papierkorb verschoben wurde, sowie Datum und Uhrzeit, an dem der Aufbewahrungszeitraum abläuft.

```
aws ec2 list-snapshots-in-recycle-bin \  
  --snapshot-id snap-01234567890abcdef
```

Ausgabe:

```
{  
  "SnapshotRecycleBinInfo": [  
    {  
      "Description": "Monthly data backup snapshot",  
      "RecycleBinEnterTime": "2022-12-01T13:00:00.000Z",  
      "RecycleBinExitTime": "2022-12-15T13:00:00.000Z",  
      "VolumeId": "vol-abcdef09876543210",  
      "SnapshotId": "snap-01234567890abcdef"  
    }  
  ]  
}
```

Weitere Informationen zum Papierkorb für Amazon EBS finden Sie unter [Schnappschüsse aus dem Papierkorb wiederherstellen](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ListSnapshotsInRecycleBin](#) in AWS CLI der Befehlsreferenz.

lock-snapshot

Das folgende Codebeispiel zeigt die Verwendung `lock-snapshot`.

AWS CLI

Beispiel 1: Um einen Snapshot im Governance-Modus zu sperren

Im folgenden `lock-snapshot` Beispiel wird der angegebene Snapshot im Governance-Modus gesperrt.

```
aws ec2 lock-snapshot \  
  --snapshot-id snap-0b5e733b4a8df6e0d \  
  --lock-mode governance \  
  --
```

```
--lock-duration 365
```

Ausgabe:

```
{
  "SnapshotId": "snap-0b5e733b4a8df6e0d",
  "LockState": "governance",
  "LockDuration": 365,
  "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",
  "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00",
  "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00"
}
```

Weitere Informationen finden Sie unter [Snapshot Lock](#) im EBSAmazon-Benutzerhandbuch.

Beispiel 2: So sperren Sie einen Snapshot im Compliance-Modus

Im folgenden lock-snapshot Beispiel wird der angegebene Snapshot im Konformitätsmodus gesperrt.

```
aws ec2 lock-snapshot \  
  --snapshot-id snap-0163a8524c5b9901f \  
  --lock-mode compliance \  
  --cool-off-period 24 \  
  --lock-duration 365
```

Ausgabe:

```
{
  "SnapshotId": "snap-0b5e733b4a8df6e0d",
  "LockState": "compliance-cooloff",
  "LockDuration": 365,
  "CoolOffPeriod": 24,
  "CoolOffPeriodExpiresOn": "2024-05-06T01:02:20.527000+00:00",
  "LockCreatedOn": "2024-05-05T01:02:20.527000+00:00",
  "LockExpiresOn": "2025-05-05T01:02:20.527000+00:00",
  "LockDurationStartTime": "2024-05-05T01:02:20.527000+00:00"
}
```

Weitere Informationen finden Sie unter [Snapshot Lock](#) im EBSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [LockSnapshot](#) in der AWS CLI Befehlsreferenz.

modify-address-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-address-attribute`.

AWS CLI

Um das mit einer elastischen IP-Adresse verknüpfte Domainnamenattribut zu ändern

In den folgenden `modify-address-attribute` Beispielen wird das Domainnamenattribut einer elastischen IP-Adresse geändert.

Linux:

```
aws ec2 modify-address-attribute \  
  --allocation-id eipalloc-abcdef01234567890 \  
  --domain-name example.com
```

Windows:

```
aws ec2 modify-address-attribute ^  
  --allocation-id eipalloc-abcdef01234567890 ^  
  --domain-name example.com
```

Ausgabe:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.net."  
      "PtrRecordUpdate": {  
        "Value": "example.com.",  
        "Status": "PENDING"  
      }  
    }  
  ]  
}
```

Informationen zur Überwachung der ausstehenden Änderung und zum Anzeigen der geänderten Attribute einer elastischen IP-Adresse finden Sie [describe-addresses-attribute](#) in der AWS CLIBefehlsreferenz.

- APIEinzelheiten finden Sie [ModifyAddressAttribute](#) unter AWS CLI Befehlsreferenz.

modify-availability-zone-group

Das folgende Codebeispiel zeigt die Verwendung `modify-availability-zone-group`.

AWS CLI

Um eine Zonengruppe zu aktivieren

Das folgende `modify-availability-zone-group` Beispiel aktiviert die angegebene Zonengruppe.

```
aws ec2 modify-availability-zone-group \  
  --group-name us-west-2-lax-1 \  
  --opt-in-status opted-in
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen finden Sie unter [Regionen und Zonen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyAvailabilityZoneGroup](#) in der AWS CLI Befehlsreferenz.

modify-capacity-reservation-fleet

Das folgende Codebeispiel zeigt die Verwendung `modify-capacity-reservation-fleet`.

AWS CLI

Beispiel 1: Um die Gesamtzielkapazität einer Kapazitätsreservierungsflotte zu ändern

Im folgenden `modify-capacity-reservation-fleet` Beispiel wird die Gesamtzielkapazität der angegebenen Kapazitätsreservierungsflotte geändert. Wenn Sie die Gesamtzielkapazität einer Kapazitätsreservierungsflotte ändern, erstellt die Flotte automatisch neue Kapazitätsreservierungen oder ändert bzw. storniert bestehende Kapazitätsreservierungen in der Flotte, um die neue Gesamtzielkapazität zu erreichen. Sie können keine zusätzlichen Änderungen an einer Flotte vornehmen, wenn sie sich im Zustand `modifying` befindet.

```
aws ec2 modify-capacity-reservation-fleet \  
  --fleet-id fleet-id \  
  --total-target-capacity total-target-capacity
```

```
--capacity-reservation-fleet-id crf-01234567890abcdef \  
--total-target-capacity 160
```

Ausgabe:

```
{  
  "Return": true  
}
```

Beispiel 2: Um das Enddatum einer Kapazitätsreservierungsflotte zu ändern

Im folgenden `modify-capacity-reservation-fleet` Beispiel wird das Enddatum der angegebenen Kapazitätsreservierungsflotte geändert. Wenn Sie das Enddatum für die Flotte ändern, werden die Enddaten für alle einzelnen Kapazitätsreservierungen entsprechend angepasst. Sie können keine zusätzlichen Änderungen an einer Flotte vornehmen, wenn sie sich im Zustand `modifying` befindet.

```
aws ec2 modify-capacity-reservation-fleet \  
  --capacity-reservation-fleet-id crf-01234567890abcdef \  
  --end-date 2022-07-04T23:59:59.000Z
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen zu Kapazitätsreservierungsflotten finden Sie unter [Kapazitätsreservierungsflotten](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ModifyCapacityReservationFleet AWS CLI Befehlsreferenz](#).

modify-capacity-reservation

Das folgende Codebeispiel zeigt die Verwendung `modify-capacity-reservation`.

AWS CLI

Beispiel 1: Um die Anzahl der Instanzen zu ändern, die durch eine bestehende Kapazitätsreservierung reserviert wurden

Im folgenden `modify-capacity-reservation` Beispiel wird die Anzahl der Instances geändert, für die durch die Kapazitätsreservierung Kapazität reserviert wird.

```
aws ec2 modify-capacity-reservation \  
  --capacity-reservation-id cr-1234abcd56EXAMPLE \  
  --instance-count 5
```

Ausgabe:

```
{  
  "Return": true  
}
```

Beispiel 2: Um das Enddatum und die Endzeit für eine bestehende Kapazitätsreservierung zu ändern

Im folgenden `modify-capacity-reservation` Beispiel wird eine bestehende Kapazitätsreservierung so geändert, dass sie am angegebenen Datum und zur angegebenen Uhrzeit endet.

```
aws ec2 modify-capacity-reservation \  
  --capacity-reservation-id cr-1234abcd56EXAMPLE \  
  --end-date-type Limited \  
  --end-date 2019-08-31T23:59:59Z
```

Weitere Informationen finden Sie unter [Ändern einer Kapazitätsreservierung](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyCapacityReservation](#) unter AWS CLI Befehlsreferenz.

modify-client-vpn-endpoint

Das folgende Codebeispiel zeigt die Verwendung `modify-client-vpn-endpoint`.

AWS CLI

Um einen VPN Client-Endpoint zu ändern

Im folgenden `modify-client-vpn-endpoint` Beispiel wird die Protokollierung von Client-Verbindungen für den angegebenen VPN Client-Endpoint aktiviert.

```
aws ec2 modify-client-vpn-endpoint \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen finden Sie unter [VPNClient-Endpunkte](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [ModifyClientVpnEndpoint](#) in der AWS CLI Befehlsreferenz.

modify-default-credit-specification

Das folgende Codebeispiel zeigt die Verwendung `modify-default-credit-specification`.

AWS CLI

Um die Standard-Kreditoption zu ändern

Im folgenden `modify-default-credit-specification` Beispiel wird die Standard-Kreditoption für T2-Instances geändert.

```
aws ec2 modify-default-credit-specification \  
  --instance-family t2 \  
  --cpu-credits unlimited
```

Ausgabe:

```
{  
  "InstanceFamilyCreditSpecification": {  
    "InstanceFamily": "t2",  
    "CpuCredits": "unlimited"  
  }  
}
```

- API-Einheiten finden Sie unter [ModifyDefaultCreditSpecification AWS CLI](#) Befehlsreferenz.

modify-ebs-default-kms-key-id

Das folgende Codebeispiel zeigt die Verwendung `modify-ebs-default-kms-key-id`.

AWS CLI

Um Ihren Standard CMK für die EBS Verschlüsselung festzulegen

Im folgenden `modify-ebs-default-kms-key-id` Beispiel wird der angegebene Wert CMK als Standard CMK für die EBS Verschlüsselung für Ihr AWS Konto in der aktuellen Region festgelegt.

```
aws ec2 modify-ebs-default-kms-key-id \  
  --kms-key-id alias/my-cmk
```

Ausgabe:

```
{  
  "KmsKeyId": "arn:aws:kms:us-  
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"  
}
```

- API-Einheiten finden Sie [ModifyEbsDefaultKmsKeyId](#) unter AWS CLI Befehlsreferenz.

modify-fleet

Das folgende Codebeispiel zeigt die Verwendung `modify-fleet`.

AWS CLI

Um eine EC2 Flotte zu skalieren

Im folgenden `modify-fleet` Beispiel wird die Zielkapazität der angegebenen EC2 Flotte geändert. Wenn der angegebene Wert größer als die aktuelle Kapazität ist, startet die EC2 Flotte zusätzliche Instances. Wenn der angegebene Wert unter der aktuellen Kapazität liegt, storniert die EC2 Flotte alle offenen Anfragen. Wenn die Kündigungsrichtlinie `giltterminate`, beendet die EC2 Flotte alle Instances, die die neue Zielkapazität überschreiten.

```
aws ec2 modify-fleet \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --target-capacity-specification TotalTargetCapacity=5
```


Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie unter [Verwaltung einer EC2 Flotte](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyFleet](#) in der AWS CLI Befehlsreferenz.

modify-fpga-image-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-fpga-image-attribute`.

AWS CLI

So ändern Sie die Attribute eines FPGA Amazon-Bilds

In diesem Beispiel werden Ladeberechtigungen `123456789012` für die angegebene Konto-ID hinzugefügt.

Befehl:

```
aws ec2 modify-fpga-image-attribute --attribute LoadPermission --fpga-image-id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

Ausgabe:

```
{
  "FpgaImageAttribute": {
    "FpgaImageId": "afi-0d123e123bfc85abc",
    "LoadPermissions": [
      {
        "UserId": "123456789012"
      }
    ]
  }
}
```

- API-Einheiten finden Sie [ModifyFpgaImageAttribute](#) unter AWS CLI Befehlsreferenz.

modify-hosts

Das folgende Codebeispiel zeigt die Verwendung `modify-hosts`.

AWS CLI

Beispiel 1: So aktivieren Sie die automatische Platzierung für einen Dedicated Host

Das folgende `modify-hosts` Beispiel aktiviert die automatische Platzierung für einen Dedicated Host, sodass er alle Instance-Starts ohne Targeting akzeptiert, die seiner Instance-Typ-Konfiguration entsprechen.

```
aws ec2 modify-hosts \  
  --host-id h-06c2f189b4EXAMPLE \  
  --auto-placement on
```

Ausgabe:

```
{  
  "Successful": [  
    "h-06c2f189b4EXAMPLE"  
  ],  
  "Unsuccessful": []  
}
```

Beispiel 2: Um die Host-Wiederherstellung für einen Dedicated Host zu aktivieren

Das folgende `modify-hosts` Beispiel aktiviert die Host-Wiederherstellung für den angegebenen Dedicated Host.

```
aws ec2 modify-hosts \  
  --host-id h-06c2f189b4EXAMPLE \  
  --host-recovery on
```

Ausgabe:

```
{  
  "Successful": [  
    "h-06c2f189b4EXAMPLE"  
  ],  
  "Unsuccessful": []  
}
```

Weitere Informationen finden Sie unter [Modifying Dedicated Host Auto Placement](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyHosts](#) in der AWS CLI Befehlsreferenz.

modify-id-format

Das folgende Codebeispiel zeigt die Verwendung `modify-id-format`.

AWS CLI

Um das längere ID-Format für eine Ressource zu aktivieren

Im folgenden `modify-id-format` Beispiel wird das längere ID-Format für den `instance` Ressourcentyp aktiviert.

```
aws ec2 modify-id-format \  
  --resource instance \  
  --use-long-ids
```

Um das längere ID-Format für eine Ressource zu deaktivieren

Im folgenden `modify-id-format` Beispiel wird das längere ID-Format für den `instance` Ressourcentyp deaktiviert.

```
aws ec2 modify-id-format \  
  --resource instance \  
  --no-use-long-ids
```

Im folgenden `modify-id-format` Beispiel wird das längere ID-Format für alle unterstützten Ressourcentypen aktiviert, die sich innerhalb ihres Anmeldezeitraums befinden.

```
aws ec2 modify-id-format \  
  --resource all-current \  
  --use-long-ids
```

- API-Einheiten finden Sie unter [ModifyIdFormat AWS CLI](#) Befehlsreferenz.

modify-identity-id-format

Das folgende Codebeispiel zeigt die Verwendung `modify-identity-id-format`.

AWS CLI

Um zu ermöglichen, dass eine IAM Rolle länger IDs für eine Ressource verwendet wird

Das folgende `modify-identity-id-format` Beispiel ermöglicht es der IAM Rolle `EC2Role` in Ihrem AWS Konto, das Long-ID-Format für den `instance` Ressourcentyp zu verwenden.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:role/EC2Role \  
  --resource instance \  
  --use-long-ids
```

Um es einem IAM Benutzer zu ermöglichen, eine Ressource IDs länger zu verwenden

Das folgende `modify-identity-id-format` Beispiel ermöglicht es dem IAM Benutzer `AdminUser` in Ihrem AWS Konto, das längere ID-Format für den `volume` Ressourcentyp zu verwenden.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource volume \  
  --use-long-ids
```

Das folgende `modify-identity-id-format` Beispiel ermöglicht es dem IAM Benutzer `AdminUser` in Ihrem AWS Konto, das längere ID-Format für alle unterstützten Ressourcentypen zu verwenden, die sich innerhalb des Anmeldezeitraums befinden.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource all-current \  
  --use-long-ids
```

- API-Einheiten finden Sie [ModifyIdentityIdFormat](#) in der AWS CLI Befehlsreferenz.

modify-image-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-image-attribute`.

AWS CLI

Beispiel 1: Um eine zu AMI veröffentlichen

Im folgenden `modify-instance-attribute` Beispiel wird das angegebene Objekt AMI öffentlich gemacht.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{Group=all}]"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um eine AMI private

Das folgende `modify-instance-attribute` Beispiel macht das angegebene Objekt AMI privat.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{Group=all}]"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 3: Um einem AWS Konto die Startberechtigung zu erteilen

Im folgenden `modify-instance-attribute` Beispiel werden dem angegebenen AWS Konto Startberechtigungen erteilt.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{UserId=123456789012}]"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 4: Um einem AWS Konto die Startberechtigung zu entziehen

Im folgenden `modify-instance-attribute` Beispiel werden die Startberechtigungen aus dem angegebenen AWS Konto entfernt.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{UserId=123456789012}]"
```

- API-Einheiten finden Sie [ModifyImageAttribute](#) in der AWS CLI Befehlsreferenz.

modify-instance-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-attribute`.

AWS CLI

Beispiel 1: Um den Instanztyp zu ändern

Im folgenden `modify-instance-attribute` Beispiel wird der Instanztyp der angegebenen Instanz geändert. Die Instance muss sich im Status `stopped` befinden.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --instance-type "{\"Value\": \"m1.small\"}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um Enhanced Networking auf einer Instance zu aktivieren

Das folgende `modify-instance-attribute` Beispiel aktiviert Enhanced Networking für die angegebene Instanz. Die Instance muss sich im Status `stopped` befinden.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --sriov-net-support simple
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 3: Um das `sourceDestCheck` Attribut zu ändern

Im folgenden `modify-instance-attribute` Beispiel wird das `sourceDestCheck` Attribut der angegebenen Instanz auf `setzttrue`. Die Instanz muss sich in einer befindenVPC.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-  
check "{\"Value\": true}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 4: Um das `deleteOnTermination` Attribut des Root-Volumes zu ändern

Im folgenden `modify-instance-attribute` Beispiel wird das `deleteOnTermination` Attribut für das Root-Volume der angegebenen EBS Amazon-gestützten Instance auf `false` festgelegt. Standardmäßig ist dieses Attribut `true` für das Root-Volume bestimmt.

Befehl:

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\", \"Ebs\":  
{\"DeleteOnTermination\":false}}]"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 5: Um die an eine Instanz angehängten Benutzerdaten zu ändern

Im folgenden `modify-instance-attribute` Beispiel wird der Inhalt der Datei `UserData.txt` als `UserData` für die angegebene Instanz hinzugefügt.

Inhalt der Originaldatei `UserData.txt`:

```
#!/bin/bash  
yum update -y  
service httpd start  
chkconfig httpd on
```

Der Inhalt der Datei muss Base64-codiert sein. Der erste Befehl konvertiert die Textdatei in Base64 und speichert sie als neue Datei.

Linux/MacOS-Version des Befehls:

```
base64 UserData.txt > UserData.base64.txt
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Windows-Version des Befehls:

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >  
UserData.base64.txt
```

Ausgabe:

```
Input Length = 67  
Output Length = 152  
CertUtil: -encode command completed successfully.
```

Jetzt können Sie im folgenden CLI Befehl auf diese Datei verweisen:

```
aws ec2 modify-instance-attribute \  
  --instance-id=i-09b5a14dbca622e76 \  
  --attribute userData --value file://UserData.base64.txt
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerdaten und AWS CLI im EC2 Benutzerhandbuch](#).

- API-Einheiten finden Sie [ModifyInstanceAttribute](#) in der AWS CLI Befehlsreferenz.

modify-instance-capacity-reservation-attributes

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-capacity-reservation-attributes`.

AWS CLI

Beispiel 1: So ändern Sie die Targeting-Einstellungen für Kapazitätsreservierungen einer Instance

Im folgenden `modify-instance-capacity-reservation-attributes` Beispiel wird eine gestoppte Instance so geändert, dass sie auf eine bestimmte Kapazitätsreservierung abzielt.

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification  
  'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

Ausgabe:

```
{  
  "Return": true  
}
```

Beispiel 2: So ändern Sie die Targeting-Einstellungen für Kapazitätsreservierungen einer Instance

Im folgenden `modify-instance-capacity-reservation-attributes` Beispiel wird eine gestoppte Instance, die auf die angegebene Kapazitätsreservierung abzielt, so geändert, dass sie in jeder Kapazitätsreservierung gestartet wird, die übereinstimmende Attribute (Instance-Typ, Plattform, Availability Zone) und offene Instance-Übereinstimmungskriterien hat.


```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification 'CapacityReservationPreference=open'
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen finden Sie unter [Ändern der Kapazitätsreservierungseinstellungen einer Instance](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyInstanceCapacityReservationAttributes](#) in der AWS CLI Befehlsreferenz.

modify-instance-credit-specification

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-credit-specification`.

AWS CLI

Um die Kreditoption für die CPU-Nutzung einer Instance zu ändern

In diesem Beispiel wird die Kreditoption für die CPU-Nutzung der angegebenen Instanz in der angegebenen Region auf „unbegrenzt“ geändert. Gültige Kreditoptionen sind „Standard“ und „unbegrenzt“.

Befehl:

```
aws ec2 modify-instance-credit-specification --instance-credit-  
specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

Ausgabe:

```
{  
  "SuccessfulInstanceCreditSpecifications": [  
    {  
      "InstanceId": "i-1234567890abcdef0"  
    }  
  ],  
}
```

```
"UnsuccessfulInstanceCreditSpecifications": []  
}
```

- API-Einheiten finden Sie [ModifyInstanceCreditSpecification](#) in der AWS CLI Befehlsreferenz.

modify-instance-event-start-time

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-event-start-time`.

AWS CLI

Um die Startzeit des Ereignisses für eine Instanz zu ändern

Der folgende `modify-instance-event-start-time` Befehl zeigt, wie die Startzeit des Ereignisses für die angegebene Instanz geändert wird. Geben Sie die Ereignis-ID mithilfe des `--instance-event-id` Parameters an. Geben Sie das neue Datum und die neue Uhrzeit mithilfe des `--not-before` Parameters an.

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0  
--instance-event-id instance-event-0abcdef1234567890 --not-  
before 2019-03-25T10:00:00.000
```

Ausgabe:

```
"Event": {  
  "InstanceEventId": "instance-event-0abcdef1234567890",  
  "Code": "system-reboot",  
  "Description": "scheduled reboot",  
  "NotAfter": "2019-03-25T12:00:00.000Z",  
  "NotBefore": "2019-03-25T10:00:00.000Z",  
  "NotBeforeDeadline": "2019-04-22T21:00:00.000Z"  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Instances](#), die für einen Neustart geplant sind, im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyInstanceEventStartTime](#) in der AWS CLI Befehlsreferenz.

modify-instance-event-window

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-event-window`.

AWS CLI

Beispiel 1: Um den Zeitraum eines Ereignisfensters zu ändern

Im folgenden `modify-instance-event-window` Beispiel wird der Zeitraum eines Ereignisfensters geändert. Geben Sie den `time-range`-Parameter an, um den Zeitbereich zu ändern. Sie können außerdem den Parameter `cron-expression` nicht angeben.

```
aws ec2 modify-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --time-range StartWeekDay=monday, StartHour=2, EndWeekDay=wednesday, EndHour=8
```

Ausgabe:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "TimeRanges": [  
      {  
        "StartWeekDay": "monday",  
        "StartHour": 2,  
        "EndWeekDay": "wednesday",  
        "EndHour": 8  
      }  
    ],  
    "Name": "myEventWindowName",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-0abcdef1234567890",  
        "i-0be35f9acb8ba01f0"  
      ],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating",  
    "Tags": [  
      {  
        "Key": "K1",  
        "Value": "V1"  
      }  
    ]  
  }  
}
```

```
}
```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 2: Um eine Reihe von Zeiträumen für ein Veranstaltungsfenster zu ändern

Im folgenden `modify-instance-event-window` Beispiel wird der Zeitraum eines Ereignisfensters geändert. Geben Sie den `time-range`-Parameter an, um den Zeitbereich zu ändern. Sie können außerdem den Parameter `cron-expression` nicht angeben.

```
aws ec2 modify-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --time-range '["StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":  
  "wednesday", "EndHour": 8],  
  {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",  
  "EndHour": 8}]'
```

Ausgabe:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "TimeRanges": [  
      {  
        "StartWeekDay": "monday",  
        "StartHour": 2,  
        "EndWeekDay": "wednesday",  
        "EndHour": 8  
      },  
      {  
        "StartWeekDay": "thursday",  
        "StartHour": 2,  
        "EndWeekDay": "friday",  
        "EndHour": 8  
      }  
    ],  
    "Name": "myEventWindowName",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-0abcdef1234567890",
```

```

        "i-0be35f9acb8ba01f0"
    ],
    "Tags": [],
    "DedicatedHostIds": []
  },
  "State": "creating",
  "Tags": [
    {
      "Key": "K1",
      "Value": "V1"
    }
  ]
}
}

```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

Beispiel 3: Um den Cron-Ausdruck eines Ereignisfensters zu ändern

Das folgende `modify-instance-event-window` Beispiel ändert den Cron-Ausdruck eines Ereignisfensters. Geben Sie den `cron-expression`-Parameter an, um den Cron-Ausdruck zu ändern. Sie können außerdem den Parameter `time-range` nicht angeben.

```

aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --cron-expression "* 21-23 * * 2,3"

```

Ausgabe:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [
        "i-0abcdef1234567890",
        "i-0be35f9acb8ba01f0"
      ],
      "Tags": [],
    }
  }
}

```

```

        "DedicatedHostIds": [],
    },
    "State": "creating",
    "Tags": [
        {
            "Key": "K1",
            "Value": "V1"
        }
    ]
}
}

```

Informationen zu den Einschränkungen des Veranstaltungsfensters finden Sie unter [Überlegungen](#) im Abschnitt Geplante Ereignisse des EC2Amazon-Benutzerhandbuchs.

- API-Einheiten finden Sie [ModifyInstanceEventWindow](#) in der AWS CLI Befehlsreferenz.

modify-instance-maintenance-options

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-maintenance-options`.

AWS CLI

Beispiel 1: Um das Wiederherstellungsverhalten einer Instanz zu deaktivieren

Im folgenden `modify-instance-maintenance-options` Beispiel wird die vereinfachte automatische Wiederherstellung für eine laufende oder angehaltene Instance deaktiviert.

```

aws ec2 modify-instance-maintenance-options \
  --instance-id i-0abcdef1234567890 \
  --auto-recovery disabled

```

Ausgabe:

```

{
  "InstanceId": "i-0abcdef1234567890",
  "AutoRecovery": "disabled"
}

```

Weitere Informationen finden Sie unter [Wiederherstellen Ihrer Instance](#) im EC2Amazon-Benutzerhandbuch für Linux-Instances.

Beispiel 2: So setzen Sie das Wiederherstellungsverhalten einer Instance auf die Standardeinstellung

Im folgenden `modify-instance-maintenance-options` Beispiel wird das automatische Wiederherstellungsverhalten auf den Standardwert gesetzt, wodurch eine vereinfachte automatische Wiederherstellung für unterstützte Instance-Typen ermöglicht wird.

```
aws ec2 modify-instance-maintenance-options \  
  --instance-id i-0abcdef1234567890 \  
  --auto-recovery default
```

Ausgabe:

```
{  
  "InstanceId": "i-0abcdef1234567890",  
  "AutoRecovery": "default"  
}
```

Weitere Informationen finden Sie unter [Wiederherstellen Ihrer Instance](#) im EC2Amazon-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyInstanceMaintenanceOptions](#) in der AWS CLI Befehlsreferenz.

modify-instance-metadata-options

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-metadata-options`.

AWS CLI

Beispiel 1: Um zu aktivieren IMDSv2

Im folgenden `modify-instance-metadata-options` Beispiel wird die Verwendung von IMDSv2 auf der angegebenen Instanz konfiguriert.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-tokens required \  
  --http-endpoint enabled
```

Ausgabe:

```
{
  "InstanceId": "i-1234567898abcdef0",
  "InstanceMetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "enabled"
  }
}
```

Weitere Informationen finden Sie unter [Instance-Metadaten und Benutzerdaten](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: So deaktivieren Sie Instance-Metadaten

Im folgenden `modify-instance-metadata-options` Beispiel wird die Verwendung aller Versionen von Instanzmetadaten auf der angegebenen Instanz deaktiviert.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-endpoint disabled
```

Ausgabe:

```
{
  "InstanceId": "i-1234567898abcdef0",
  "InstanceMetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "disabled"
  }
}
```

Weitere Informationen finden Sie unter [Instance-Metadaten und Benutzerdaten](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 3: So aktivieren Sie den IPv6 Instance-Metadaten-Endpunkt für Ihre Instance

Das folgende `modify-instance-metadata-options` Beispiel zeigt Ihnen, wie Sie den IPv6 Endpunkt für den Instanz-Metadatendienst aktivieren.


```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-protocol-ipv6 enabled \  
  --http-endpoint enabled
```

Ausgabe:

```
{  
  "InstanceId": "i-1234567898abcdef0",  
  "InstanceMetadataOptions": {  
    "State": "pending",  
    "HttpTokens": "required",  
    "HttpPutResponseHopLimit": 1,  
    "HttpEndpoint": "enabled",  
    "HttpProtocolIpv6": "enabled"  
  }  
}
```

Standardmäßig ist der IPv6 Endpunkt deaktiviert. Dies gilt auch dann, wenn Sie eine Instance in einem Subnetz gestartet haben, das IPv6 nur für das Subnetz zuständig ist. Der IPv6 Endpunkt für IMDS ist nur auf Instances zugänglich, die auf dem Nitro-System basieren. Weitere Informationen finden Sie unter [Instance-Metadaten und Benutzerdaten](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyInstanceMetadataOptions](#) in der AWS CLI Befehlsreferenz.

modify-instance-placement

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-placement`.

AWS CLI

Beispiel 1: Um die Affinität einer Instance zu einem Dedicated Host zu entfernen

Das folgende `modify-instance-placement` Beispiel entfernt die Affinität einer Instance zu einem Dedicated Host und ermöglicht es ihr, auf jedem verfügbaren Dedicated Host in Ihrem Konto zu starten, der ihren Instance-Typ unterstützt.

```
aws ec2 modify-instance-placement \  
  --instance-id i-0e6ddf6187EXAMPLE \  
  --affinity default
```

Ausgabe:

```
{
  "Return": true
}
```

Beispiel 2: Um eine Affinität zwischen einer Instance und dem angegebenen Dedicated Host herzustellen

Im folgenden `modify-instance-placement` Beispiel wird eine Startbeziehung zwischen einer Instance und einem Dedicated Host hergestellt. Die Instance kann nur auf dem angegebenen Dedicated Host ausgeführt werden.

```
aws ec2 modify-instance-placement \
  --instance-id i-0e6ddf6187EXAMPLE \
  --affinity host \
  --host-id i-0e6ddf6187EXAMPLE
```

Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie unter [Modifying Instance Tenancy and Affinity](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 3: So verschieben Sie eine Instance in eine Platzierungsgruppe

Im folgenden `modify-instance-placement` Beispiel wird eine Instance in eine Placement-Gruppe verschoben, die Instance gestoppt, die Instance-Platzierung geändert und die Instance anschließend neu gestartet.

```
aws ec2 stop-instances \
  --instance-ids i-0123a456700123456

aws ec2 modify-instance-placement \
  --instance-id i-0123a456700123456 \
  --group-name MySpreadGroup

aws ec2 start-instances \
```

```
--instance-ids i-0123a456700123456
```

Weitere Informationen finden Sie unter [Ändern der Platzierungsgruppe für eine Instance](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Beispiel 4: So entfernen Sie eine Instance aus einer Placement-Gruppe

Im folgenden `modify-instance-placement` Beispiel wird eine Instance aus einer Placement-Gruppe entfernt, indem die Instance gestoppt, die Instance-Platzierung geändert und die Instance anschließend neu gestartet wird. Im folgenden Beispiel wird eine leere Zeichenfolge („“) für den Namen der Platzierungsgruppe angegeben, um anzugeben, dass sich die Instance nicht in einer Platzierungsgruppe befinden soll.

Stoppen Sie die Instanz:

```
aws ec2 stop-instances \  
  --instance-ids i-0123a456700123456
```

Ändern Sie die Platzierung (Windows-Eingabeaufforderung, Linux und macOS):

```
aws ec2 modify-instance-placement \  
  --instance-id i-0123a456700123456 \  
  --group-name ""
```

Ändern Sie die Platzierung (Windows PowerShell):

```
aws ec2 modify-instance-placement `  
  --instance-id i-0123a456700123456 `  
  --group-name ""
```

Starten Sie die Instanz neu:

```
aws ec2 start-instances \  
  --instance-ids i-0123a456700123456
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen finden Sie unter [Modifying Instance Tenancy and Affinity](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [ModifyInstancePlacement](#) in der AWS CLI Befehlsreferenz.

modify-ipam-pool

Das folgende Codebeispiel zeigt die Verwendung `modify-ipam-pool`.

AWS CLI

Um einen IPAM Pool zu ändern

Im folgenden `modify-ipam-pool` Beispiel wird ein IPAM Pool geändert.

(Linux):

```
aws ec2 modify-ipam-pool \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --add-allocation-resource-tags "Key=Owner,Value=Build Team" \  
  --clear-allocation-default-netmask-length \  
  --allocation-min-netmask-length 14
```

(Windows):

```
aws ec2 modify-ipam-pool ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --add-allocation-resource-tags "Key=Owner,Value=Build Team" ^  
  --clear-allocation-default-netmask-length ^  
  --allocation-min-netmask-length 14
```

Ausgabe:

```
{  
  "IpamPool": {  
    "OwnerId": "123456789012",  
    "IpamPoolId": "ipam-pool-0533048da7d823723",  
    "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-  
pool-0533048da7d823723",  
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-02fc38cd4c48e7d38",  
    "IpamScopeType": "private",  
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
```

```
"IpamRegion": "us-east-1",
"Locale": "None",
"PoolDepth": 1,
"State": "modify-complete",
"AutoImport": true,
"AddressFamily": "ipv4",
"AllocationMinNetmaskLength": 14,
"AllocationMaxNetmaskLength": 26,
"AllocationResourceTags": [
  {
    "Key": "Environment",
    "Value": "Preprod"
  },
  {
    "Key": "Owner",
    "Value": "Build Team"
  }
]
}
```

Weitere Informationen finden Sie unter [Einen Pool bearbeiten](#) im VPCIPAM Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyIpamPool](#) in der AWS CLI Befehlsreferenz.

modify-ipam-resource-cidr

Das folgende Codebeispiel zeigt die Verwendung `modify-ipam-resource-cidr`.

AWS CLI

Um das einer Ressource CIDR zugewiesene zu ändern

Im folgenden `modify-ipam-resource-cidr` Beispiel wird eine Ressource CIDR geändert.

(Linux):

```
aws ec2 modify-ipam-resource-cidr \
  --current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
  --destination-ipam-scope-id ipam-scope-0da34c61fd189a141 \
  --resource-id vpc-010e1791024eb0af9 \
  --resource-cidr 10.0.1.0/24 \
```

```
--resource-region us-east-1 \  
--monitored
```

(Windows):

```
aws ec2 modify-ipam-resource-cidr ^  
--current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
--destination-ipam-scope-id ipam-scope-0da34c61fd189a141 ^  
--resource-id vpc-010e1791024eb0af9 ^  
--resource-cidr 10.0.1.0/24 ^  
--resource-region us-east-1 ^  
--monitored
```

Ausgabe:

```
{  
  "IpamResourceCidr": {  
    "IpamId": "ipam-08440e7a3acde3908",  
    "IpamScopeId": "ipam-scope-0da34c61fd189a141",  
    "IpamPoolId": "ipam-pool-0533048da7d823723",  
    "ResourceRegion": "us-east-1",  
    "ResourceOwnerId": "123456789012",  
    "ResourceId": "vpc-010e1791024eb0af9",  
    "ResourceCidr": "10.0.1.0/24",  
    "ResourceType": "vpc",  
    "ResourceTags": [  
      {  
        "Key": "Environment",  
        "Value": "Preprod"  
      },  
      {  
        "Key": "Owner",  
        "Value": "Build Team"  
      }  
    ],  
    "IpUsage": 0.0,  
    "ComplianceStatus": "noncompliant",  
    "ManagementState": "managed",  
    "OverlapStatus": "overlapping",  
    "VpcId": "vpc-010e1791024eb0af9"  
  }  
}
```

Weitere Informationen zum Verschieben von Ressourcen finden Sie unter [Verschieben von Ressourcen CIDRs zwischen Bereichen](#) im VPCIPAMAmazon-Benutzerhandbuch.

Weitere Informationen zum Ändern des Überwachungsstatus finden Sie unter [Ändern des Überwachungsstatus einer Ressource CIDRs](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [ModifyIpamResourceCidr](#) in der AWS CLI Befehlsreferenz.

modify-ipam-resource-discovery

Das folgende Codebeispiel zeigt die Verwendung `modify-ipam-resource-discovery`.

AWS CLI

Um die Betriebsregionen einer Ressourcenerkennung zu ändern

In diesem Beispiel sind Sie ein IPAM delegierter Administrator, der die Betriebsregionen einer Resource Discovery ändern möchte.

Gehen Sie wie folgt vor, um diese Anfrage abzuschließen:

Sie können eine standardmäßige Ressourcensuche nicht ändern und müssen der Besitzer der Ressourcenerkennung sein. Sie benötigen die Ressourcenerkennungs-ID, die Sie abrufen können. [describe-ipam-resource-discoveries](#)

Im folgenden `modify-ipam-resource-discovery` Beispiel wird eine nicht standardmäßige Ressourcensuche in Ihrem Konto geändert. AWS

```
aws ec2 modify-ipam-resource-discovery \  
  --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \  
  --add-operating-regions RegionName='us-west-1' \  
  --remove-operating-regions RegionName='us-east-2' \  
  --region us-east-1
```

Ausgabe:

```
{  
  "IpamResourceDiscovery": {  
    "OwnerId": "149977607591",  
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",  
    "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-  
discovery/ipam-res-disco-0365d2977fc1672fe",
```

```
    "IpamResourceDiscoveryRegion": "us-east-1",
    "Description": "Example",
    "OperatingRegions": [
      {
        "RegionName": "us-east-1"
      },
      {
        "RegionName": "us-west-1"
      }
    ],
    "IsDefault": false,
    "State": "modify-in-progress"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Ressourcenentdeckungen](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyIpamResourceDiscovery](#) in der AWS CLI Befehlsreferenz.

modify-ipam-scope

Das folgende Codebeispiel zeigt die Verwendung `modify-ipam-scope`.

AWS CLI

Um die Beschreibung eines Bereichs zu ändern

In diesem Szenario sind Sie ein IPAM delegierter Administrator, der die Beschreibung eines IPAM Bereichs ändern möchte.

Um diese Anfrage abzuschließen, benötigen Sie die Bereichs-ID, die Sie erhalten können.

[describe-ipam-scopes](#)

Im folgenden `modify-ipam-scope` Beispiel wird die Beschreibung des Bereichs aktualisiert.

```
aws ec2 modify-ipam-scope \
  --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \
  --description example \
  --region us-east-1
```

Ausgabe:


```
{
  "IpamScope": {
    "OwnerId": "320805250157",
    "IpamScopeId": "ipam-scope-0d3539a30b57dcdd1",
    "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0d3539a30b57dcdd1",
    "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
    "IpamRegion": "us-east-1",
    "IpamScopeType": "public",
    "IsDefault": true,
    "Description": "example",
    "PoolCount": 1,
    "State": "modify-in-progress"
  }
}
```

Weitere Informationen zu Bereichen finden Sie unter [So IPAM funktioniert](#) es im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyIpamScope](#) in der AWS CLI Befehlsreferenz.

modify-ipam

Das folgende Codebeispiel zeigt die Verwendung `modify-ipam`.

AWS CLI

Um ein zu ändern IPAM

Im folgenden `modify-ipam` Beispiel wird eine geändert, IPAM indem eine Betriebsregion hinzugefügt wird.

(Linux):

```
aws ec2 modify-ipam \
  --ipam-id ipam-08440e7a3acde3908 \
  --add-operating-regions RegionName=us-west-2
```

(Windows):

```
aws ec2 modify-ipam ^
  --ipam-id ipam-08440e7a3acde3908 ^
```

```
--add-operating-regions RegionName=us-west-2
```

Ausgabe:

```
{
  "Ipam": {
    "OwnerId": "123456789012",
    "IpamId": "ipam-08440e7a3acde3908",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
    "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
    "ScopeCount": 3,
    "OperatingRegions": [
      {
        "RegionName": "us-east-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-west-1"
      },
      {
        "RegionName": "us-west-2"
      }
    ],
    "State": "modify-in-progress"
  }
}
```

- API-Einheiten finden Sie [ModifyIpam](#) in der AWS CLI Befehlsreferenz.

modify-launch-template

Das folgende Codebeispiel zeigt die Verwendung `modify-launch-template`.

AWS CLI

Um die Standardversion der Startvorlage zu ändern

In diesem Beispiel wird Version 2 der angegebenen Startvorlage als Standardversion angegeben.

Befehl:

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-version 2
```

Ausgabe:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "WebServers",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}
```

- API-Einheiten finden Sie [ModifyLaunchTemplate](#) in der AWS CLI Befehlsreferenz.

modify-managed-prefix-list

Das folgende Codebeispiel zeigt die Verwendung `modify-managed-prefix-list`.

AWS CLI

Um eine Präfixliste zu ändern

Das folgende `modify-managed-prefix-list` Beispiel fügt der angegebenen Präfixliste einen Eintrag hinzu.

```
aws ec2 modify-managed-prefix-list \
  --prefix-list-id pl-0123456abcabc1 \
  --add-entries Cidr=10.1.0.0/16,Description=vpc-c \
  --current-version 1
```

Ausgabe:

```
{
  "PrefixList": {
    "PrefixListId": "pl-0123456abcabc1",
  }
}
```

```
    "AddressFamily": "IPv4",
    "State": "modify-in-progress",
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
    "PrefixListName": "vpc-cidrs",
    "MaxEntries": 10,
    "Version": 1,
    "OwnerId": "123456789012"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyManagedPrefixList](#) in der AWS CLI Befehlsreferenz.

modify-network-interface-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-network-interface-attribute`.

AWS CLI

Um das Attachment-Attribut einer Netzwerkschnittstelle zu ändern

Dieser Beispielbefehl ändert das `attachment` Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

Um das Beschreibungsattribut einer Netzwerkschnittstelle zu ändern

Dieser Beispielbefehl ändert das `description` Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
description "My description"
```

Um das `groupSet` Attribut einer Netzwerkschnittstelle zu ändern

Dieser Beispielbefehl ändert das `groupSet` Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --groups sg-903004f8 sg-1a2b3c4d
```

Um das sourceDestCheck Attribut einer Netzwerkschnittstelle zu ändern

Dieser Beispielbefehl ändert das sourceDestCheck Attribut der angegebenen Netzwerkschnittstelle.

Befehl:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-source-dest-check
```

- API-Einheiten finden Sie unter [ModifyNetworkInterfaceAttribute AWS CLI](#) Befehlsreferenz.

modify-private-dns-name-options

Das folgende Codebeispiel zeigt die Verwendung `modify-private-dns-name-options`.

AWS CLI

Um die Optionen für Instanz-Hostnamen zu ändern

Im folgenden `modify-private-dns-name-options` Beispiel wird die Option deaktiviert, auf DNS Abfragen für Instanz-Hostnamen mit DNS A-Einträgen zu antworten.

```
aws ec2 modify-private-dns-name-options \  
  --instance-id i-1234567890abcdef0 \  
  --no-enable-resource-name-dns-a-record
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen finden Sie unter [Hostnamentypen für EC2 Amazon-Instances](#) im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyPrivateDnsNameOptions](#) in der AWS CLI Befehlsreferenz.

modify-reserved-instances

Das folgende Codebeispiel zeigt die Verwendung `modify-reserved-instances`.

AWS CLI

Um Reserved Instances zu ändern

Mit diesem Beispielbefehl wird eine Reserved Instance in eine andere Availability Zone in derselben Region verschoben.

Befehl:

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=10
```

Ausgabe:

```
{  
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"  
}
```

Um die Netzwerkplattform von Reserved Instances zu ändern

Dieser Beispielbefehl konvertiert EC2 -Classic Reserved Instances in EC2 -VPC.

Befehl:

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-VPC,InstanceCount=5
```

Ausgabe:

```
{  
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"  
}
```

Weitere Informationen finden Sie unter [Modifying Your Reserved Instances](#) im EC2 Amazon-Benutzerhandbuch.

So ändern Sie die Instance-Größe von Reserved Instances

Mit diesem Beispielbefehl wird eine Reserved Instance mit 10 m1.small UNIX Linux/-Instances in us-west-1c geändert, sodass aus 8 m1.small-Instances 2 m1.large-Instances werden und die verbleibenden 2 m1.small zu 1 m1.medium-Instance in derselben Availability Zone werden. Befehl:

```
aws ec2 modify-reserved-instances --reserved-instances-ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=1,InstanceType=m1.medium
```

Ausgabe:

```
{
  "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

Weitere Informationen finden Sie unter Ändern der Instance-Größe Ihrer Reservierungen im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [ModifyReservedInstances](#) unter AWS CLI Befehlsreferenz.

modify-security-group-rules

Das folgende Codebeispiel zeigt die Verwendung `modify-security-group-rules`.

AWS CLI

Um die Regeln einer Sicherheitsgruppe zu ändern, um die Regelbeschreibung, das IP-Protokoll und den CidrIpv 4-Adressbereich zu aktualisieren

Im folgenden `modify-security-group-rules` Beispiel werden die Beschreibung, das IP-Protokoll und der IPV4 CIDR Bereich einer angegebenen Sicherheitsgruppenregel aktualisiert. Verwenden Sie den `security-group-rules` Parameter, um die Updates für die angegebenen Sicherheitsgruppenregeln einzugeben. `-1` spezifiziert alle Protokolle.

```
aws ec2 modify-security-group-rules \
  --group-id sg-1234567890abcdef0 \
```

```
--security-group-rules SecurityGroupId=sgr-  
abcdef01234567890,SecurityGroupRule='{Description=test,IpProtocol=-1,CidrIpv4=0.0.0.0/0}'
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen zu Sicherheitsgruppenregeln finden Sie unter [Sicherheitsgruppenregeln](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifySecurityGroupRules](#) in der AWS CLI Befehlsreferenz.

modify-snapshot-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-snapshot-attribute`.

AWS CLI

Beispiel 1: Um ein Snapshot-Attribut zu ändern

Das folgende `modify-snapshot-attribute` Beispiel aktualisiert das `createVolumePermission` Attribut für den angegebenen Snapshot und entfernt die Volumenberechtigungen für den angegebenen Benutzer.

```
aws ec2 modify-snapshot-attribute \  
  --snapshot-id snap-1234567890abcdef0 \  
  --attribute createVolumePermission \  
  --operation-type remove \  
  --user-ids 123456789012
```

Beispiel 2: Um einen Snapshot öffentlich zu machen

Das folgende `modify-snapshot-attribute` Beispiel macht den angegebenen Snapshot öffentlich.

```
aws ec2 modify-snapshot-attribute \  
  --snapshot-id snap-1234567890abcdef0 \  
  --attribute createVolumePermission \  
  --operation-type add \  
  --user-ids 123456789012
```



```
--group-names all
```

- API-Einheiten finden Sie [ModifySnapshotAttribute](#) in der AWS CLI Befehlsreferenz.

modify-snapshot-tier

Das folgende Codebeispiel zeigt die Verwendung `modify-snapshot-tier`.

AWS CLI

Beispiel 1: Um einen Snapshot zu archivieren

Im folgenden `modify-snapshot-tier` Beispiel wird der angegebene Snapshot archiviert.

```
aws ec2 modify-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --storage-tier archive
```

Ausgabe:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "TieringStartTime": "2021-09-15T16:44:37.574Z"  
}
```

Der `TieringStartTime` Antwortparameter gibt das Datum und die Uhrzeit, zu denen der Archivierungsvorgang gestartet wurde, im UTC Zeitformat (YYYY-MM DDTHH -:MM:) an. SSZ

Weitere Informationen zur Snapshot-Archivierung finden Sie unter [Archivieren von EBS Amazon-Snapshots](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifySnapshotTier](#) in der AWS CLI Befehlsreferenz.

modify-spot-fleet-request

Das folgende Codebeispiel zeigt die Verwendung `modify-spot-fleet-request`.

AWS CLI

Um eine Spot-Flottenanfrage zu ändern

Dieser Beispielbefehl aktualisiert die Zielkapazität der angegebenen Spot-Flottenanforderung.

Befehl:

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Ausgabe:

```
{
  "Return": true
}
```

Dieser Beispielbefehl verringert die Zielkapazität der angegebenen Spot-Flottenanforderung, ohne dass dadurch Spot-Instances beendet werden.

Befehl:

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden Sie [ModifySpotFleetRequest](#) in der AWS CLI Befehlsreferenz.

modify-subnet-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-subnet-attribute`.

AWS CLI

Um das Verhalten eines Subnetzes bei der öffentlichen IPv4 Adressierung zu ändern

In diesem Beispiel wird Subnet-1a2b3c4d dahingehend geändert, dass allen in diesem Subnetz gestarteten Instances eine öffentliche Adresse zugewiesen wird. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-launch
```

Um IPv6 das Adressierungsverhalten eines Subnetzes zu ändern

In diesem Beispiel wird Subnet-1a2b3c4d dahingehend geändert, dass allen in diesem Subnetz gestarteten Instances eine Adresse aus dem Bereich des Subnetzes zugewiesen wird. IPv6

Befehl:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

Weitere Informationen finden Sie unter IP Addressing in Your im Virtual Private Cloud-Benutzerhandbuch. VPC AWS

- API-Einheiten finden Sie [ModifySubnetAttribute](#) in der AWS CLI Befehlsreferenz.

modify-traffic-mirror-filter-network-services

Das folgende Codebeispiel zeigt die Verwendung `modify-traffic-mirror-filter-network-services`.

AWS CLI

Um Netzwerkdienste zu einem Traffic Mirror-Filter hinzuzufügen

Im folgenden `modify-traffic-mirror-filter-network-services` Beispiel werden die DNS Amazon-Netzwerkdienste zum angegebenen Filter hinzugefügt.

```
aws ec2 modify-traffic-mirror-filter-network-services \
  --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \
  --add-network-service amazon-dns
```

Ausgabe:

```
{
  "TrafficMirrorFilter": {
    "Tags": [
      {
        "Key": "Name",
```

```

        "Value": "Production"
    }
],
"EgressFilterRules": [],
"NetworkServices": [
    "amazon-dns"
],
"TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
"IngressFilterRules": [
    {
        "SourceCidrBlock": "0.0.0.0/0",
        "RuleNumber": 1,
        "DestinationCidrBlock": "0.0.0.0/0",
        "Description": "TCP Rule",
        "Protocol": 6,
        "TrafficDirection": "ingress",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "RuleAction": "accept",
        "TrafficMirrorFilterRuleId": "tmf-04812ff784EXAMPLE"
    }
]
}
}

```

Weitere Informationen finden Sie unter [Modify Traffic Mirror Filter Network Services](#) im AWS Traffic Mirroring Guide.

- API-Einheiten finden Sie unter [ModifyTrafficMirrorFilterNetworkServices AWS CLI](#) Befehlsreferenz.

modify-traffic-mirror-filter-rule

Das folgende Codebeispiel zeigt die Verwendung `modify-traffic-mirror-filter-rule`.

AWS CLI

Um eine Traffic Mirror-Filterregel zu ändern

Im folgenden `modify-traffic-mirror-filter-rule` Beispiel wird die Beschreibung der angegebenen Filterregel für Traffic Mirror geändert.

```

aws ec2 modify-traffic-mirror-filter-rule \
  --traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \

```

```
--description "TCP Rule"
```

Ausgabe:

```
{
  "TrafficMirrorFilterRule": {
    "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
    "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
    "TrafficDirection": "ingress",
    "RuleNumber": 100,
    "RuleAction": "accept",
    "Protocol": 6,
    "DestinationCidrBlock": "10.0.0.0/24",
    "SourceCidrBlock": "10.0.0.0/24",
    "Description": "TCP Rule"
  }
}
```

Weitere Informationen finden Sie unter [Ändern Ihrer Traffic Mirroring-Filterregeln](#) im AWS Traffic Mirroring Guide.

- API-Einheiten finden Sie unter [ModifyTrafficMirrorFilterRule AWS CLI Befehlsreferenz](#).

modify-traffic-mirror-session

Das folgende Codebeispiel zeigt die Verwendung `modify-traffic-mirror-session`.

AWS CLI

Um eine Traffic Mirror-Sitzung zu ändern

Im folgenden `modify-traffic-mirror-session` Beispiel werden die Beschreibung der Traffic Mirror-Sitzung und die Anzahl der zu spiegelnden Pakete geändert.

```
aws ec2 modify-traffic-mirror-session \
  --description "Change packet length" \
  --traffic-mirror-session-id tms-08a33b1214EXAMPLE \
  --remove-fields "packet-length"
```

Ausgabe:

```
{
```

```

    "TrafficMirrorSession": {
      "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
      "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
      "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
      "NetworkInterfaceId": "eni-070203f901EXAMPLE",
      "OwnerId": "111122223333",
      "SessionNumber": 1,
      "VirtualNetworkId": 7159709,
      "Description": "Change packet length",
      "Tags": []
    }
  }
}

```

Weitere Informationen finden Sie unter [Ändern Ihrer Traffic Mirror-Sitzung](#) im Traffic Mirroring Guide.

- API-Einheiten finden Sie unter [ModifyTrafficMirrorSession AWS CLI Befehlsreferenz](#).

modify-transit-gateway-prefix-list-reference

Das folgende Codebeispiel zeigt die Verwendung `modify-transit-gateway-prefix-list-reference`.

AWS CLI

Um einen Verweis auf eine Präfixliste zu ändern

Im folgenden `modify-transit-gateway-prefix-list-reference` Beispiel wird der Verweis auf die Präfixliste in der angegebenen Routentabelle geändert, indem der Anhang geändert wird, an den der Verkehr weitergeleitet wird.

```

aws ec2 modify-transit-gateway-prefix-list-reference \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --prefix-list-id pl-1111112222222333 \
  --transit-gateway-attachment-id tgw-attach-aabbccddaabbccaab

```

Ausgabe:

```

{
  "TransitGatewayPrefixListReference": {
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-1111112222222333",
  }
}

```

```

    "PrefixListOwnerId": "123456789012",
    "State": "modifying",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}

```

Weitere Informationen finden Sie unter [Referenzen zur Präfixliste](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Modify Transit Gateway Prefix List Reference AWS CLI Befehlsreferenz](#).

modify-transit-gateway-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `modify-transit-gateway-vpc-attachment`.

AWS CLI

Um einen VPC Transit-Gateway-Anhang zu ändern

Im folgenden `modify-transit-gateway-vpc-attachment` Beispiel wird dem angegebenen VPC Transit-Gateway-Anhang ein Subnetz hinzugefügt.

```

aws ec2 modify-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
  --add-subnet-ids subnet-0e51f45802EXAMPLE

```

Ausgabe:

```

{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-09fbd47ddfEXAMPLE",
    "TransitGatewayId": "tgw-0560315ccfEXAMPLE",
    "VpcId": "vpc-5eccc927",
    "VpcOwnerId": "111122223333",
    "State": "modifying",
    "SubnetIds": [
      "subnet-0e51f45802EXAMPLE",
    ]
  }
}

```

```

        "subnet-1EXAMPLE"
    ],
    "CreationTime": "2019-08-08T16:47:38.000Z",
    "Options": {
        "DnsSupport": "enable",
        "Ipv6Support": "disable"
    }
}
}

```

Weitere Informationen finden Sie unter [Transit Gateway-Anlagen zu einem VPC](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Modify Transit Gateway Vpc Attachment AWS CLI](#) Befehlsreferenz.

modify-transit-gateway

Das folgende Codebeispiel zeigt die Verwendung `modify-transit-gateway`.

AWS CLI

Um ein Transit-Gateway zu ändern

Im folgenden `modify-transit-gateway` Beispiel wird das angegebene Transit-Gateway geändert, indem die ECMP Unterstützung für VPN Anlagen aktiviert wird.

```

aws ec2 modify-transit-gateway \
  --transit-gateway-id tgw-111111222222aaaaa \
  --options VpnEcmpSupport=enable

```

Ausgabe:

```

{
  "TransitGateway": {
    "TransitGatewayId": "tgw-111111222222aaaaa",
    "TransitGatewayArn": "64512",
    "State": "modifying",
    "OwnerId": "123456789012",
    "CreationTime": "2020-04-30T08:41:37.000Z",
    "Options": {
      "AmazonSideAsn": 64512,

```



```

        "AutoAcceptSharedAttachments": "disable",
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
    }
}
}

```

Weitere Informationen finden Sie unter [Transit Gateways](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Modify Transit Gateway AWS CLI Befehlsreferenz](#).

modify-verified-access-endpoint-policy

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-endpoint-policy`.

AWS CLI

Um die Verified Access-Richtlinie für einen Endpunkt zu konfigurieren

Im folgenden `modify-verified-access-endpoint-policy` Beispiel wird die angegebene Verified Access-Richtlinie zum angegebenen Verified Access-Endpunkt hinzugefügt.

```

aws ec2 modify-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --policy-enabled \
  --policy-document file://policy.txt

```

Inhalt von `policy.txt`:

```

permit(principal,action,resource)
when {
  context.identity.groups.contains("finance") &&
  context.identity.email.verified == true
};

```

Ausgabe:

```
{
```

```
"PolicyEnabled": true,
"PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Weitere Informationen finden Sie unter [Richtlinien für verifizierten Zugriff](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessEndpointPolicy](#) in der AWS CLI Befehlsreferenz.

modify-verified-access-endpoint

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-endpoint`.

AWS CLI

Um die Konfiguration eines Verified Access-Endpunkts zu ändern

Im folgenden `modify-verified-access-endpoint` Beispiel wird dem angegebenen Verified Access-Endpunkt die angegebene Beschreibung hinzugefügt.

```
aws ec2 modify-verified-access-endpoint \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --description "Testing Verified Access"
```

Ausgabe:

```
{
  "VerifiedAccessEndpoint": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
    "ApplicationDomain": "example.com",
    "EndpointType": "network-interface",
    "AttachmentType": "vpc",
    "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
    "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
```

```
"SecurityGroupIds": [
  "sg-004915970c4c8f13a"
],
"NetworkInterfaceOptions": {
  "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
  "Protocol": "https",
  "Port": 443
},
"Status": {
  "Code": "updating"
},
"Description": "Testing Verified Access",
"CreationTime": "2023-08-25T20:54:43",
"LastUpdatedTime": "2023-08-25T22:46:32"
}
}
```

Weitere Informationen finden Sie unter [Verified Access-Endpunkte](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessEndpoint](#) in der AWS CLI Befehlsreferenz.

modify-verified-access-group-policy

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-group-policy`.

AWS CLI

So konfigurieren Sie eine Richtlinie für verifizierten Zugriff für eine Gruppe

Im folgenden `modify-verified-access-group-policy` Beispiel wird die angegebene Verified Access-Richtlinie zur angegebenen Verified Access-Gruppe hinzugefügt.

```
aws ec2 modify-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --policy-enabled \
  --policy-document file://policy.txt
```

Inhalt von `policy.txt`:

```
permit(principal,action,resource)
when {
```

```
context.identity.groups.contains("finance") &&
context.identity.email.verified == true
};
```

Ausgabe:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Weitere Informationen finden Sie unter [Verified Access-Gruppen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessGroupPolicy](#) in der AWS CLI Befehlsreferenz.

modify-verified-access-group

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-group`.

AWS CLI

Um die Konfiguration einer Verified Access-Gruppe zu ändern

Im folgenden `modify-verified-access-group` Beispiel wird die angegebene Beschreibung der angegebenen Verified Access-Gruppe hinzugefügt.

```
aws ec2 modify-verified-access-group \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --description "Testing Verified Access"
```

Ausgabe:

```
{
  "VerifiedAccessGroup": {
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "Owner": "123456789012",
```

```

    "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
    "CreationTime": "2023-08-25T19:55:19",
    "LastUpdatedTime": "2023-08-25T22:17:25"
  }
}

```

Weitere Informationen finden Sie unter [Verified Access-Gruppen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessGroup](#) in der AWS CLI Befehlsreferenz.

modify-verified-access-instance-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-instance-logging-configuration`.

AWS CLI

Um die Protokollierung für eine Verified Access-Instanz zu aktivieren

Im folgenden `modify-verified-access-instance-logging-configuration` Beispiel wird die Zugriffsprotokollierung für die angegebene Verified Access-Instanz aktiviert. Die Protokolle werden an die angegebene CloudWatch Logs-Protokollgruppe übermittelt.

```

aws ec2 modify-verified-access-instance-logging-configuration \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}

```

Ausgabe:

```

{
  "LoggingConfiguration": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "AccessLogs": {
      "S3": {
        "Enabled": false
      },
      "CloudWatchLogs": {
        "Enabled": true,
        "DeliveryStatus": {
          "Code": "success"
        }
      }
    }
  }
}

```

```

        },
        "LogGroup": "my-log-group"
    },
    "KinesisDataFirehose": {
        "Enabled": false
    },
    "LogVersion": "ocsf-1.0.0-rc.2",
    "IncludeTrustContext": false
}
}
}
}

```

Weitere Informationen finden Sie unter [Verified Access-Logs](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessInstanceLoggingConfiguration](#) in der AWS CLI Befehlsreferenz.

modify-verified-access-instance

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-instance`.

AWS CLI

Um die Konfiguration einer Verified Access-Instanz zu ändern

Im folgenden `modify-verified-access-instance` Beispiel wird der angegebenen Verified Access-Instanz die angegebene Beschreibung hinzugefügt.

```

aws ec2 modify-verified-access-instance \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --description "Testing Verified Access"

```

Ausgabe:

```

{
  "VerifiedAccessInstance": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "VerifiedAccessTrustProviders": [
      {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",

```

```
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center"
    }
],
"CreationTime": "2023-08-25T18:27:56",
"LastUpdatedTime": "2023-08-25T22:41:04"
}
}
```

Weitere Informationen finden Sie unter [Verified Access-Instanzen](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessInstance](#) in der AWS CLI Befehlsreferenz.

modify-verified-access-trust-provider

Das folgende Codebeispiel zeigt die Verwendung `modify-verified-access-trust-provider`.

AWS CLI

Um die Konfiguration eines Vertrauensanbieters mit verifiziertem Zugriff zu ändern

Im folgenden `modify-verified-access-trust-provider` Beispiel wird dem angegebenen Verified Access-Vertrauensanbieter die angegebene Beschreibung hinzugefügt.

```
aws ec2 modify-verified-access-trust-provider \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
  --description "Testing Verified Access"
```

Ausgabe:

```
{
  "VerifiedAccessTrustProvider": {
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
    "Description": "Testing Verified Access",
    "TrustProviderType": "user",
    "UserTrustProviderType": "iam-identity-center",
    "PolicyReferenceName": "idc",
    "CreationTime": "2023-08-25T19:00:38",
    "LastUpdatedTime": "2023-08-25T19:18:21"
  }
}
```

Weitere Informationen finden Sie unter [Vertrauensanbietern für verifizierten Zugriff](#) im AWS Verified Access-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyVerifiedAccessTrustProvider](#) in der AWS CLI Befehlsreferenz.

modify-volume-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-volume-attribute`.

AWS CLI

Um ein Volumenattribut zu ändern

In diesem Beispiel wird das `autoEnableIo` Attribut des Volumes mit der ID `vol-1234567890abcdef0` auf festgelegt `true`. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- API-Einheiten finden Sie [ModifyVolumeAttribute](#) in der AWS CLI Befehlsreferenz.

modify-volume

Das folgende Codebeispiel zeigt die Verwendung `modify-volume`.

AWS CLI

Beispiel 1: Um ein Volumen zu ändern, indem man seine Größe ändert

Im folgenden `modify-volume` Beispiel wird die Größe des angegebenen Volumes auf 150 GB geändert.

Befehl:

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

Ausgabe:

```
{  
  "VolumeModification": {
```



```

    "TargetSize": 150,
    "TargetVolumeType": "io1",
    "ModificationState": "modifying",
    "VolumeId": "vol-1234567890abcdef0",
    "TargetIops": 100,
    "StartTime": "2019-05-17T11:27:19.000Z",
    "Progress": 0,
    "OriginalVolumeType": "io1",
    "OriginalIops": 100,
    "OriginalSize": 100
  }
}

```

Beispiel 2: Um ein Volume zu ändern, indem Typ, Größe und IOPS Wert geändert werden

Im folgenden `modify-volume` Beispiel wird der Volumetyp in Bereitgestellt geändert IOPSSSD, die IOPS Zielrate auf 10000 und die Volumegröße auf 350 GB festgelegt.

```

aws ec2 modify-volume \
  --volume-type io1 \
  --iops 10000 \
  --size 350 \
  --volume-id vol-1234567890abcdef0

```

Ausgabe:

```

{
  "VolumeModification": {
    "TargetSize": 350,
    "TargetVolumeType": "io1",
    "ModificationState": "modifying",
    "VolumeId": "vol-0721c1a9d08c93bf6",
    "TargetIops": 10000,
    "StartTime": "2019-05-17T11:38:57.000Z",
    "Progress": 0,
    "OriginalVolumeType": "gp2",
    "OriginalIops": 150,
    "OriginalSize": 50
  }
}

```

- APIEinzelheiten finden Sie unter [ModifyVolume AWS CLI Befehlsreferenz](#).

modify-vpc-attribute

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-attribute`.

AWS CLI

Um das `enableDnsSupport` Attribut zu ändern

In diesem Beispiel wird das `enableDnsSupport` Attribut geändert. Dieses Attribut gibt an, ob die DNS Auflösung für aktiviert ist. VPC Wenn dieses Attribut zutrifft `true`, löst der DNS Amazon-Server DNS Hostnamen für Ihre Instances in die entsprechenden IP-Adressen auf; andernfalls nicht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value\n\":false}"
```

Um das Attribut zu ändern `enableDnsHostnames`

In diesem Beispiel wird das `enableDnsHostnames` Attribut geändert. Dieses Attribut gibt an, ob Instances im VPC Get DNS Hostnames gestartet wurden. Wenn dieses Attribut zutrifft `true`, VPC erhalten Instanzen in der DNS Gruppe Hostnamen, andernfalls nicht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value\n\":false}"
```

- API-Einheiten finden Sie [ModifyVpcAttribute](#) in der AWS CLI Befehlsreferenz.

modify-vpc-endpoint-connection-notification

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-endpoint-connection-notification`.

AWS CLI

Um eine Benachrichtigung über eine Endpunktverbindung zu ändern

In diesem Beispiel wird das SNS Thema für die angegebene Endpunktverbindungsbenachrichtigung geändert.

Befehl:

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-id vpce-nfn-008776de7e03f5abc --connection-events Accept Reject --connection-notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

Ausgabe:

```
{  
  "ReturnValue": true  
}
```

- API-Einheiten finden Sie [ModifyVpcEndpointConnectionNotification](#) in der AWS CLI Befehlsreferenz.

modify-vpc-endpoint-service-configuration

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-endpoint-service-configuration`.

AWS CLI

Um eine Endpunkt-Servicekonfiguration zu ändern

In diesem Beispiel wird die Akzeptanzanforderung für den angegebenen Endpunktdienst geändert.

Befehl:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --no-acceptance-required
```

Ausgabe:

```
{  
  "ReturnValue": true  
}
```

- API-Einheiten finden Sie [ModifyVpcEndpointServiceConfiguration](#) in der AWS CLI Befehlsreferenz.

modify-vpc-endpoint-service-payer-responsibility

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-endpoint-service-payer-responsibility`.

AWS CLI

Um die Verantwortung des Kostenträgers zu ändern

Im folgenden `modify-vpc-endpoint-service-payer-responsibility` Beispiel wird die Verantwortung des Zahlers für den angegebenen Endpunktdienst geändert.

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \  
  --service-id vpce-svc-071afff70666e61e0 \  
  --payer-responsibility ServiceOwner
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [ModifyVpcEndpointServicePayerResponsibility AWS CLI](#) Befehlsreferenz.

modify-vpc-endpoint-service-permissions

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-endpoint-service-permissions`.

AWS CLI

Um die Berechtigungen für Endpunktdienste zu ändern

In diesem Beispiel wird einem AWS Konto die Berechtigung hinzugefügt, eine Verbindung mit dem angegebenen Endpunktdienst herzustellen.

Befehl:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

Ausgabe:

```
{
  "ReturnValue": true
}
```

In diesem Beispiel wird einem bestimmten IAM Benutzer (admin) die Berechtigung hinzugefügt, eine Verbindung zum angegebenen Endpunktdienst herzustellen.

Befehl:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/  
admin"]'
```

- API-Einheiten finden Sie [ModifyVpcEndpointServicePermissions](#) unter AWS CLI Befehlsreferenz.

modify-vpc-endpoint

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-endpoint`.

AWS CLI

Um einen Gateway-Endpunkt zu ändern

In diesem Beispiel wird der Gateway-Endpunkt geändert, `vpce-1a2b3c4d` indem `rtb-aaa222bb` dem Endpunkt eine Routentabelle zugeordnet und das Richtliniendokument zurückgesetzt wird.

Befehl:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-  
ids rtb-aaa222bb --reset-policy
```

Ausgabe:

```
{
  "Return": true
}
```

```
}
```

Um einen Schnittstellenendpunkt zu ändern

In diesem Beispiel wird der Schnittstellenendpunkt geändert, `vpce-0fe5b17a0707d6fa5` indem dem Endpunkt ein Subnetz `subnet-d6fcaa8d` hinzugefügt wird.

Befehl:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-id subnet-d6fcaa8d
```

Ausgabe:

```
{  
  "Return": true  
}
```

- API-Einheiten finden Sie [ModifyVpcEndpoint](#) in der AWS CLI Befehlsreferenz.

modify-vpc-peering-connection-options

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-peering-connection-options`.

AWS CLI

Um die Kommunikation über eine VPC Peering-Verbindung von Ihrer lokalen ClassicLink Verbindung aus zu ermöglichen

In diesem Beispiel VPC ändert der Besitzer des Anforderers für eine Peering-Verbindung `pcx-aaaabbb` die VPC Peering-Verbindungsoptionen, sodass eine lokale ClassicLink Verbindung mit dem Peer kommunizieren kann. VPC

Befehl:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbb --requester-peering-connection-options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

Ausgabe:

```
{
  "RequesterPeeringConnectionOptions": {
    "AllowEgressFromLocalClassicLinkToRemoteVpc": true
  }
}
```

Um die Kommunikation über eine VPC Peering-Verbindung von Ihrer lokalen zu einer Remoteverbindung zu ermöglichen VPC ClassicLink

In diesem Beispiel VPC ändert der Besitzer des Akzepters die VPC Peering-Verbindungsoptionen, sodass der lokale Server mit der ClassicLink Verbindung im Peer kommunizieren VPC kann. VPC

Befehl:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --accepter-peering-connection-options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

Ausgabe:

```
{
  "AcceptorPeeringConnectionOptions": {
    "AllowEgressFromLocalVpcToRemoteClassicLink": true
  }
}
```

Um die DNS Auflösungsunterstützung für die Peering-Verbindung zu aktivieren VPC

In diesem Beispiel VPC ändert der Besitzer des Anforderers die VPC Peering-Verbindungsoptionen, sodass der lokale VPC Server öffentliche DNS Hostnamen in private IP-Adressen auflösen kann, wenn er von Instanzen im Peer abgefragt wird. *pcx-aaaabbbb* VPC

Befehl:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

Ausgabe:

```
{
```

```
"RequesterPeeringConnectionOptions": {  
  "AllowDnsResolutionFromRemoteVpc": true  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz. [ModifyVpcPeeringConnectionOptions](#) AWS CLI

modify-vpc-tenancy

Das folgende Codebeispiel zeigt die Verwendung `modify-vpc-tenancy`.

AWS CLI

Um das Mietverhältnis eines zu ändern VPC

In diesem Beispiel wird das Mietverhältnis von `default` bis geändert. VPC `vpc-1a2b3c4d`

Befehl:

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

Ausgabe:

```
{  
  "Return": true  
}
```

- API-Einheiten finden Sie [ModifyVpcTenancy](#) in der AWS CLI Befehlsreferenz.

modify-vpn-connection-options

Das folgende Codebeispiel zeigt die Verwendung `modify-vpn-connection-options`.

AWS CLI

Um Ihre VPN Verbindungsoptionen zu ändern

Im folgenden `modify-vpn-connection-options` Beispiel wird die lokale Verbindung IPv4 CIDR auf der Kunden-Gateway-Seite der angegebenen VPN Verbindung geändert.


```
aws ec2 modify-vpn-connection-options \  
--vpn-connection-id vpn-1122334455aabbccd \  
--local-ipv4-network-cidr 10.0.0.0/16
```

Ausgabe:

```
{  
  "VpnConnections": [  
    {  
      "CustomerGatewayConfiguration": "...configuration information...",  
      "CustomerGatewayId": "cgw-01234567abcde1234",  
      "Category": "VPN",  
      "State": "modifying",  
      "Type": "ipsec.1",  
      "VpnConnectionId": "vpn-1122334455aabbccd",  
      "TransitGatewayId": "tgw-00112233445566aab",  
      "Options": {  
        "EnableAcceleration": false,  
        "StaticRoutesOnly": true,  
        "LocalIpv4NetworkCidr": "10.0.0.0/16",  
        "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
        "TunnelInsideIpVersion": "ipv4"  
      },  
      "Routes": [],  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "CanadaVPN"  
        }  
      ],  
      "VgwTelemetry": [  
        {  
          "AcceptedRouteCount": 0,  
          "LastStatusChange": "2020-07-29T10:35:11.000Z",  
          "OutsideIpAddress": "203.0.113.3",  
          "Status": "DOWN",  
          "StatusMessage": ""  
        },  
        {  
          "AcceptedRouteCount": 0,  
          "LastStatusChange": "2020-09-02T09:09:33.000Z",  
          "OutsideIpAddress": "203.0.113.5",  
          "Status": "UP",  
        }  
      ]  
    }  
  ]  
}
```

```

    "StatusMessage": ""
  }
]
}

```

Weitere Informationen finden Sie im AWS Site-to-Site VPN Benutzerhandbuch unter [Ändern der Site-to-Site VPN Verbindungsoptionen](#).

- API Einzelheiten finden Sie [ModifyVpnConnectionOptions](#) unter AWS CLI Befehlsreferenz.

modify-vpn-connection

Das folgende Codebeispiel zeigt die Verwendung `modify-vpn-connection`.

AWS CLI

Um eine VPN Verbindung zu ändern

Im folgenden `modify-vpn-connection` Beispiel wird das Ziel-Gateway für die VPN Verbindung `vpn-12345678901234567` zum Virtual Private Gateway geändert `vgw-11223344556677889`:

```

aws ec2 modify-vpn-connection \
  --vpn-connection-id vpn-12345678901234567 \
  --vpn-gateway-id vgw-11223344556677889

```

Ausgabe:

```

{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "...configuration information...",
    "CustomerGatewayId": "cgw-aabbccdde1122334",
    "Category": "VPN",
    "State": "modifying",
    "Type": "ipsec.1",
    "VpnConnectionId": "vpn-12345678901234567",
    "VpnGatewayId": "vgw-11223344556677889",
    "Options": {
      "StaticRoutesOnly": false
    },
    "VgwTelemetry": [

```

```

    {
      "AcceptedRouteCount": 0,
      "LastStatusChange": "2019-07-17T07:34:00.000Z",
      "OutsideIpAddress": "18.210.3.222",
      "Status": "DOWN",
      "StatusMessage": "IPSEC IS DOWN"
    },
    {
      "AcceptedRouteCount": 0,
      "LastStatusChange": "2019-07-20T21:20:16.000Z",
      "OutsideIpAddress": "34.193.129.33",
      "Status": "DOWN",
      "StatusMessage": "IPSEC IS DOWN"
    }
  ]
}

```

- API-Einheiten finden Sie [ModifyVpnConnection](#) in der AWS CLI Befehlsreferenz.

modify-vpn-tunnel-certificate

Das folgende Codebeispiel zeigt die Verwendung `modify-vpn-tunnel-certificate`.

AWS CLI

Um ein VPN Tunnelzertifikat zu rotieren

Im folgenden `modify-vpn-tunnel-certificate` Beispiel wird das Zertifikat für den angegebenen Tunnel für eine VPN Verbindung rotiert

```

aws ec2 modify-vpn-tunnel-certificate \
  --vpn-tunnel-outside-ip-address 203.0.113.17 \
  --vpn-connection-id vpn-12345678901234567

```

Ausgabe:

```

{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "...configuration information...",
    "CustomerGatewayId": "cgw-aabbccdde1122334",
    "Category": "VPN",

```

```

    "State": "modifying",
    "Type": "ipsec.1",
    "VpnConnectionId": "vpn-12345678901234567",
    "VpnGatewayId": "vgw-11223344556677889",
    "Options": {
      "StaticRoutesOnly": false
    },
    "VgwTelemetry": [
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2019-09-11T17:27:14.000Z",
        "OutsideIpAddress": "203.0.113.17",
        "Status": "DOWN",
        "StatusMessage": "IPSEC IS DOWN",
        "CertificateArn": "arn:aws:acm:us-east-1:123456789101:certificate/c544d8ce-20b8-4fff-98b0-example"
      },
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2019-09-11T17:26:47.000Z",
        "OutsideIpAddress": "203.0.114.18",
        "Status": "DOWN",
        "StatusMessage": "IPSEC IS DOWN",
        "CertificateArn": "arn:aws:acm:us-east-1:123456789101:certificate/5ab64566-761b-4ad3-b259-example"
      }
    ]
  }
}

```

- API-Einheiten finden Sie unter [ModifyVpnTunnelCertificate AWS CLI](#) Befehlsreferenz.

modify-vpn-tunnel-options

Das folgende Codebeispiel zeigt die Verwendung `modify-vpn-tunnel-options`.

AWS CLI

Um die Tunneloptionen für eine VPN-Verbindung zu ändern

Im folgenden `modify-vpn-tunnel-options` Beispiel werden die Diffie-Hellman-Gruppen aktualisiert, die für den angegebenen Tunnel und die angegebene Verbindung zugelassen sind.

VPN

```
aws ec2 modify-vpn-tunnel-options \  
  --vpn-connection-id vpn-12345678901234567 \  
  --vpn-tunnel-outside-ip-address 203.0.113.17 \  
  --tunnel-options Phase1DHGroupNumbers=[{Value=14},{Value=15},{Value=16},  
{Value=17},{Value=18}],Phase2DHGroupNumbers=[{Value=14},{Value=15},{Value=16},  
{Value=17},{Value=18}]
```

Ausgabe:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-aabbccdde1122334",  
    "Category": "VPN",  
    "State": "available",  
    "Type": "ipsec.1",  
    "VpnConnectionId": "vpn-12345678901234567",  
    "VpnGatewayId": "vgw-11223344556677889",  
    "Options": {  
      "StaticRoutesOnly": false,  
      "TunnelOptions": [  
        {  
          "OutsideIpAddress": "203.0.113.17",  
          "Phase1DHGroupNumbers": [  
            {  
              "Value": 14  
            },  
            {  
              "Value": 15  
            },  
            {  
              "Value": 16  
            },  
            {  
              "Value": 17  
            },  
            {  
              "Value": 18  
            }  
          ],  
          "Phase2DHGroupNumbers": [  
            {  
              "Value": 14  
            }  
          ]  
        }  
      ]  
    }  
  }  
}
```

```
    },
    {
      "Value": 15
    },
    {
      "Value": 16
    },
    {
      "Value": 17
    },
    {
      "Value": 18
    }
  ]
},
{
  "OutsideIpAddress": "203.0.114.19"
}
]
},
"VgwTelemetry": [
  {
    "AcceptedRouteCount": 0,
    "LastStatusChange": "2019-09-10T21:56:54.000Z",
    "OutsideIpAddress": "203.0.113.17",
    "Status": "DOWN",
    "StatusMessage": "IPSEC IS DOWN"
  },
  {
    "AcceptedRouteCount": 0,
    "LastStatusChange": "2019-09-10T21:56:43.000Z",
    "OutsideIpAddress": "203.0.114.19",
    "Status": "DOWN",
    "StatusMessage": "IPSEC IS DOWN"
  }
]
}
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [ModifyVpnTunnelOptions](#).AWS CLI

monitor-instances

Das folgende Codebeispiel zeigt die Verwendung `monitor-instances`.

AWS CLI

So aktivieren Sie eine detaillierte Überwachung für eine Instance

Dieser Beispielbefehl aktiviert die detaillierte Überwachung für die angegebene Instance.

Befehl:

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "pending"
      }
    }
  ]
}
```

- API-Einheiten finden Sie [MonitorInstances](#) in der AWS CLI Befehlsreferenz.

move-address-to-vpc

Das folgende Codebeispiel zeigt die Verwendung `move-address-to-vpc`.

AWS CLI

Um eine Adresse zu verschieben nach EC2 - VPC

In diesem Beispiel wird die Elastic IP-Adresse 54.123.4.56 auf die Plattform - verschoben. EC2 VPC

Befehl:

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

Ausgabe:

```
{
  "Status": "MoveInProgress"
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [MoveAddressToVpc](#).AWS CLI

move-byoip-cidr-to-ipam

Das folgende Codebeispiel zeigt die Verwendung `move-byoip-cidr-to-ipam`.

AWS CLI

Um eine BYOIP CIDR zu übertragen IPAM

Das folgende `move-byoip-cidr-to-ipam` Beispiel überträgt ein BYOIP CIDR an IPAM.

(Linux):

```
aws ec2 move-byoip-cidr-to-ipam \
  --region us-west-2 \
  --ipam-pool-id ipam-pool-0a03d430ca3f5c035 \
  --ipam-pool-owner 111111111111 \
  --cidr 130.137.249.0/24
```

(Windows):

```
aws ec2 move-byoip-cidr-to-ipam ^
  --region us-west-2 ^
  --ipam-pool-id ipam-pool-0a03d430ca3f5c035 ^
  --ipam-pool-owner 111111111111 ^
  --cidr 130.137.249.0/24
```

Ausgabe:

```
{
  "ByoipCidr": {
```



```

    "Cidr": "130.137.249.0/24",
    "State": "pending-transfer"
  }
}

```

Weitere Informationen finden Sie unter [Tutorial: Eine bestehende Datei BYOIP IPv4 CIDR übertragen auf IPAM](#) im VPCIPAMAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [MoveByoipCidrToIpam](#) in der AWS CLI Befehlsreferenz.

network-insights-access-scope

Das folgende Codebeispiel zeigt die Verwendung `network-insights-access-scope`.

AWS CLI

Um Network Insights-Zugriffsbereiche zu erstellen

Im folgenden `create-network-insights-access-scope` Beispiel wird ein Network Insights-Zugriffsbereich in Ihrem AWS Konto erstellt.

```

aws ec2 create-network-insights-access-scope \
  --cli-input-json file://access-scope-file.json

```

Inhalt von `access-scope-file.json`:

```

{
  {
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "Resources": [
              "vpc-abcd12e3"
            ]
          }
        }
      }
    ],
    "ExcludePaths": [
      {
        "Source": {

```

```

        "ResourceStatement": {
            "ResourceTypes": [
                "AWS::EC2::InternetGateway"
            ]
        }
    ]
}

```

Ausgabe:

```

{
  "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111"
}{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-123456789222",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789222:network-insights-access-scope/nis-123456789222",
    "CreateDate": "2022-01-25T19:20:28.796000+00:00",
    "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
  },
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-04c0c0fbca737c404",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "Resources": [
              "vpc-abcd12e3"
            ]
          }
        }
      }
    ],
    "ExcludePaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
}

```

```
}  
  }  
    }  
      ]  
        }  
          }  
            }
```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer using the AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [NetworkInsightsAccessScope](#) unter AWS CLI Befehlsreferenz.

provision-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `provision-byoip-cidr`.

AWS CLI

Um einen Adressbereich bereitzustellen

Das folgende `provision-byoip-cidr` Beispiel stellt einen öffentlichen IP-Adressbereich zur Verwendung mit bereit AWS.

```
aws ec2 provision-byoip-cidr \  
  --cidr 203.0.113.25/24 \  
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Ausgabe:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "State": "pending-provision"  
  }  
}
```

Weitere Informationen zum Erstellen der Nachrichtenzeichenfolgen für den Autorisierungskontext finden Sie unter [Bring Your Own IP Addresses](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ProvisionByoipCidr](#) in der AWS CLI Befehlsreferenz.

provision-ipam-pool-cidr

Das folgende Codebeispiel zeigt die Verwendung `provision-ipam-pool-cidr`.

AWS CLI

Um a für CIDR einen IPAM Pool bereitzustellen

Das folgende `provision-ipam-pool-cidr` Beispiel stellt a CIDR für einen IPAM Pool bereit.

(Linux):

```
aws ec2 provision-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --cidr 10.0.0.0/24
```

(Windows):

```
aws ec2 provision-ipam-pool-cidr ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --cidr 10.0.0.0/24
```

Ausgabe:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "10.0.0.0/24",  
    "State": "pending-provision"  
  }  
}
```

Weitere Informationen finden Sie unter [Bereitstellung CIDRs für einen Pool](#) im VPCIPAMAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ProvisionIpamPoolCidr](#) in der AWS CLI Befehlsreferenz.

purchase-host-reservation

Das folgende Codebeispiel zeigt die Verwendung `purchase-host-reservation`.

AWS CLI

Um eine Reservierung für einen Dedicated Host zu erwerben

In diesem Beispiel wird das angegebene Reservierungsangebot für Dedicated Hosts für den angegebenen Dedicated Host in Ihrem Konto erworben.

Befehl:

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

Ausgabe:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
    {
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "HostIdSet": [
        "h-013abcd2a00cbd123"
      ],
      "HostReservationId": "hr-0d418a3a4ffc669ae",
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- API-Einheiten finden Sie [PurchaseHostReservation](#) in der AWS CLI Befehlsreferenz.

purchase-reserved-instances-offering

Das folgende Codebeispiel zeigt die Verwendung `purchase-reserved-instances-offering`.

AWS CLI

Um ein Reserved Instance-Angebot zu erwerben

Dieser Beispielbefehl veranschaulicht den Kauf eines Reserved Instance-Angebots, wobei eine Angebots-ID und die Anzahl der Instanzen angegeben werden.

Befehl:

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

Ausgabe:

```
{
  "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
}
```

- API-Einheiten finden Sie [PurchaseReservedInstancesOffering](#) unter AWS CLI Befehlsreferenz.

purchase-scheduled-instances

Das folgende Codebeispiel zeigt die Verwendung `purchase-scheduled-instances`.

AWS CLI

Um eine Scheduled Instance zu kaufen

In diesem Beispiel wird eine Scheduled Instance gekauft.

Befehl:

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

purchase-request.json:

```
[
  {
    "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
    "InstanceCount": 1
  }
]
```

Ausgabe:

```
{
  "ScheduledInstanceSet": [
    {
```

```

    "AvailabilityZone": "us-west-2b",
    "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
    "HourlyPrice": "0.095",
    "CreateDate": "2016-01-25T21:43:38.612Z",
    "Recurrence": {
      "OccurrenceDaySet": [
        1
      ],
      "Interval": 1,
      "Frequency": "Weekly",
      "OccurrenceRelativeToEnd": false,
      "OccurrenceUnit": ""
    },
    "Platform": "Linux/UNIX",
    "TermEndDate": "2017-01-31T09:00:00Z",
    "InstanceCount": 1,
    "SlotDurationInHours": 32,
    "TermStartDate": "2016-01-31T09:00:00Z",
    "NetworkPlatform": "EC2-VPC",
    "TotalScheduledInstanceHours": 1696,
    "NextSlotStartTime": "2016-01-31T09:00:00Z",
    "InstanceType": "c4.large"
  }
]
}

```

- API-Einheiten finden Sie in der Befehlsreferenz [PurchaseScheduledInstances](#).AWS CLI

reboot-instances

Das folgende Codebeispiel zeigt die Verwendung `reboot-instances`.

AWS CLI

Um eine EC2 Amazon-Instance neu zu starten

In diesem Beispiel wird die angegebene Instance neu gestartet. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

Weitere Informationen finden Sie unter Ihre Instance neu starten im Benutzerhandbuch für Amazon Elastic Compute Cloud.

- API-Einheiten finden Sie [RebootInstances](#) in der AWS CLI Befehlsreferenz.

register-image

Das folgende Codebeispiel zeigt die Verwendung `register-image`.

AWS CLI

Beispiel 1: Um einen AMI mithilfe einer Manifestdatei zu registrieren

Das folgende `register-image` Beispiel registriert eine AMI unter Verwendung der angegebenen Manifestdatei in Amazon S3.

```
aws ec2 register-image \  
  --name my-image \  
  --image-location my-s3-bucket/myimage/image.manifest.xml
```

Ausgabe:

```
{  
  "ImageId": "ami-1234567890EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Amazon Machine Images \(AMI\)](#) im EC2 Amazon-Benutzerhandbuch.

Beispiel 2: Um einen zu registrieren, AMI verwenden Sie einen Snapshot eines Root-Geräts

Im folgenden `register-image` Beispiel wird ein Benutzer registriert, der den AMI angegebenen Snapshot eines EBS Root-Volumes als Gerät verwendet `/dev/xvda`. Die Blockgerätozuweisung beinhaltet auch ein leeres EBS 100-GiB-Volume als Gerät `/dev/xvdf`.

```
aws ec2 register-image \  
  --name my-image \  
  --root-device-name /dev/xvda \  
  --block-device-mappings DeviceName=/dev/  
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f} DeviceName=/dev/  
xvdf,Ebs={VolumeSize=100}
```


Ausgabe:

```
{
  "ImageId": "ami-1a2b3c4d5eEXAMPLE"
}
```

Weitere Informationen finden Sie unter [Amazon Machine Images \(AMI\)](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [RegisterImage](#) in der AWS CLI Befehlsreferenz.

register-instance-event-notification-attributes

Das folgende Codebeispiel zeigt die Verwendung `register-instance-event-notification-attributes`.

AWS CLI

Beispiel 1: Um alle Tags in Ereignisbenachrichtigungen aufzunehmen

Das folgende `register-instance-event-notification-attributes` Beispiel umfasst alle Tags in Ereignisbenachrichtigungen.

```
aws ec2 register-instance-event-notification-attributes \
  --instance-tag-attribute IncludeAllTagsOfInstance=true
```

Ausgabe:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [],
    "IncludeAllTagsOfInstance": true
  }
}
```

Weitere Informationen finden Sie unter [Geplante Ereignisse für Ihre Instances](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Beispiel 2: Um bestimmte Tags in Ereignisbenachrichtigungen aufzunehmen

Das folgende `register-instance-event-notification-attributes` Beispiel enthält die angegebenen Tags in Ereignisbenachrichtigungen. Sie können keine Tags angeben, wenn dies `IncludeAllTagsOfInstance` der Fall ist `true`.

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

Ausgabe:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [  
      "tag-key1",  
      "tag-key2"  
    ],  
    "IncludeAllTagsOfInstance": false  
  }  
}
```

Weitere Informationen finden Sie unter [Geplante Ereignisse für Ihre Instances](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

- API-Einheiten finden Sie [RegisterInstanceEventNotificationAttributes](#) in der AWS CLI Befehlsreferenz.

register-transit-gateway-multicast-group-sources

Das folgende Codebeispiel zeigt die Verwendung `register-transit-gateway-multicast-group-sources`.

AWS CLI

Um eine Quelle bei einer Transit-Gateway-Multicast-Gruppe zu registrieren.

Im folgenden `register-transit-gateway-multicast-group-sources` Beispiel wird die angegebene Netzwerkschnittstellengruppenquelle bei einer Multicast-Gruppe registriert.

```
aws ec2 register-transit-gateway-multicast-group-sources \  
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \  
  --group-ip-address 224.0.1.0 \  
  --group-name tgw-mcast-domain-0c4905cef79d6e597
```

```
--network-interface-ids eni-07f290fc3c090cbae
```

Ausgabe:

```
{
  "RegisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Weitere Informationen finden Sie unter [Registrieren von Quellen mit einer Multicast-Gruppe](#) im AWS Transit Gateways-Benutzerhandbuch.

- API-Einheiten finden Sie unter [RegisterTransitGatewayMulticastGroupSources AWS CLI-Befehlsreferenz](#).

register-transit-gateway-multicast-group-members

Das folgende Codebeispiel zeigt die Verwendung `register-transit-gateway-multicast-group-members`.

AWS CLI

Um die Informationen zu den Multicast-Domänenzuordnungen des Transit-Gateways anzuzeigen

Im folgenden `register-transit-gateway-multicast-group-members` Beispiel werden die Zuordnungen für die angegebene Multicast-Domäne zurückgegeben.

```
aws ec2 register-transit-gateway-multicast-group-members \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
  --group-ip-address 224.0.1.0 \
  --network-interface-ids eni-0e246d32695012e81
```

Ausgabe:

```
{
  "RegisteredMulticastGroupMembers": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
```

```

    "RegisteredNetworkInterfaceIds": [
      "eni-0e246d32695012e81"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}

```

Weitere Informationen finden Sie unter [Verwaltung von Multicast-Domänen](#) im Transit Gateways-Benutzerhandbuch.

- API-Einheiten finden Sie unter [RegisterTransitGatewayMulticastGroupMembers AWS CLI-Befehlsreferenz](#).

register-transit-gateway-multicast-group-sources

Das folgende Codebeispiel zeigt die Verwendung `register-transit-gateway-multicast-group-sources`.

AWS CLI

Um eine Quelle bei einer Transit-Gateway-Multicast-Gruppe zu registrieren.

Im folgenden `register-transit-gateway-multicast-group-sources` Beispiel wird die angegebene Netzwerkschnittstellengruppenquelle bei einer Multicast-Gruppe registriert.

```

aws ec2 register-transit-gateway-multicast-group-sources \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
  --group-ip-address 224.0.1.0 \
  --network-interface-ids eni-07f290fc3c090cbae

```

Ausgabe:

```

{
  "RegisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}

```

Weitere Informationen finden Sie im Transit Gateways Guide unter [Managing Multicast-Domains](#).

- API-Einheiten finden Sie [RegisterTransitGatewayMulticastGroupSources](#) in der AWS CLI Befehlsreferenz.

reject-transit-gateway-peering-attachment

Das folgende Codebeispiel zeigt die Verwendung `reject-transit-gateway-peering-attachment`.

AWS CLI

Um einen Transit-Gateway-Peering-Anhang zurückzuweisen

Im folgenden `reject-transit-gateway-peering-attachment` Beispiel wird die angegebene Transit-Gateway-Peering-Anhangsanforderung zurückgewiesen. Der `--region` Parameter gibt die Region an, in der sich das Transit-Gateway für den Akzeptierer befindet.

```
aws ec2 reject-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
  --region us-east-2
```

Ausgabe:

```
{
  "TransitGatewayPeeringAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
    "RequesterTgwInfo": {
      "TransitGatewayId": "tgw-123abc05e04123abc",
      "OwnerId": "123456789012",
      "Region": "us-west-2"
    },
    "AcceptorTgwInfo": {
      "TransitGatewayId": "tgw-11223344aabbcc112",
      "OwnerId": "123456789012",
      "Region": "us-east-2"
    },
    "State": "rejecting",
    "CreationTime": "2019-12-09T11:50:31.000Z"
  }
}
```

Weitere Informationen finden Sie unter [Transit Gateway Peering Attachments](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Reject Transit Gateway Peering Attachment AWS CLI Befehlsreferenz](#).

reject-transit-gateway-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `reject-transit-gateway-vpc-attachment`.

AWS CLI

Um einen VPC Transit-Gateway-Anhang zurückzuweisen

Im folgenden `reject-transit-gateway-vpc-attachment` Beispiel wird der angegebene VPC Transit-Gateway-Anhang zurückgewiesen.

```
aws ec2 reject-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Ausgabe:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "VpcId": "vpc-07e8ffd50fEXAMPLE",
    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
      "subnet-0752213d59EXAMPLE"
    ],
    "CreationTime": "2019-07-10T17:33:46.000Z",
    "Options": {
      "DnsSupport": "enable",
      "Ipv6Support": "disable"
    }
  }
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Anlagen zu a VPC](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [RejectTransitGatewayVpcAttachment AWS CLI-Befehlsreferenz](#).

reject-transit-gateway-vpc-attachments

Das folgende Codebeispiel zeigt die Verwendung `reject-transit-gateway-vpc-attachments`.

AWS CLI

Um einen VPC Transit-Gateway-Anhang zurückzuweisen

Im folgenden `reject-transit-gateway-vpc-attachment` Beispiel wird der angegebene VPC Transit-Gateway-Anhang zurückgewiesen.

```
aws ec2 reject-transit-gateway-vpc-attachment \  
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Ausgabe:

```
{  
  "TransitGatewayVpcAttachment": {  
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",  
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",  
    "VpcId": "vpc-07e8ffd50fEXAMPLE",  
    "VpcOwnerId": "111122223333",  
    "State": "pending",  
    "SubnetIds": [  
      "subnet-0752213d59EXAMPLE"  
    ],  
    "CreationTime": "2019-07-10T17:33:46.000Z",  
    "Options": {  
      "DnsSupport": "enable",  
      "Ipv6Support": "disable"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Anlagen zu a VPC](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [RejectTransitGatewayVpcAttachments AWS CLI-Befehlsreferenz](#).

reject-vpc-endpoint-connections

Das folgende Codebeispiel zeigt die Verwendung `reject-vpc-endpoint-connections`.

AWS CLI

Um eine Verbindungsanfrage für einen Schnittstellenendpunkt abzulehnen

In diesem Beispiel wird die angegebene Endpunkt-Verbindungsanforderung für den angegebenen Endpunktdienst abgelehnt.

Befehl:

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --  
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Ausgabe:

```
{  
  "Unsuccessful": []  
}
```

- API-Einheiten finden Sie [RejectVpcEndpointConnections](#) in der AWS CLI Befehlsreferenz.

reject-vpc-peering-connection

Das folgende Codebeispiel zeigt die Verwendung `reject-vpc-peering-connection`.

AWS CLI

Um eine VPC Peering-Verbindung abzulehnen

In diesem Beispiel wird die angegebene VPC Peering-Verbindungsanforderung zurückgewiesen.

Befehl:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Ausgabe:

```
{
```



```
"Return": true
}
```

- API-Einheiten finden Sie [RejectVpcPeeringConnection](#) in der AWS CLI Befehlsreferenz.

release-address

Das folgende Codebeispiel zeigt die Verwendung `release-address`.

AWS CLI

Um Elastic IP-Adressen für EC2 -Classic zu veröffentlichen

In diesem Beispiel wird eine Elastic IP-Adresse für die Verwendung mit Instanzen in EC2 -Classic veröffentlicht. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 release-address --public-ip 198.51.100.0
```

Um eine Elastic IP-Adresse für EC2 - freizugeben VPC

In diesem Beispiel wird eine Elastic IP-Adresse zur Verwendung mit Instanzen in a veröffentlicht VPC. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- API-Einheiten finden Sie [ReleaseAddress](#) in der AWS CLI Befehlsreferenz.

release-hosts

Das folgende Codebeispiel zeigt die Verwendung `release-hosts`.

AWS CLI

Um einen Dedicated Host von Ihrem Konto freizugeben

Um einen Dedicated Host von deinem Konto freizugeben. Instanzen, die sich auf dem Host befinden, müssen gestoppt oder beendet werden, bevor der Host freigegeben werden kann.

Befehl:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

Ausgabe:

```
{
  "Successful": [
    "h-0029d6e3cacf1b3da"
  ],
  "Unsuccessful": []
}
```

- API-Einheiten finden Sie [ReleaseHosts](#) in der AWS CLI Befehlsreferenz.

release-ipam-pool-allocation

Das folgende Codebeispiel zeigt die Verwendung `release-ipam-pool-allocation`.

AWS CLI

Um eine IPAM Poolzuweisung freizugeben

In diesem Beispiel sind Sie ein IPAM delegierter Administrator, der versucht hat, einen IPAM Pool zu löschen, aber die Fehlermeldung erhalten hat, dass Sie den Pool nicht löschen können, solange der Pool über Zuweisungen verfügt. Sie verwenden diesen Befehl, um eine Poolzuweisung freizugeben.

Beachten Sie Folgendes:

Sie können diesen Befehl nur für benutzerdefinierte Zuweisungen verwenden. Um eine Zuordnung für eine Ressource zu entfernen, ohne die Ressource zu löschen, setzen Sie ihren überwachten Status mit [modify-ipam-resource-cidr](#). Um diese Anfrage abzuschließen, benötigen Sie die IPAM Pool-ID, die Sie abrufen können. [describe-ipam-pools](#) Sie benötigen außerdem die Zuordnungs-ID, die Sie mit abrufen können. [get-ipam-pool-allocations](#) Wenn Sie Zuweisungen nicht einzeln entfernen möchten, können Sie beim Löschen eines Pools die Option `--cascade option` verwenden, um alle Zuordnungen im Pool automatisch freizugeben, bevor Sie ihn löschen. Es gibt eine Reihe von Voraussetzungen, bevor Sie diesen Befehl ausführen können. Weitere Informationen finden Sie unter [Eine Zuordnung](#)

[freigeben](#) im VPCIPAMAmazon-Benutzerhandbuch. Das, `--region` in dem Sie diesen Befehl ausführen, muss das Gebietsschema des IPAM Pools sein, in dem sich die Zuordnung befindet.

Das folgende `release-ipam-pool-allocation` Beispiel gibt eine IPAM Pool-Zuweisung frei.

```
aws ec2 release-ipam-pool-allocation \  
  --ipam-pool-id ipam-pool-07bdd12d7c94e4693 \  
  --cidr 10.0.0.0/23 \  
  --ipam-pool-allocation-id ipam-pool-alloc-0e66a1f730da54791b99465b79e7d1e89 \  
  --region us-west-1
```

Ausgabe:

```
{  
  "Success": true  
}
```

Sobald Sie eine Zuordnung freigegeben haben, möchten Sie sie möglicherweise ausführen [delete-ipam-pool](#).

- APIEinzelheiten finden Sie [ReleaselpamPoolAllocation](#) in der AWS CLI Befehlsreferenz.

replace-iam-instance-profile-association

Das folgende Codebeispiel zeigt die Verwendung `replace-iam-instance-profile-association`.

AWS CLI

Um ein IAM Instanzprofil für eine Instanz zu ersetzen

In diesem Beispiel wird das durch die Zuordnung dargestellte IAM Instanzprofil `iip-assoc-060bae234aac2e7fa` mit dem genannten IAM Instanzprofil `AdminRole` ersetzt.

```
aws ec2 replace-iam-instance-profile-association \  
  --iam-instance-profile Name=AdminRole \  
  --association-id iip-assoc-060bae234aac2e7fa
```

Ausgabe:

```
{
```

```

    "IamInstanceProfileAssociation": {
      "InstanceId": "i-087711ddaf98f9489",
      "State": "associating",
      "AssociationId": "iip-assoc-0b215292fab192820",
      "IamInstanceProfile": {
        "Id": "AIPAJLNLDX3AMYZNWYYAY",
        "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"
      }
    }
  }
}

```

- API-Einheiten finden Sie [ReplacelamInstanceProfileAssociation](#) unter AWS CLI Befehlsreferenz.

replace-network-acl-association

Das folgende Codebeispiel zeigt die Verwendung `replace-network-acl-association`.

AWS CLI

Um das einem Subnetz ACL zugeordnete Netzwerk zu ersetzen

In diesem Beispiel wird das angegebene Netzwerk ACL dem Subnetz für die angegebene ACL Netzwerkzuordnung zugeordnet.

Befehl:

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --
network-acl-id acl-5fb85d36
```

Ausgabe:

```
{
  "NewAssociationId": "aclassoc-3999875b"
}
```

- API-Einheiten finden Sie unter [ReplaceNetworkAclAssociation AWS CLI](#) Befehlsreferenz.

replace-network-acl-entry

Das folgende Codebeispiel zeigt die Verwendung `replace-network-acl-entry`.

AWS CLI

Um einen ACL Netzwerkeintrag zu ersetzen

Dieses Beispiel ersetzt einen Eintrag für das angegebene NetzwerkACL. Die neue Regel 100 erlaubt eingehenden Verkehr von 203.0.113.12/24 auf UDP Port 53 () in jedes zugehörige Subnetz. DNS

Befehl:

```
aws ec2 replace-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 203.0.113.12/24 --rule-action allow
```

- Einzelheiten finden Sie in der Befehlsreferenz. API [ReplaceNetworkAcEntry](#) AWS CLI

replace-route-table-association

Das folgende Codebeispiel zeigt die Verwendung `replace-route-table-association`.

AWS CLI

Um die einem Subnetz zugeordnete Routing-Tabelle zu ersetzen

In diesem Beispiel wird die angegebene Routing-Tabelle dem Subnetz für die angegebene Routentabellenzuordnung zugeordnet.

Befehl:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

Ausgabe:

```
{
  "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- API Einzelheiten finden Sie unter [ReplaceRouteTableAssociation AWS CLI](#) Befehlsreferenz.

replace-route

Das folgende Codebeispiel zeigt die Verwendung `replace-route`.

AWS CLI

Um eine Route zu ersetzen

Dieses Beispiel ersetzt die angegebene Route in der angegebenen Routentabelle. Die neue Route entspricht der angegebenen CIDR und sendet den Verkehr an das angegebene Virtual Private Gateway. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- API-Einheiten finden Sie [ReplaceRoute](#) in der AWS CLI Befehlsreferenz.

replace-transit-gateway-route

Das folgende Codebeispiel zeigt die Verwendung `replace-transit-gateway-route`.

AWS CLI

Um die angegebene Route in der angegebenen Transit-Gateway-Routentabelle zu ersetzen

Das folgende `replace-transit-gateway-route` Beispiel ersetzt die Route in der angegebenen Transit-Gateway-Routentabelle.

```
aws ec2 replace-transit-gateway-route \
  --destination-cidr-block 10.0.2.0/24 \
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

Ausgabe:

```
{
  "Route": {
    "DestinationCidrBlock": "10.0.2.0/24",
    "TransitGatewayAttachments": [
      {
```

```
        "ResourceId": "vpc-4EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceType": "vpc"
    }
],
  "Type": "static",
  "State": "active"
}
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [Replace Transit Gateway Route AWS CLI Befehlsreferenz](#).

report-instance-status

Das folgende Codebeispiel zeigt die Verwendung `report-instance-status`.

AWS CLI

Um Statusfeedback für eine Instanz zu melden

Dieser Beispielbefehl meldet Statusfeedback für die angegebene Instanz.

Befehl:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --reason-codes unresponsive
```

- API-Einheiten finden Sie [ReportInstanceStatus](#) in der AWS CLI Befehlsreferenz.

request-spot-fleet

Das folgende Codebeispiel zeigt die Verwendung `request-spot-fleet`.

AWS CLI

Um eine Spot-Flotte im Subnetz mit dem niedrigsten Preis anzufordern

Dieser Beispielbefehl erstellt eine Spot-Flottenanforderung mit zwei Startspezifikationen, die sich nur je nach Subnetz unterscheiden. Die Spot-Flotte startet die Instances im angegebenen

Subnetz mit dem niedrigsten Preis. Wenn die Instances standardmäßig gestartet werdenVPC, erhalten sie standardmäßig eine öffentliche IP-Adresse. Wenn die Instances nicht standardmäßig gestartet werdenVPC, erhalten sie standardmäßig keine öffentliche IP-Adresse.

Beachten Sie, dass Sie in einer Spot-Flottenanfrage nicht verschiedene Subnetze aus derselben Availability Zone angeben können.

Befehl:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "SubnetId": "subnet-1a2b3c4d, subnet-3c4d5e6f",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Ausgabe:

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Um eine Spot-Flotte in der Availability Zone mit dem niedrigsten Preis anzufordern

Dieser Beispielbefehl erstellt eine Spot-Flottenanfrage mit zwei Startspezifikationen, die sich nur je nach Availability Zone unterscheiden. Die Spot-Flotte startet die Instances in der angegebenen Availability Zone mit dem niedrigsten Preis. Wenn Ihr Konto VPC nur EC2 — unterstützt, EC2 startet Amazon die Spot-Instances im Standardsubnetz der Availability Zone. Wenn Ihr Konto EC2 -Classic unterstützt, EC2 startet Amazon die Instances in EC2 -Classic in der Availability Zone.

Befehl:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "Placement": {
        "AvailabilityZone": "us-west-2a, us-west-2b"
      },
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Um Spot-Instances in einem Subnetz zu starten und ihnen öffentliche IP-Adressen zuzuweisen

Dieser Beispielbefehl weist Instances, die nicht standardmäßig gestartet wurden, öffentliche Adressen zu. VPC Beachten Sie, dass Sie bei der Angabe einer Netzwerkschnittstelle die Subnetz-ID und die Sicherheitsgruppen-ID über die Netzwerkschnittstelle angeben müssen.

Befehl:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "InstanceType": "m3.medium",
      "NetworkInterfaces": [
        {
          "DeviceIndex": 0,
          "SubnetId": "subnet-1a2b3c4d",
          "Groups": [ "sg-1a2b3c4d" ],
          "AssociatePublicIpAddress": true
        }
      ],
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Um eine Spot-Flotte unter Verwendung der diversifizierten Zuweisungsstrategie anzufordern

Dieser Beispielbefehl erstellt eine Spot-Flottenanforderung, die 30 Instances unter Verwendung der diversifizierten Zuweisungsstrategie startet. Die Startspezifikationen unterscheiden sich je nach Instance-Typ. Die Spot-Flotte verteilt die Instances auf die Startspezifikationen, sodass es von jedem Typ 10 Instances gibt.

Befehl:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c4.2xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "m3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    }
  ]
}
```

Weitere Informationen finden Sie unter Spot Fleet Requests im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [RequestSpotFleet](#) in der AWS CLI Befehlsreferenz.

request-spot-instances

Das folgende Codebeispiel zeigt die Verwendung `request-spot-instances`.

AWS CLI

Um Spot-Instances anzufordern

Dieser Beispielbefehl erstellt eine einmalige Spot-Instance-Anfrage für fünf Instances in der angegebenen Availability Zone. Wenn Ihr Konto VPC nur EC2 — unterstützt, EC2 startet Amazon die Instances im Standardsubnetz der angegebenen Availability Zone. Wenn Ihr Konto EC2 -Classic unterstützt, EC2 startet Amazon die Instances in EC2 -Classic in der angegebenen Availability Zone.

Befehl:

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Ausgabe:

```
{
  "SpotInstanceRequests": [
    {
      "Status": {
        "UpdateTime": "2014-03-25T20:54:21.000Z",
        "Code": "pending-evaluation",
        "Message": "Your Spot request has been submitted for review, and is pending evaluation."
      },
      "ProductDescription": "Linux/UNIX",
      "SpotInstanceRequestId": "sir-df6f405d",
      "State": "open",
      "LaunchSpecification": {
        "Placement": {
          "AvailabilityZone": "us-west-2a"
        },
        "ImageId": "ami-1a2b3c4d",
        "KeyName": "my-key-pair",
        "SecurityGroups": [
          {
            "GroupName": "my-security-group",
```

```

        "GroupId": "sg-1a2b3c4d"
      }
    ],
    "Monitoring": {
      "Enabled": false
    },
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    },
    "InstanceType": "m3.medium"
  },
  "Type": "one-time",
  "CreateTime": "2014-03-25T20:54:20.000Z",
  "SpotPrice": "0.050000"
},
...
]
}

```

Dieser Beispielbefehl erstellt eine einmalige Spot-Instance-Anfrage für fünf Instances im angegebenen Subnetz. Amazon EC2 startet die Instances im angegebenen Subnetz. Wenn VPC es sich nicht um eine Standardadresse handelt, erhalten die Instances standardmäßig keine öffentliche IP-Adresse.

Befehl:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```

{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}

```

Ausgabe:

```
{
  "SpotInstanceRequests": [
    {
      "Status": {
        "UpdateTime": "2014-03-25T22:21:58.000Z",
        "Code": "pending-evaluation",
        "Message": "Your Spot request has been submitted for review, and is
pending evaluation."
      },
      "ProductDescription": "Linux/UNIX",
      "SpotInstanceRequestId": "sir-df6f405d",
      "State": "open",
      "LaunchSpecification": {
        "Placement": {
          "AvailabilityZone": "us-west-2a"
        }
        "ImageId": "ami-1a2b3c4d"
        "SecurityGroups": [
          {
            "GroupName": "my-security-group",
            "GroupID": "sg-1a2b3c4d"
          }
        ]
        "SubnetId": "subnet-1a2b3c4d",
        "Monitoring": {
          "Enabled": false
        },
        "IamInstanceProfile": {
          "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        },
        "InstanceType": "m3.medium",
      },
      "Type": "one-time",
      "CreateTime": "2014-03-25T22:21:58.000Z",
      "SpotPrice": "0.050000"
    },
    ...
  ]
}
```

In diesem Beispiel wird den Spot-Instances, die Sie nicht standardmäßig starten, eine öffentliche IP-Adresse zugewiesen. VPC Beachten Sie, dass Sie bei der Angabe einer Netzwerkschnittstelle die Subnetz-ID und die Sicherheitsgruppen-ID über die Netzwerkschnittstelle angeben müssen.

Befehl:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
      "Groups": [ "sg-1a2b3c4d" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

- API-Einheiten finden Sie [RequestSpotInstances](#) in der AWS CLI Befehlsreferenz.

reset-address-attribute

Das folgende Codebeispiel zeigt die Verwendung `reset-address-attribute`.

AWS CLI

Um das mit einer elastischen IP-Adresse verknüpfte Domainnamenattribut zurückzusetzen

In den folgenden `reset-address-attribute` Beispielen wird das Domainnamenattribut einer elastischen IP-Adresse zurückgesetzt.

Linux:

```
aws ec2 reset-address-attribute \
  --allocation-id eipalloc-abcdef01234567890 \
  --attribute domain-name
```

Windows:

```
aws ec2 reset-address-attribute ^
  --allocation-id eipalloc-abcdef01234567890 ^
  --attribute domain-name
```

Ausgabe:

```
{
  "Addresses": [
    {
      "PublicIp": "192.0.2.0",
      "AllocationId": "eipalloc-abcdef01234567890",
      "PtrRecord": "example.com."
      "PtrRecordUpdate": {
        "Value": "example.net.",
        "Status": "PENDING"
      }
    }
  ]
}
```

Informationen zur Überwachung der ausstehenden Änderung finden Sie [describe-addresses-attribute](#) in der AWS CLIBefehlsreferenz.

- APIEinzelheiten finden Sie [ResetAddressAttribute](#) unter AWS CLI Befehlsreferenz.

reset-efs-default-kms-key-id

Das folgende Codebeispiel zeigt die Verwendung `reset-efs-default-kms-key-id`.

AWS CLI

So setzen Sie Ihre Standardeinstellung CMK für die EBS Verschlüsselung zurück

Im folgenden `reset-efs-default-kms-key-id` Beispiel wird die EBS Standardverschlüsselung CMK für Ihr AWS Konto in der aktuellen Region zurückgesetzt.

```
aws ec2 reset-efs-default-kms-key-id
```

Ausgabe:

```
{
```



```
"KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513eEXAMPLE"
}
```

- API-Einheiten finden Sie [ResetEbsDefaultKmsKeyId](#) in der AWS CLI Befehlsreferenz.

reset-fpga-image-attribute

Das folgende Codebeispiel zeigt die Verwendung `reset-fpga-image-attribute`.

AWS CLI

So setzen Sie die Attribute eines FPGA Amazon-Images zurück

In diesem Beispiel werden die Ladeberechtigungen für das angegebene AFI Objekt zurückgesetzt.

Befehl:

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --attribute LoadPermission
```

Ausgabe:

```
{
  "Return": true
}
```

- API-Einheiten finden Sie [ResetFpgaImageAttribute](#) in der AWS CLI Befehlsreferenz.

reset-image-attribute

Das folgende Codebeispiel zeigt die Verwendung `reset-image-attribute`.

AWS CLI

Um das `launchPermission` Attribut zurückzusetzen

In diesem Beispiel wird das angegebene `launchPermission` Attribut AMI auf den Standardwert zurückgesetzt. Standardmäßig AMIs sind sie privat. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute LaunchPermission
```

- API-Einheiten finden Sie [ResetImageAttribute](#) in der AWS CLI Befehlsreferenz.

reset-instance-attribute

Das folgende Codebeispiel zeigt die Verwendung `reset-instance-attribute`.

AWS CLI

Um das `sourceDestCheck` Attribut zurückzusetzen

In diesem Beispiel wird das `sourceDestCheck` Attribut der angegebenen Instanz zurückgesetzt. Die Instanz muss sich in einer VPC befinden. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute sourceDestCheck
```

Um das `kernel`-Attribut zurückzusetzen

In diesem Beispiel wird das `kernel` Attribut der angegebenen Instanz zurückgesetzt. Die Instanz muss sich im Status `stopped` befinden. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute kernel
```

Um das `ramdisk`-Attribut zurückzusetzen

In diesem Beispiel wird das `ramdisk` Attribut der angegebenen Instanz zurückgesetzt. Die Instanz muss sich im Status `stopped` befinden. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute ramdisk
```

- API-Einheiten finden Sie [ResetInstanceAttribute](#) in der AWS CLI Befehlsreferenz.

reset-network-interface-attribute

Das folgende Codebeispiel zeigt die Verwendung `reset-network-interface-attribute`.

AWS CLI

Um ein Netzwerkschnittstellenattribut zurückzusetzen

Im folgenden `reset-network-interface-attribute` Beispiel wird der Wert des Attributs für die Quell-/Zielprüfung auf zurückgesetzt. `true`

```
aws ec2 reset-network-interface-attribute \  
--network-interface-id eni-686ea200 \  
--source-dest-check
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [ResetNetworkInterfaceAttribute](#) in AWS CLI der Befehlsreferenz.

reset-snapshot-attribute

Das folgende Codebeispiel zeigt die Verwendung `reset-snapshot-attribute`.

AWS CLI

Um ein Snapshot-Attribut zurückzusetzen

In diesem Beispiel werden die Berechtigungen zum Erstellen eines Volumes für den Snapshot `snap-1234567890abcdef0` zurückgesetzt. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --  
attribute createVolumePermission
```

- API-Einheiten finden Sie [ResetSnapshotAttribute](#) in der AWS CLI Befehlsreferenz.

restore-address-to-classic

Das folgende Codebeispiel zeigt die Verwendung `restore-address-to-classic`.

AWS CLI

Um eine Adresse auf EC2 -Classic wiederherzustellen

In diesem Beispiel wird die Elastic IP-Adresse 198.51.100.0 auf der -Classic-Plattform wiederhergestellt. EC2

Befehl:

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

Ausgabe:

```
{
  "Status": "MoveInProgress",
  "PublicIp": "198.51.100.0"
}
```

- API-Einheiten finden Sie unter Befehlsreferenz. [RestoreAddressToClassic](#) AWS CLI

restore-image-from-recycle-bin

Das folgende Codebeispiel zeigt die Verwendung `restore-image-from-recycle-bin`.

AWS CLI

Um ein Bild aus dem Papierkorb wiederherzustellen

Im folgenden `restore-image-from-recycle-bin` Beispiel wird AMI `ami-0111222333444abcd` aus dem Papierkorb wiederhergestellt.

```
aws ec2 restore-image-from-recycle-bin \
  --image-id ami-0111222333444abcd
```

Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie unter [AMIs Aus dem Papierkorb wiederherstellen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

- API-Einheiten finden Sie [RestoreImageFromRecycleBin](#) in der AWS CLI Befehlsreferenz.

restore-managed-prefix-list-version

Das folgende Codebeispiel zeigt die Verwendung `restore-managed-prefix-list-version`.

AWS CLI

US-WEST-2**Um eine Version der Präfixliste wiederherzustellen**

Im Folgenden werden die Einträge aus Version 1 der angegebenen Präfixliste `restore-managed-prefix-list-version` wiederhergestellt.

```
aws ec2 restore-managed-prefix-list-version \
  --prefix-list-id pl-0123456abcabc1 \
  --current-version 2 \
  --previous-version 1
```

Ausgabe:

```
{
  "PrefixList": {
    "PrefixListId": "pl-0123456abcabc1",
    "AddressFamily": "IPv4",
    "State": "restore-in-progress",
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabc1",
    "PrefixListName": "vpc-cidrs",
    "MaxEntries": 10,
    "Version": 2,
    "OwnerId": "123456789012"
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Verwaltete Präfixlisten](#) im VPCAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [RestoreManagedPrefixListVersion](#) in der AWS CLI Befehlsreferenz.

restore-snapshot-from-recycle-bin

Das folgende Codebeispiel zeigt die Verwendung `restore-snapshot-from-recycle-bin`.

AWS CLI

Um Schnappschüsse aus dem Papierkorb wiederherzustellen

Im folgenden `restore-snapshot-from-recycle-bin` Beispiel wird ein Snapshot aus dem Papierkorb wiederhergestellt. Wenn Sie einen Snapshot aus dem Papierkorb wiederherstellen, kann er sofort verwendet werden und wird aus dem Papierkorb entfernt. Sie können einen wiederhergestellten Snapshot genauso verwenden wie jeden anderen Snapshot in Ihrem Konto.

```
aws ec2 restore-snapshot-from-recycle-bin \  
  --snapshot-id snap-01234567890abcdef
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zum Papierkorb für Amazon EBS finden Sie unter [Schnappschüsse aus dem Papierkorb wiederherstellen](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [RestoreSnapshotFromRecycleBin](#) in AWS CLI der Befehlsreferenz.

restore-snapshot-tier

Das folgende Codebeispiel zeigt die Verwendung `restore-snapshot-tier`.

AWS CLI

Beispiel 1: Um einen archivierten Snapshot dauerhaft wiederherzustellen

Im folgenden `restore-snapshot-tier` Beispiel wird der angegebene Snapshot dauerhaft wiederhergestellt. Geben Sie die `permanent-restore` Option an `--snapshot-id` und schließen Sie sie ein.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

Ausgabe:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

Weitere Informationen zur Snapshot-Archivierung finden Sie unter EBS Amazon-Snapshots archivieren <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html> im Amazon-Benutzerhandbuch. EC2

Beispiel 2: Um einen archivierten Snapshot vorübergehend wiederherzustellen

Im folgenden `restore-snapshot-tier` Beispiel wird der angegebene Snapshot vorübergehend wiederhergestellt. Lassen Sie die `--permanent-restore`-Option weg. Geben Sie den `--snapshot-id` Wert und für die Anzahl der Tage `antemporary-restore-days`, für die der Snapshot wiederhergestellt werden soll. `temporary-restore-days` muss in Tagen angegeben werden. Der zulässige Bereich ist 1 bis 180. Wenn Sie keinen Wert angeben, wird standardmäßig 1 Tag verwendet.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 5
```

Ausgabe:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 5,  
  "IsPermanentRestore": false  
}
```

Weitere Informationen zur Snapshot-Archivierung finden Sie unter EBS Amazon-Snapshots archivieren <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html> im Amazon-Benutzerhandbuch. EC2

Beispiel 3: Um den Wiederherstellungszeitraum zu ändern

Im folgenden `restore-snapshot-tier` Beispiel wird der Wiederherstellungszeitraum für den angegebenen Snapshot auf 10 Tage geändert.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 10
```

Ausgabe:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 10,  
  "IsPermanentRestore": false  
}
```

Weitere Informationen zur Snapshot-Archivierung finden Sie unter EBS Amazon-Snapshots archivieren <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html> im Amazon-Benutzerhandbuch. EC2

Beispiel 4: Um den Wiederherstellungstyp zu ändern

Im folgenden `restore-snapshot-tier` Beispiel wird der Wiederherstellungstyp für den angegebenen Snapshot von temporär auf permanent geändert.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

Ausgabe:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

Weitere Informationen zur Snapshot-Archivierung finden Sie unter EBS Amazon-Snapshots archivieren <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html> im Amazon-Benutzerhandbuch. EC2

- API-Einheiten finden Sie in der Befehlsreferenz. [RestoreSnapshotTier](#) AWS CLI

revoke-client-vpn-ingress

Das folgende Codebeispiel zeigt die Verwendung `revoke-client-vpn-ingress`.

AWS CLI

Um eine Autorisierungsregel für einen VPN Client-Endpunkt zu widerrufen

Im folgenden `revoke-client-vpn-ingress` Beispiel wird eine Regel für den Internetzugang (`0.0.0.0/0`) für alle Gruppen aufgehoben.

```
aws ec2 revoke-client-vpn-ingress \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

Ausgabe:

```
{
  "Status": {
    "Code": "revoking"
  }
}
```

Weitere Informationen finden Sie unter [Autorisierungsregeln](#) im AWS Client VPN Administrator Guide.

- API-Einheiten finden Sie [RevokeClientVpnIngress](#) in der AWS CLI Befehlsreferenz.

revoke-security-group-egress

Das folgende Codebeispiel zeigt die Verwendung `revoke-security-group-egress`.

AWS CLI

Beispiel 1: Um die Regel zu entfernen, die ausgehenden Verkehr in einen bestimmten Adressbereich zulässt

Mit dem folgenden `revoke-security-group-egress` Beispielbefehl wird die Regel entfernt, die Zugriff auf die angegebenen Adressbereiche an TCP Port 80 gewährt.

```
aws ec2 revoke-security-group-egress \  
  --group-id sg-026c12253ce15eff7 \  
  --ip-  
permissions [[{IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Um die Regel zu entfernen, die ausgehenden Datenverkehr zu einer bestimmten Sicherheitsgruppe zulässt

Mit dem folgenden `revoke-security-group-egress` Beispielbefehl wird die Regel entfernt, die Zugriff auf die angegebene Sicherheitsgruppe an TCP Port 80 gewährt.

```
aws ec2 revoke-security-group-egress \  
  --group-id sg-026c12253ce15eff7 \  
  --ip-permissions '[[{"IpProtocol": "tcp", "FromPort": 443, "ToPort":  
443,"UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [RevokeSecurityGroupEgress](#) in der AWS CLI Befehlsreferenz.

revoke-security-group-ingress

Das folgende Codebeispiel zeigt die Verwendung `revoke-security-group-ingress`.

AWS CLI

Beispiel 1: Um eine Regel aus einer Sicherheitsgruppe zu entfernen

Im folgenden `revoke-security-group-ingress` Beispiel wird der Zugriff auf TCP Port 22 für den `203.0.113.0/24` Adressbereich standardmäßig aus der angegebenen Sicherheitsgruppe entfernt.

```
aws ec2 revoke-security-group-ingress \  
  --group-name mySecurityGroup \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

```
--cidr 203.0.113.0/24
```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Weitere Informationen finden Sie unter [Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Um eine Regel mithilfe des IP-Berechtigungssatzes zu entfernen

Im folgenden `revoke-security-group-ingress` Beispiel wird der `ip-permissions` Parameter verwendet, um eine eingehende Regel zu entfernen, die die ICMP Nachricht zulässt `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (Typ 3, Code 4).

```
aws ec2 revoke-security-group-ingress \  
  --group-id sg-026c12253ce15eff7 \  
  --ip-  
permissions IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]
```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Weitere Informationen finden Sie unter [Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [RevokeSecurityGroupIngress](#) in der AWS CLI Befehlsreferenz.

run-instances

Das folgende Codebeispiel zeigt die Verwendung `run-instances`.

AWS CLI

Beispiel 1: So starten Sie eine Instance in einem Standard-Subnetz

Im folgenden `run-instances` Beispiel wird eine einzelne Instance des Typs `t2.micro` im Standardsubnetz für die aktuelle Region gestartet und sie dem Standardsubnetz für das Standardsubnetz für die Region zugeordnet. VPC Das key pair ist optional, wenn Sie nicht vorhaben, über SSH (Linux) oder RDP (Windows) eine Verbindung zu Ihrer Instance herzustellen.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --key-name MyKeyPair
```

Ausgabe:

```
{
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0abcdef1234567890",
      "InstanceId": "i-1231231230abcdef0",
      "InstanceType": "t2.micro",
      "KeyName": "MyKeyPair",
      "LaunchTime": "2018-05-10T08:05:20.000Z",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-2a",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
      "PrivateIpAddress": "10.0.0.157",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-04a636d18e83cfac",
      "VpcId": "vpc-1234567890abcdef0",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "",
      "EbsOptimized": false,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2018-05-10T08:05:20.000Z",
            "AttachmentId": "eni-attach-0e325c07e928a0405",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching"
          }
        }
      ],
    }
  ]
}
```

```
    "Description": "",
    "Groups": [
      {
        "GroupName": "MySecurityGroup",
        "GroupId": "sg-0598c7d356eba48d7"
      }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "0a:ab:58:e0:67:e2",
    "NetworkInterfaceId": "eni-0c0a29997760baee7",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
    "PrivateIpAddress": "10.0.0.157",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
        "PrivateIpAddress": "10.0.0.157"
      }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-04a636d18e83cfacb",
    "VpcId": "vpc-1234567890abcdef0",
    "InterfaceType": "interface"
  }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
  {
    "GroupName": "MySecurityGroup",
    "GroupId": "sg-0598c7d356eba48d7"
  }
],
"SourceDestCheck": true,
"StateReason": {
  "Code": "pending",
  "Message": "pending"
},
"Tags": [],
"VirtualizationType": "hvm",
"CpuOptions": {
```

```

        "CoreCount": 1,
        "ThreadsPerCore": 1
    },
    "CapacityReservationSpecification": {
        "CapacityReservationPreference": "open"
    },
    "MetadataOptions": {
        "State": "pending",
        "HttpTokens": "optional",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
}
],
"OwnerId": "123456789012",
"ReservationId": "r-02a3f596d91211712"
}

```

Beispiel 2: So starten Sie eine Instance in einem nicht standardmäßigen Subnetz und fügen eine öffentliche IP-Adresse hinzu

Im folgenden `run-instances`-Beispiel wird eine öffentliche IP-Adresse für eine Instance angefordert, die Sie in einem nicht standardmäßigen Subnetz starten. Die Instance ist mit der angegebenen Sicherheitsgruppe verbunden.

```

aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --subnet-id subnet-08fc749671b2d077c \
  --security-group-ids sg-0b0384b66d7d692f9 \
  --associate-public-ip-address \
  --key-name MyKeyPair

```

Ein Beispiel für die Ausgabe von `run-instances` finden Sie in Beispiel 1.

Beispiel 3: So starten Sie eine Instance mit zusätzlichen Volumes

Das folgende `run-instances`-Beispiel verwendet eine Blockgerät-Zuweisung, die in `mapping.json` angegeben ist, um beim Start zusätzliche Volumes anzufügen. Bei der Zuordnung von Blockgeräten können EBS Volumes, Instance-Speicher-Volumes oder sowohl EBS Volumes als auch Instance-Speicher-Volumes angegeben werden.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --subnet-id subnet-08fc749671b2d077c \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --key-name MyKeyPair \  
  --block-device-mappings file://mapping.json
```

Inhalt von `mapping.json`. In diesem Beispiel wird `/dev/sdh` ein leeres EBS Volume mit einer Größe von 100 GiB hinzugefügt.

```
[  
  {  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
      "VolumeSize": 100  
    }  
  }  
]
```

Inhalt von `mapping.json`. In diesem Beispiel wird `ephemeral1` als Instance-Speicher-Volume hinzugefügt.

```
[  
  {  
    "DeviceName": "/dev/sdc",  
    "VirtualName": "ephemeral1"  
  }  
]
```

Ein Beispiel für die Ausgabe von `run-instances` finden Sie in Beispiel 1.

Weitere Informationen zu Blockgerätezuordnungen finden Sie unter [Blockgerätezuordnungen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 4: So starten Sie eine Instance und fügen bei der Erstellung Tags hinzu

Im folgenden `run-instances`-Beispiel wird ein Tag mit dem Schlüssel `webserver` und dem Wert `production` zur Instance hinzugefügt. Mit dem Befehl wird außerdem jedem erstellten Volume (in diesem Fall dem EBS Root-Volume) ein Tag mit dem Schlüssel `cost-center` und dem Wert von `cc123` zugewiesen.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --tag-specifications  
  'ResourceType=instance,Tags=[{Key=webserver,Value=production}]'  
  'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

Ein Beispiel für die Ausgabe von `run-instances` finden Sie in Beispiel 1.

Beispiel 5: So starten Sie eine Instance mit Benutzerdaten

Im folgenden `run-instances`-Beispiel werden Benutzerdaten in eine Datei mit dem Namen `my_script.txt` übergeben, die ein Konfigurationsskript für Ihre Instance enthält. Das Skript wird beim Start ausgeführt.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --user-data file://my_script.txt
```

Ein Beispiel für die Ausgabe von `run-instances` finden Sie in Beispiel 1.

Weitere Informationen zu Instance-Benutzerdaten finden Sie unter [Arbeiten mit Instance-Benutzerdaten](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 6: So starten Sie eine Burstable Performance Instance

Im folgenden `run-instances`-Beispiel wird eine `t2.micro`-Instance mit der `unlimited`-Kreditoption gestartet. Wenn Sie eine T2-Instance starten und keinen `--credit-specification` angeben, wird standardmäßig die Kreditoption `standard` verwendet. Wenn Sie eine T3-Instance starten, ist die Standardeinstellung die Kreditoption `unlimited`.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --credit-specification unlimited
```



```
--image-id ami-0abcdef1234567890 \  
--instance-type t2.micro \  
--count 1 \  
--subnet-id subnet-08fc749671b2d077c \  
--key-name MyKeyPair \  
--security-group-ids sg-0b0384b66d7d692f9 \  
--credit-specification CpuCredits=unlimited
```

Ein Beispiel für die Ausgabe von `run-instances` finden Sie in Beispiel 1.

Weitere Informationen zu Burstable Performance Instances finden Sie unter [Burstable Performance Instances](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [RunInstances AWS CLI](#) Befehlsreferenz.

run-scheduled-instances

Das folgende Codebeispiel zeigt die Verwendung von `run-scheduled-instances`.

AWS CLI

Um eine geplante Instance zu starten

In diesem Beispiel wird die angegebene Scheduled Instance in einem gestartetVPC.

Befehl:

```
aws ec2 run-scheduled-instances --scheduled-instance-  
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-  
specification file://launch-specification.json
```

launch-specification.json:

```
{  
  "ImageId": "ami-12345678",  
  "KeyName": "my-key-pair",  
  "InstanceType": "c4.large",  
  "NetworkInterfaces": [  
    {  
      "DeviceIndex": 0,  
      "SubnetId": "subnet-12345678",  
      "AssociatePublicIpAddress": true,  
      "Groups": ["sg-12345678"]  
    }  
  ]  
}
```

```
    }
  ],
  "IamInstanceProfile": {
    "Name": "my-iam-role"
  }
}
```

Ausgabe:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

In diesem Beispiel wird die angegebene Scheduled Instance in -Classic gestartet. EC2

Befehl:

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

launch-specification.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
  "IamInstanceProfile": {
    "Name": "my-iam-role"
  }
}
```

Ausgabe:

```
{
```

```
"InstanceIdSet": [  
  "i-1234567890abcdef0"  
]  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [RunScheduledInstances](#).AWS CLI

search-local-gateway-routes

Das folgende Codebeispiel zeigt die Verwendung `search-local-gateway-routes`.

AWS CLI

Um in einer Routentabelle eines lokalen Gateways nach Routen zu suchen

Im folgenden `search-local-gateway-routes` Beispiel wird in der angegebenen lokalen Gateway-Routentabelle nach statischen Routen gesucht.

```
aws ec2 search-local-gateway-routes \  
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \  
  --filters "Name=type,Values=static"
```

Ausgabe:

```
{  
  "Route": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",  
    "Type": "static",  
    "State": "deleted",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"  
  }  
}
```

- API-Einheiten finden Sie [SearchLocalGatewayRoutes](#) unter AWS CLI Befehlsreferenz.

search-transit-gateway-multicast-groups

Das folgende Codebeispiel zeigt die Verwendung `search-transit-gateway-multicast-groups`.

AWS CLI

Um eine oder mehrere Transit-Gateway-Multicast-Gruppen zu durchsuchen und die Informationen zur Gruppenmitgliedschaft zurückzugeben

Im folgenden `search-transit-gateway-multicast-groups` Beispiel wird die Gruppenmitgliedschaft der angegebenen Multicast-Gruppe zurückgegeben.

```
aws ec2 search-transit-gateway-multicast-groups \  
  --transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

Ausgabe:

```
{  
  "MulticastGroups": [  
    {  
      "GroupIpAddress": "224.0.1.0",  
      "TransitGatewayAttachmentId": "tgw-attach-0372e72386EXAMPLE",  
      "SubnetId": "subnet-0187aff814EXAMPLE",  
      "ResourceId": "vpc-0065acced4EXAMPLE",  
      "ResourceType": "vpc",  
      "NetworkInterfaceId": "eni-03847706f6EXAMPLE",  
      "GroupMember": false,  
      "GroupSource": true,  
      "SourceType": "static"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im Transit Gateways Guide unter [Managing Multicast-Gruppen](#).

- API-Einheiten finden Sie unter [SearchTransitGatewayMulticastGroups AWS CLI-Befehlsreferenz](#).

`search-transit-gateway-routes`

Das folgende Codebeispiel zeigt die Verwendung `search-transit-gateway-routes`.

AWS CLI

Um in der angegebenen Transit-Gateway-Routentabelle nach Routen zu suchen

Im folgenden `search-transit-gateway-routes` Beispiel werden alle Routen zurückgegeben, deren Typ `static` in der angegebenen Routentabelle enthalten ist.

```
aws ec2 search-transit-gateway-routes \
  --transit-gateway-route-table-id tgw-rtb-0a823eddbdeEXAMPLE \
  --filters "Name=type,Values=static"
```

Ausgabe:

```
{
  "Routes": [
    {
      "DestinationCidrBlock": "10.0.2.0/24",
      "TransitGatewayAttachments": [
        {
          "ResourceId": "vpc-4EXAMPLE",
          "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
          "ResourceType": "vpc"
        }
      ],
      "Type": "static",
      "State": "active"
    },
    {
      "DestinationCidrBlock": "10.1.0.0/24",
      "TransitGatewayAttachments": [
        {
          "ResourceId": "vpc-4EXAMPLE",
          "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
          "ResourceType": "vpc"
        }
      ],
      "Type": "static",
      "State": "active"
    }
  ],
  "AdditionalRoutesAvailable": false
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Routentabellen](#) im Transit Gateways Guide.

- API-Einheiten finden Sie unter [SearchTransitGatewayRoutes AWS CLI-Befehlsreferenz](#).

send-diagnostic-interrupt

Das folgende Codebeispiel zeigt die Verwendung send-diagnostic-interrupt.

AWS CLI

Um einen Diagnose-Interrupt zu senden

Im folgenden send-diagnostic-interrupt Beispiel wird ein Diagnose-Interrupt an die angegebene Instanz gesendet.

```
aws ec2 send-diagnostic-interrupt \  
  --instance-id i-1234567890abcdef0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [SendDiagnosticInterrupt AWS CLI](#) Befehlsreferenz.

start-instances

Das folgende Codebeispiel zeigt die Verwendung start-instances.

AWS CLI

Um eine EC2 Amazon-Instance zu starten

In diesem Beispiel wird die angegebene EBS Amazon-gestützte Instance gestartet.

Befehl:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{  
  "StartingInstances": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CurrentState": {  
        "Code": 0,  

```

```

        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter Ihre Instance anhalten und starten im Benutzerhandbuch zu Amazon Elastic Compute Cloud.

- API-Einheiten finden Sie [StartInstances](#) in der AWS CLI Befehlsreferenz.

start-network-insights-access-scope-analysis

Das folgende Codebeispiel zeigt die Verwendung `start-network-insights-access-scope-analysis`.

AWS CLI

Um eine Network Insights-Zugriffsumfangsanalyse zu starten

Im folgenden `start-network-insights-access-scope-analysis` Beispiel wird die Umfangsanalyse in Ihrem AWS Konto gestartet.

```

aws ec2 start-network-insights-access-scope-analysis \
  --region us-east-1 \
  --network-insights-access-scope-id nis-123456789111

```

Ausgabe:

```

{
  "NetworkInsightsAccessScopeAnalysis": {
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789222",
    "NetworkInsightsAccessScopeId": "nis-123456789111",
    "Status": "running",
    "StartDate": "2022-01-26T00:47:06.814000+00:00"
  }
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Network Access Analyzer using the AWS CLI](#) im Network Access Analyzer-Handbuch.

- API-Einheiten finden Sie [StartNetworkInsightsAccessScopeAnalysis](#) unter AWS CLI Befehlsreferenz.

start-network-insights-analysis

Das folgende Codebeispiel zeigt die Verwendung `start-network-insights-analysis`.

AWS CLI

Um einen Pfad zu analysieren

Im folgenden `start-network-insights-analysis` Beispiel wird der Pfad zwischen Quelle und Ziel analysiert. Verwenden Sie den `describe-network-insights-analyses` Befehl, um die Ergebnisse der Pfadanalyse anzuzeigen.

```
aws ec2 start-network-insights-analysis \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

Ausgabe:

```
{  
  "NetworkInsightsAnalysis": {  
    "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",  
    "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-  
insights-analysis/nia-02207aa13eb480c7a",  
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",  
    "StartDate": "2021-01-20T22:58:37.495Z",  
    "Status": "running"  
  }  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS CLI](#) im Reachability Analyzer Guide.

- API-Einheiten finden Sie [StartNetworkInsightsAnalysis](#) in der AWS CLI Befehlsreferenz.

start-vpc-endpoint-service-private-dns-verification

Das folgende Codebeispiel zeigt die Verwendung `start-vpc-endpoint-service-private-dns-verification`.

AWS CLI

Um den DNS Überprüfungsprozess einzuleiten

Das folgende `start-vpc-endpoint-service-private-dns-verification` Beispiel initiiert den DNS Überprüfungsprozess für den angegebenen Endpunktdienst.

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \  
  --service-id vpce-svc-071afff70666e61e0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS PrivateLink Benutzerhandbuch unter [DNSNamen verwalten](#).

- API-Einheiten finden Sie [StartVpcEndpointServicePrivateDnsVerification](#) in der AWS CLI Befehlsreferenz.

stop-instances

Das folgende Codebeispiel zeigt die Verwendung `stop-instances`.

AWS CLI

Beispiel 1: Um eine EC2 Amazon-Instance zu stoppen

Das folgende `stop-instances` Beispiel stoppt die angegebene EBS Amazon-gestützte Instance.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{  
  "StoppingInstances": [  
    {
```

```
    "InstanceId": "i-1234567890abcdef0",
    "CurrentState": {
      "Code": 64,
      "Name": "stopping"
    },
    "PreviousState": {
      "Code": 16,
      "Name": "running"
    }
  }
]
}
```

Weitere Informationen finden Sie unter [Ihre Instance anhalten und starten](#) im Benutzerhandbuch zu Amazon Elastic Compute Cloud.

Beispiel 2: Um eine Amazon-Instance in den Ruhezustand zu versetzen EC2

Im folgenden `stop-instances` Beispiel wird eine EBS Amazon-gestützte Instance in den Ruhezustand versetzt, wenn die Instance für den Ruhezustand aktiviert ist und die Voraussetzungen für den Ruhezustand erfüllt. Nachdem die Instance in den Ruhezustand versetzt wurde, wird die Instance angehalten.

```
aws ec2 stop-instances \
  --instance-ids i-1234567890abcdef0 \
  --hibernate
```

Ausgabe:

```
{
  "StoppingInstances": [
    {
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "InstanceId": "i-1234567890abcdef0",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Ihre On-Demand-Linux-Instance in den Ruhezustand versetzen](#) im Benutzerhandbuch zu Amazon Elastic Cloud Compute.

- Einzelheiten finden Sie unter Befehlsreferenz. API [StopInstances](#) AWS CLI

terminate-client-vpn-connections

Das folgende Codebeispiel zeigt die Verwendung `terminate-client-vpn-connections`.

AWS CLI

Um eine Verbindung zu einem VPN Client-Endpunkt zu beenden

Im folgenden `terminate-client-vpn-connections` Beispiel wird die angegebene Verbindung zum VPN Client-Endpunkt beendet.

```
aws ec2 terminate-client-vpn-connections \
  --client-vpn-endpoint-id vpn-endpoint-123456789123abcde \
  --connection-id cvpn-connection-04edd76f5201e0cb8
```

Ausgabe:

```
{
  "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",
  "ConnectionStatuses": [
    {
      "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",
      "PreviousStatus": {
        "Code": "active"
      },
      "CurrentStatus": {
        "Code": "terminating"
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Client-Verbindungen](#) im AWS VPNClient-Administratorhandbuch.

- API-Einheiten finden Sie [TerminateClientVpnConnections](#) in der AWS CLI Befehlsreferenz.

terminate-instances

Das folgende Codebeispiel zeigt die Verwendung `terminate-instances`.

AWS CLI

Um eine EC2 Amazon-Instance zu beenden

In diesem Beispiel wird die angegebene Instance beendet.

Befehl:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Using Amazon EC2 Instances](#) im AWS Command Line Interface User Guide.

- API-Einheiten finden Sie [TerminateInstances](#) unter AWS CLI Befehlsreferenz.

unassign-ipv6-addresses

Das folgende Codebeispiel zeigt die Verwendung `unassign-ipv6-addresses`.

AWS CLI

Um die Zuweisung einer IPv6 Adresse zu einer Netzwerkschnittstelle aufzuheben

In diesem Beispiel wird die Zuweisung der angegebenen IPv6 Adresse zur angegebenen Netzwerkschnittstelle aufgehoben.

Befehl:

```
aws ec2 unassign-ipv6-addresses --ipv6-  
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

Ausgabe:

```
{  
  "NetworkInterfaceId": "eni-23c49b68",  
  "UnassignedIpv6Addresses": [  
    "2001:db8:1234:1a00:3304:8879:34cf:4071"  
  ]  
}
```

- API-Einheiten finden Sie unter [UnassignIpv6Addresses](#) in AWS CLI der Befehlsreferenz.

unassign-private-ip-addresses

Das folgende Codebeispiel zeigt die Verwendung `unassign-private-ip-addresses`.

AWS CLI

Um die Zuweisung einer sekundären privaten IP-Adresse zu einer Netzwerkschnittstelle aufzuheben

In diesem Beispiel wird die Zuweisung der angegebenen privaten IP-Adresse zur angegebenen Netzwerkschnittstelle aufgehoben. Wird der Befehl erfolgreich ausgeführt, wird keine Ausgabe zurückgegeben.

Befehl:

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-  
ip-addresses 10.0.0.82
```

- API-Einheiten finden Sie unter [UnassignPrivateIpAddresses](#) [AWS CLI](#) Befehlsreferenz.

unassign-private-nat-gateway-address

Das folgende Codebeispiel zeigt die Verwendung `unassign-private-nat-gateway-address`.

AWS CLI

Um die Zuweisung einer privaten IP-Adresse zu Ihrem privaten NAT Gateway aufzuheben

Im folgenden `unassign-private-nat-gateway-address` Beispiel wird die Zuweisung der angegebenen IP-Adresse zum angegebenen privaten Gateway aufgehoben. NAT

```
aws ec2 unassign-private-nat-gateway-address \
  --nat-gateway-id nat-1234567890abcdef0 \
  --private-ip-addresses 10.0.20.197
```

Ausgabe:

```
{
  "NatGatewayId": "nat-0ee3edd182361f662",
  "NatGatewayAddresses": [
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "PrivateIp": "10.0.20.197",
      "IsPrimary": false,
      "Status": "unassigning"
    }
  ]
}
```

Weitere Informationen finden Sie unter [NATGateways](#) im VPCAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UnassignPrivateNatGatewayAddress](#) in der AWS CLI Befehlsreferenz.

unlock-snapshot

Das folgende Codebeispiel zeigt die Verwendung `unlock-snapshot`.

AWS CLI

Um einen Snapshot zu entsperren

Im folgenden `unlock-snapshot` Beispiel wird der angegebene Snapshot entsperrt.

```
aws ec2 unlock-snapshot \  
  --snapshot-id snap-0b5e733b4a8df6e0d
```

Ausgabe:

```
{  
  "SnapshotId": "snap-0b5e733b4a8df6e0d"  
}
```

Weitere Informationen finden Sie unter [Snapshot Lock](#) im EBSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [UnlockSnapshot](#) in der AWS CLI Befehlsreferenz.

unmonitor-instances

Das folgende Codebeispiel zeigt die Verwendung `unmonitor-instances`.

AWS CLI

So deaktivieren Sie die detaillierte Überwachung für eine Instance

Dieser Beispielbefehl deaktiviert die detaillierte Überwachung für die angegebene Instance.

Befehl:

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

Ausgabe:

```
{  
  "InstanceMonitorings": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "Monitoring": {  
        "State": "disabling"  
      }  
    }  
  ]  
}
```

- API-Einheiten finden Sie [UnmonitorInstances](#) in der AWS CLI Befehlsreferenz.

update-security-group-rule-descriptions-egress

Das folgende Codebeispiel zeigt die Verwendung `update-security-group-rule-descriptions-egress`.

AWS CLI

Um die Beschreibung einer Sicherheitsgruppenregel für ausgehenden Datenverkehr zu aktualisieren

Im folgenden `update-security-group-rule-descriptions-egress` Beispiel wird die Beschreibung der Sicherheitsgruppenregel für den angegebenen Port und IPv4 Adressbereich aktualisiert. Die Beschreibung `'Outbound HTTP access to server 2'` ersetzt alle vorhandenen Beschreibungen für die Regel.

```
aws ec2 update-security-group-rule-descriptions-egress \
  --group-id sg-02f0d35a850ba727f \
  --ip-permissions
  IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=203.0.113.0/24,Description="Outbound
  HTTP access to server 2"}]
```

Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie unter [Regeln für Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSecurityGroupRuleDescriptionsEgress](#) in der AWS CLI Befehlsreferenz.

update-security-group-rule-descriptions-ingress

Das folgende Codebeispiel zeigt die Verwendung `update-security-group-rule-descriptions-ingress`.

AWS CLI

Beispiel 1: Um die Beschreibung einer Sicherheitsgruppenregel für eingehenden Datenverkehr mit einer CIDR Quelle zu aktualisieren

Im folgenden `update-security-group-rule-descriptions-ingress` Beispiel wird die Beschreibung der Sicherheitsgruppenregel für den angegebenen Port und IPv4 Adressbereich aktualisiert. Die Beschreibung 'SSH access from ABC office' ersetzt alle vorhandenen Beschreibungen für die Regel.

```
aws ec2 update-security-group-rule-descriptions-ingress \  
  --group-id sg-02f0d35a850ba727f \  
  --ip-permissions  
  IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='[{"CidrIp=203.0.113.0/16,Description="SSH  
access from corpnet"}]'
```

Ausgabe:

```
{  
  "Return": true  
}
```

Weitere Informationen finden Sie unter [Regeln für Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

Beispiel 2: Um die Beschreibung einer Sicherheitsgruppenregel für eingehenden Datenverkehr mit einer Präfixlistenquelle zu aktualisieren

Im folgenden `update-security-group-rule-descriptions-ingress` Beispiel wird die Beschreibung der Sicherheitsgruppenregel für die angegebene Port- und Präfixliste aktualisiert. Die Beschreibung 'SSH access from ABC office' ersetzt alle vorhandenen Beschreibungen für die Regel.

```
aws ec2 update-security-group-rule-descriptions-ingress \  
  --group-id sg-02f0d35a850ba727f \  
  --ip-permissions  
  IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds='[{"PrefixListId=pl-12345678,Description=  
access from corpnet"}]'
```

Ausgabe:

```
{
  "Return": true
}
```

Weitere Informationen finden Sie unter [Regeln für Sicherheitsgruppen](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [UpdateSecurityGroupRuleDescriptionsIngress](#) in der AWS CLI Befehlsreferenz.

withdraw-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `withdraw-byoip-cidr`.

AWS CLI

Um die Werbung für einen Adressbereich zu beenden

Im folgenden `withdraw-byoip-cidr` Beispiel wird die Werbung für den angegebenen Adressbereich beendet.

```
aws ec2 withdraw-byoip-cidr
  --cidr 203.0.113.25/24
```

Ausgabe:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    "State": "advertised"
  }
}
```

- APIEinzelheiten finden Sie [WithdrawByoipCidr](#) in der AWS CLI Befehlsreferenz.

Amazon EC2 Instance Connect-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon EC2 Instance Connect Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

send-ssh-public-key

Das folgende Codebeispiel zeigt die Verwendungsend-ssh-public-key.

AWS CLI

Um einen SSH öffentlichen Schlüssel an eine Instanz zu senden

Das folgende send-ssh-public-key Beispiel sendet den angegebenen SSH öffentlichen Schlüssel an die angegebene Instanz. Der Schlüssel wird verwendet, um den angegebenen Benutzer zu authentifizieren.

```
aws ec2-instance-connect send-ssh-public-key \  
  --instance-id i-1234567890abcdef0 \  
  --instance-os-user ec2-user \  
  --availability-zone us-east-2b \  
  --ssh-public-key file://path/my-rsa-key.pub
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- APIEinzelheiten finden Sie [SendSshPublicKey](#) in der AWS CLI Befehlsreferenz.

ECRAmazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementierenECR.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-check-layer-availability

Das folgende Codebeispiel zeigt die Verwendung `batch-check-layer-availability`.

AWS CLI

Um die Verfügbarkeit eines Layers zu überprüfen

Im folgenden `batch-check-layer-availability` Beispiel wird die Verfügbarkeit eines Layers mit dem Digest

`sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed` im `cluster-autoscaler` Repository überprüft.

```
aws ecr batch-check-layer-availability \
  --repository-name cluster-autoscaler \
  --layer-
digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

Ausgabe:

```
{
  "layers": [
    {
      "layerDigest":
"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",
      "layerAvailability": "AVAILABLE",
      "layerSize": 2777,
      "mediaType": "application/vnd.docker.container.image.v1+json"
    }
  ]
}
```

```
  ],  
  "failures": []  
}
```

- API-Einheiten finden Sie [BatchCheckLayerAvailability](#) in der AWS CLI Befehlsreferenz.

batch-delete-image

Das folgende Codebeispiel zeigt die Verwendung `batch-delete-image`.

AWS CLI

Beispiel 1: Um ein Bild zu löschen

Im folgenden `batch-delete-image` Beispiel wird ein Bild mit dem Tag `precise` im angegebenen Repository in der Standardregistrierung für ein Konto gelöscht.

```
aws ecr batch-delete-image \  
  --repository-name ubuntu \  
  --image-ids imageTag=precise
```

Ausgabe:

```
{  
  "failures": [],  
  "imageIds": [  
    {  
      "imageTag": "precise",  
      "imageDigest":  
"sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"  
    }  
  ]  
}
```

Beispiel 2: Um mehrere Bilder zu löschen

Im folgenden `batch-delete-image` Beispiel werden alle Bilder gelöscht, die mit `prod` und `team1` im angegebenen Repository markiert sind.

```
aws ecr batch-delete-image \  
  --repository-name MyRepository \  
  --image-ids prod,team1
```

```
--image-ids imageTag=prod imageTag=team1
```

Ausgabe:

```
{
  "imageIds": [
    {
      "imageDigest": "sha256:123456789012",
      "imageTag": "prod"
    },
    {
      "imageDigest": "sha256:567890121234",
      "imageTag": "team1"
    }
  ],
  "failures": []
}
```

Weitere Informationen finden Sie unter [Löschen eines Bilds](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchDeleteImage](#) in der AWS CLI Befehlsreferenz.

batch-get-image

Das folgende Codebeispiel zeigt die Verwendung `batch-get-image`.

AWS CLI

Beispiel 1: Um ein Bild zu erhalten

Im folgenden `batch-get-image` Beispiel wird ein Bild mit dem Tag `v1.13.6` in einem Repository abgerufen, das `cluster-autoscaler` in der Standardregistrierung für ein Konto aufgerufen wird.

```
aws ecr batch-get-image \
  --repository-name cluster-autoscaler \
  --image-ids imageTag=v1.13.6
```

Ausgabe:

```
{
```

```

    "images": [
      {
        "registryId": "012345678910",
        "repositoryName": "cluster-autoscaler",
        "imageId": {
          "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
          "imageTag": "v1.13.6"
        },
        "imageManifest": "{\n  \"schemaVersion\": 2,\n
  \"mediaType\": \"application/vnd.docker.distribution.manifest.v2+json
  \",\n  \"config\": {\n    \"mediaType\": \"application/
  vnd.docker.container.image.v1+json\", \n    \"size\": 2777, \n    \"digest
  \": \"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
  \"\n  }, \n  \"layers\": [\n    {\n      \"mediaType
  \": \"application/vnd.docker.image.rootfs.diff.tar.gzip
  \", \n      \"size\": 17743696, \n      \"digest\":
  \"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffef5a85f900caaf6e1\"\n
    }, \n    {\n      \"mediaType\": \"application/
  vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 2565026, \n
    \"digest\":
  \"sha256:8c8a779d3a537b767ae1091fe6e00c2590afd16767aa6096d1b318d75494819f
  \"\n    }, \n    {\n      \"mediaType\": \"application/
  vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 28005981, \n
    \"digest\":
  \"sha256:c44ba47496991c9982ee493b47fd25c252caabf2b4ae7dd679c9a27b6a3c8fb7\"\n
    }, \n    {\n      \"mediaType\": \"application/
  vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 775, \n      \"digest
  \": \"sha256:e2c388b44226544363ca007be7b896bcce1baebea04da23cbd165eac30be650f\"\n
    } \n  ] \n}"
      },
      {
        "failures": []
      }
    ]
  }

```

Beispiel 2: Um mehrere Bilder zu erhalten

Im folgenden `batch-get-image` Beispiel werden Details zu allen Bildern angezeigt, die mit `prod` und `team1` im angegebenen Repository markiert sind.

```

aws ecr batch-get-image \
  --repository-name MyRepository \
  --image-ids imageTag=prod imageTag=team1

```

Ausgabe:

```
{
  "images": [
    {
      "registryId": "123456789012",
      "repositoryName": "MyRepository",
      "imageId": {
        "imageDigest": "sha256:123456789012",
        "imageTag": "prod"
      },
      "imageManifest": "manifestExample1"
    },
    {
      "registryId": "567890121234",
      "repositoryName": "MyRepository",
      "imageId": {
        "imageDigest": "sha256:123456789012",
        "imageTag": "team1"
      },
      "imageManifest": "manifestExample2"
    }
  ],
  "failures": []
}
```

Weitere Informationen finden Sie unter [Bilder](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetImage](#) in der AWS CLI Befehlsreferenz.

complete-layer-upload

Das folgende Codebeispiel zeigt die Verwendung `complete-layer-upload`.

AWS CLI

Um den Upload einer Bildebene abzuschließen

Im folgenden `complete-layer-upload` Beispiel wird der Upload einer Bildebene in das `layer-test` Repository abgeschlossen.

```
aws ecr complete-layer-upload \
  --repository-name layer-test \
```



```
--upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
--layer-digests 6cb64b8a-9378-0e33-2ab1-
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

Ausgabe:

```
{
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
  "layerDigest":
    "sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",
  "repositoryName": "layer-test",
  "registryId": "130757420319"
}
```

- API-Einheiten finden Sie [CompleteLayerUpload](#) in der AWS CLI Befehlsreferenz.

create-repository

Das folgende Codebeispiel zeigt die Verwendung `create-repository`.

AWS CLI

Beispiel 1: Um ein Repository zu erstellen

Im folgenden `create-repository` Beispiel wird ein Repository innerhalb des angegebenen Namespace in der Standardregistrierung für ein Konto erstellt.

```
aws ecr create-repository \
  --repository-name project-a/sample-repo
```

Ausgabe:

```
{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "project-a/sample-repo",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/sample-repo"
  }
}
```

Weitere Informationen finden Sie unter [Creating a Repository](#) im ECR Amazon-Benutzerhandbuch.

Beispiel 2: Um ein Repository zu erstellen, das mit der Unveränderlichkeit des Image-Tags konfiguriert ist

Im folgenden `create-repository` Beispiel wird in der Standardregistrierung für ein Konto ein Repository erstellt, das für die Unveränderlichkeit von Tags konfiguriert ist.

```
aws ecr create-repository \  
  --repository-name project-a/sample-repo \  
  --image-tag-mutability IMMUTABLE
```

Ausgabe:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "project-a/sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/  
sample-repo",  
    "imageTagMutability": "IMMUTABLE"  
  }  
}
```

Weitere Informationen finden Sie unter [Image Tag Mutability](#) im ECR Amazon-Benutzerhandbuch.

Beispiel 3: So erstellen Sie ein Repository, das mit einer Scan-Konfiguration konfiguriert ist

Im folgenden `create-repository` Beispiel wird ein Repository erstellt, das so konfiguriert ist, dass es beim Image-Push in der Standardregistrierung für ein Konto einen Schwachstellenscan durchführt.

```
aws ecr create-repository \  
  --repository-name project-a/sample-repo \  
  --image-scanning-configuration scanOnPush=true
```

Ausgabe:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "project-a/sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/  
sample-repo",  
    "imageScanningConfiguration": {  
      "scanOnPush": true  
    }  
  }  
}
```

```
    "imageScanningConfiguration": {
      "scanOnPush": true
    }
  }
}
```

Weitere Informationen finden Sie unter [Scannen von Bildern](#) im ECRAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CreateRepository](#) in der AWS CLI Befehlsreferenz.

delete-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-lifecycle-policy`.

AWS CLI

Um die Lebenszyklusrichtlinie für ein Repository zu löschen

Im folgenden `delete-lifecycle-policy` Beispiel wird die Lebenszyklusrichtlinie für das `hello-world` Repository gelöscht.

```
aws ecr delete-lifecycle-policy \
  --repository-name hello-world
```

Ausgabe:

```
{
  "registryId": "012345678910",
  "repositoryName": "hello-world",
  "lifecyclePolicyText": "{\"rules\": [{\"rulePriority\": 1, \"description\": \"Remove untagged images.\", \"selection\": {\"tagStatus\": \"untagged\", \"countType\": \"sinceImagePushed\", \"countUnit\": \"days\", \"countNumber\": 10}, \"action\": {\"type\": \"expire\"}}]}",
  "lastEvaluatedAt": 0.0
}
```

- APIEinzelheiten finden Sie [DeleteLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

delete-repository-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-repository-policy`.

AWS CLI

Um die Repository-Richtlinie für ein Repository zu löschen

Im folgenden `delete-repository-policy` Beispiel wird die Repository-Richtlinie für das `cluster-autoscaler` Repository gelöscht.

```
aws ecr delete-repository-policy \  
  --repository-name cluster-autoscaler
```

Ausgabe:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "cluster-autoscaler",  
  "policyText": "{\n  \"Version\" : \"2008-10-17\",  
  \"Statement\" : [ {\n    \"Sid\" : \"allow public pull\",  
    \"Effect\" : \"Allow\",  
    \"Principal\" :  
    \"*\",  
    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage  
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
```

- API-Einheiten finden Sie [DeleteRepositoryPolicy](#) in der AWS CLI Befehlsreferenz.

`delete-repository`

Das folgende Codebeispiel zeigt die Verwendung `delete-repository`.

AWS CLI

So löschen Sie ein Repository

Das folgende `delete-repository` Beispiel mit `Command Force` löscht das angegebene Repository in der Standardregistrierung für ein Konto. Das `--force` Flag ist erforderlich, wenn das Repository Bilder enthält.

```
aws ecr delete-repository \  
  --repository-name ubuntu \  
  --force
```

Ausgabe:

```
{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "ubuntu",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines Repositorys](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteRepository](#) in der AWS CLI Befehlsreferenz.

describe-image-scan-findings

Das folgende Codebeispiel zeigt die Verwendung `describe-image-scan-findings`.

AWS CLI

Um die Scanergebnisse für ein Bild zu beschreiben

Im folgenden `describe-image-scan-findings` Beispiel werden die Ergebnisse des Bildscans für ein Bild mithilfe des Image Digest im angegebenen Repository in der Standardregistrierung für ein Konto zurückgegeben.

```
aws ecr describe-image-scan-findings \
  --repository-name sample-repo \
  --image-id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

Ausgabe:

```
{
  "imageScanFindings": {
    "findings": [
      {
        "name": "CVE-2019-5188",
        "description": "A code execution vulnerability exists in the directory rehashing functionality of E2fsprogs e2fsck 1.45.4. A specially crafted ext4 directory can cause an out-of-bounds write on the stack, resulting in code execution. An attacker can corrupt a partition to trigger this vulnerability.",
```

```
"uri": "http://people.ubuntu.com/~ubuntu-security/cve/CVE-2019-5188",
"severity": "MEDIUM",
"attributes": [
  {
    "key": "package_version",
    "value": "1.44.1-1ubuntu1.1"
  },
  {
    "key": "package_name",
    "value": "e2fsprogs"
  },
  {
    "key": "CVSS2_VECTOR",
    "value": "AV:L/AC:L/Au:N/C:P/I:P/A:P"
  },
  {
    "key": "CVSS2_SCORE",
    "value": "4.6"
  }
]
},
"imageScanCompletedAt": 1579839105.0,
"vulnerabilitySourceUpdatedAt": 1579811117.0,
"findingSeverityCounts": {
  "MEDIUM": 1
}
},
"registryId": "123456789012",
"repositoryName": "sample-repo",
"imageId": {
  "imageDigest":
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
},
"imageScanStatus": {
  "status": "COMPLETE",
  "description": "The scan was completed successfully."
}
}
```

Weitere Informationen finden Sie unter [Scannen von Bildern](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeImageScanFindings](#) in der AWS CLI Befehlsreferenz.

describe-images

Das folgende Codebeispiel zeigt die Verwendung `describe-images`.

AWS CLI

Um ein Bild in einem Repository zu beschreiben

Im folgenden `describe-images` Beispiel werden Details zu einem Bild im `cluster-autoscaler` Repository mit dem Tag `v1.13.6` angezeigt.

```
aws ecr describe-images \  
  --repository-name cluster-autoscaler \  
  --image-ids imageTag=v1.13.6
```

Ausgabe:

```
{  
  "imageDetails": [  
    {  
      "registryId": "012345678910",  
      "repositoryName": "cluster-autoscaler",  
      "imageDigest":  
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",  
      "imageTags": [  
        "v1.13.6"  
      ],  
      "imageSizeInBytes": 48318255,  
      "imagePushedAt": 1565128275.0  
    }  
  ]  
}
```

- API-Einheiten finden Sie [Beschreiben von Bildern](#) in der AWS CLI Befehlsreferenz.

describe-repositories

Das folgende Codebeispiel zeigt die Verwendung `describe-repositories`.

AWS CLI

Um die Repositories in einer Registrierung zu beschreiben

In diesem Beispiel werden die Repositorys in der Standardregistrierung für ein Konto beschrieben.

Befehl:

```
aws ecr describe-repositories
```

Ausgabe:

```
{
  "repositories": [
    {
      "registryId": "012345678910",
      "repositoryName": "ubuntu",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/ubuntu"
    },
    {
      "registryId": "012345678910",
      "repositoryName": "test",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/test"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeRepositories](#) in der AWS CLI Befehlsreferenz.

get-authorization-token

Das folgende Codebeispiel zeigt die Verwendung `get-authorization-token`.

AWS CLI

Um ein Autorisierungstoken für Ihre Standardregistrierung zu erhalten

Mit dem folgenden `get-authorization-token` Beispielbefehl wird ein Autorisierungstoken für Ihre Standardregistrierung abgerufen.

```
aws ecr get-authorization-token
```

Ausgabe:

```
{
```



```

    "authorizationData": [
      {
        "authorizationToken": "QVdT0kN...",
        "expiresAt": 1448875853.241,
        "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
      }
    ]
  }
}

```

- API-Einheiten finden Sie [GetAuthorizationToken](#) in der AWS CLI Befehlsreferenz.

get-download-url-for-layer

Das folgende Codebeispiel zeigt die Verwendung `get-download-url-for-layer`.

AWS CLI

Um einen Layer herunterzuladen URL

Das folgende `get-download-url-for-layer` Beispiel zeigt den Download URL eines Layers mit dem Digest

`sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed`
im `cluster-autoscaler` Repository.

```

aws ecr get-download-url-for-layer \
  --repository-name cluster-autoscaler \
  --layer-
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed

```

Ausgabe:

```

{
  "downloadUrl": "https://prod-us-west-2-starport-layer-bucket.s3.us-
west-2.amazonaws.com/e501-012345678910-9cb60dc0-7284-5643-3987-
da6dac0465f0/04620aac-66a5-4167-8232-55ee7ef6d565?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20190814T220617Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-
Amz-Credential=AKIA32P3D2JDNMVAJLGF%2F20190814%2Fus-west-2%2Fs3%2Faws4_request&X-
Amz-Signature=9161345894947a1672467a0da7a1550f2f7157318312fe4941b59976239c3337",
  "layerDigest":
  "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed"
}

```

- API-Einheiten finden Sie [GetDownloadUrlForLayer](#) in der AWS CLI Befehlsreferenz.

get-lifecycle-policy-preview

Das folgende Codebeispiel zeigt die Verwendung `get-lifecycle-policy-preview`.

AWS CLI

Um Details für eine Lifecycle-Policy-Vorschau abzurufen

Im folgenden `get-lifecycle-policy-preview` Beispiel wird das Ergebnis einer Lifecycle-Policy-Vorschau für das angegebene Repository in der Standardregistrierung für ein Konto abgerufen.

Befehl:

```
aws ecr get-lifecycle-policy-preview \
  --repository-name "project-a/amazon-ecs-sample"
```

Ausgabe:

```
{
  "registryId": "012345678910",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\": 1,\n      \"description\": \"Expire images older than 14 days\",\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",\n        \"countNumber\": 14\n      },\n      \"action\": {\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}",
  "status": "COMPLETE",
  "previewResults": [],
  "summary": {
    "expiringImageTotalCount": 0
  }
}
```

Weitere Informationen finden Sie unter [Lebenszyklusrichtlinien](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetLifecyclePolicyPreview](#) in der AWS CLI Befehlsreferenz.

get-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `get-lifecycle-policy`.

AWS CLI

Um eine Lebenszyklusrichtlinie abzurufen

Im folgenden `get-lifecycle-policy` Beispiel werden Details der Lebenszyklusrichtlinie für das angegebene Repository in der Standardregistrierung für das Konto angezeigt.

```
aws ecr get-lifecycle-policy \
  --repository-name "project-a/amazon-ecs-sample"
```

Ausgabe:

```
{
  "registryId": "123456789012",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\"rules\": [{\"rulePriority\": 1, \"description\": \"Expire images older than 14 days\", \"selection\": {\"tagStatus\": \"untagged\", \"countType\": \"sinceImagePushed\", \"countUnit\": \"days\", \"countNumber\": 14}, \"action\": {\"type\": \"expire\"}}]}",
  "lastEvaluatedAt": 1504295007.0
}
```

Weitere Informationen finden Sie unter [Lebenszyklusrichtlinien](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

get-login-password

Das folgende Codebeispiel zeigt die Verwendung `get-login-password`.

AWS CLI

Um ein Passwort zur Authentifizierung bei einer Registrierung abzurufen

Im Folgenden `get-login-password` wird ein Passwort angezeigt, das Sie mit einem Container-Client Ihrer Wahl verwenden können, um sich bei jeder ECR Amazon-Registrierung zu authentifizieren, auf die Ihr IAM Principal Zugriff hat.

```
aws ecr get-login-password
```

Ausgabe:

```
<password>
```

Um es mit dem Docker zu verwendenCLI, leiten Sie die Ausgabe des Befehls über die Pipeline an den `get-login-password docker login` Befehl weiter. Achten Sie beim Abrufen des Passworts darauf, dass Sie dieselbe Region angeben, in der sich Ihre ECR Amazon-Registrierung befindet.

```
aws ecr get-login-password \  
  --region <region> \  
 \  
| docker login \  
  --username AWS \  
  --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Weitere Informationen finden Sie unter [Registry Authentication](#) im ECR Amazon-Benutzerhandbuch.

- API Einheiten finden Sie [GetLoginPassword](#) in der AWS CLI Befehlsreferenz.

get-login

Das folgende Codebeispiel zeigt die Verwendung `get-login`.

AWS CLI

Um einen Docker-Login-Befehl für Ihre Standardregistrierung abzurufen

In diesem Beispiel wird ein Befehl gedruckt, mit dem Sie sich bei Ihrer Amazon-Standardregistrierung anmelden ECR können.

Befehl:

```
aws ecr get-login
```

Ausgabe:

```
docker login -u AWS -p <password> -e none https://  
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Um sich bei der Registrierung eines anderen Kontos anzumelden

In diesem Beispiel werden ein oder mehrere Befehle gedruckt, mit denen Sie sich Amazon ECR Amazon-Registern anmelden können, die mit anderen Konten verknüpft sind.

Befehl:

```
aws ecr get-login --registry-ids 012345678910 023456789012
```

Ausgabe:

```
docker login -u <username> -p <token-1> -e none <endpoint-1>  
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- API-Einheiten finden Sie unter [GetLogin AWS CLI](#) Befehlsreferenz.

get-repository-policy

Das folgende Codebeispiel zeigt die Verwendung `get-repository-policy`.

AWS CLI

Um die Repository-Richtlinie für ein Repository abzurufen

Im folgenden `get-repository-policy` Beispiel werden Details zur Repository-Richtlinie für das `cluster-autoscaler` Repository angezeigt.

```
aws ecr get-repository-policy \  
  --repository-name cluster-autoscaler
```

Ausgabe:

```
{
```

```

    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n\n    \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :\n    \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\n    \", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
  }

```

- API-Einheiten finden Sie [GetRepositoryPolicy](#) unter AWS CLI Befehlsreferenz.

initiate-layer-upload

Das folgende Codebeispiel zeigt die Verwendung `initiate-layer-upload`.

AWS CLI

Um einen Upload einer Bildebene zu initiieren

Im folgenden `initiate-layer-upload` Beispiel wird ein Upload einer Bildebene in das `layer-test` Repository initiiert.

```

aws ecr initiate-layer-upload \
  --repository-name layer-test

```

Ausgabe:

```

{
  "partSize": 10485760,
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}

```

- API-Einheiten finden Sie [InitiateLayerUpload](#) in der AWS CLI Befehlsreferenz.

list-images

Das folgende Codebeispiel zeigt die Verwendung `list-images`.

AWS CLI

Um die Bilder in einem Repository aufzulisten

Im folgenden `list-images` Beispiel wird eine Liste der Bilder im `cluster-autoscaler` Repository angezeigt.

```
aws ecr list-images \  
  --repository-name cluster-autoscaler
```

Ausgabe:

```
{  
  "imageIds": [  
    {  
      "imageDigest":  
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",  
      "imageTag": "v1.13.8"  
    },  
    {  
      "imageDigest":  
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",  
      "imageTag": "v1.13.7"  
    },  
    {  
      "imageDigest":  
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",  
      "imageTag": "v1.13.6"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListImages](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für das Repository aufzulisten

Im folgenden `list-tags-for-resource` Beispiel wird eine Liste der mit dem `hello-world` Repository verknüpften Tags angezeigt.

```
aws ecr list-tags-for-resource \  
  --repository-name hello-world
```

```
--resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world
```

Ausgabe:

```
{
  "tags": [
    {
      "Key": "Stage",
      "Value": "Integ"
    }
  ]
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-image-scanning-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-image-scanning-configuration`.

AWS CLI

Um die Konfiguration für das Scannen von Bildern für ein Repository zu aktualisieren

Im folgenden `put-image-scanning-configuration` Beispiel wird die Konfiguration für das Scannen von Bildern für das angegebene Repository aktualisiert.

```
aws ecr put-image-scanning-configuration \
  --repository-name sample-repo \
  --image-scanning-configuration scanOnPush=true
```

Ausgabe:

```
{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageScanningConfiguration": {
    "scanOnPush": true
  }
}
```

Weitere Informationen finden Sie unter [Scannen von Bildern](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [PutImageScanningConfiguration](#) in der AWS CLI Befehlsreferenz.

put-image-tag-mutability

Das folgende Codebeispiel zeigt die Verwendung `put-image-tag-mutability`.

AWS CLI

Um die Veränderbarkeitseinstellung für Image-Tags für ein Repository zu aktualisieren

Im folgenden `put-image-tag-mutability` Beispiel wird das angegebene Repository für die Unveränderlichkeit von Tags konfiguriert. Dadurch wird verhindert, dass alle Image-Tags innerhalb des Repositories überschrieben werden.

```
aws ecr put-image-tag-mutability \  
  --repository-name hello-repository \  
  --image-tag-mutability IMMUTABLE
```

Ausgabe:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",  
  "imageTagMutability": "IMMUTABLE"  
}
```

Weitere Informationen finden Sie unter [Image Tag Mutability](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [PutImageTagMutability](#) in der AWS CLI Befehlsreferenz.

put-image

Das folgende Codebeispiel zeigt die Verwendung `put-image`.

AWS CLI

Um ein Bild mit seinem Manifest neu zu taggen

Im folgenden `put-image` Beispiel wird ein neues Tag im `hello-world` Repository mit einem vorhandenen Image-Manifest erstellt.

```
aws ecr put-image \  
  --repository-name hello-world \  
  --image-tag 2019.08 \  
  --image-manifest file://hello-world.manifest.json
```

Inhalt von `hello-world.manifest.json`:

```
{  
  "schemaVersion": 2,  
  "mediaType": "application/vnd.docker.distribution.manifest.v2+json",  
  "config": {  
    "mediaType": "application/vnd.docker.container.image.v1+json",  
    "size": 5695,  
    "digest":  
    "sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980"  
  },  
  "layers": [  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 39096921,  
      "digest":  
      "sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295"  
    },  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 57938,  
      "digest":  
      "sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed"  
    },  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 423,  
      "digest":  
      "sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610"  
    },  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 680,  
      "digest":  
      "sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c"  
    },  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
```

```
        "size": 162,
        "digest":
"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b"
    },
    {
        "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
        "size": 28268840,
        "digest":
"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f"
    },
    {
        "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
        "size": 35369152,
        "digest":
"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276"
    },
    {
        "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
        "size": 155,
        "digest":
"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85"
    },
    {
        "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
        "size": 28737,
        "digest":
"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4"
    },
    {
        "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
        "size": 190,
        "digest":
"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7"
    },
    {
        "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
        "size": 28748,
        "digest":
"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee"
    }
]
}
```

Ausgabe:

```
{
  "image": {
    "registryId": "130757420319",
    "repositoryName": "hello-world",
    "imageId": {
      "imageDigest":
"sha256:8ece96b74f87652876199d83bd107d0435a196133af383ac54cb82b6cc5283ae",
      "imageTag": "2019.08"
    },
    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType
\": \"application/vnd.docker.distribution.manifest.v2+json
\",,\n  \"config\": {\n    \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",,\n    \"size\": 5695,\n    \"digest\":
\"sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980\"\n
  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 39096921,\n      \"digest
\": \"sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 57938,\n      \"digest
\": \"sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 423,\n      \"digest\":
\"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610\"\n
    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",,\n
      \"size\": 680,\n      \"digest\":
\"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c
\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 162,\n      \"digest
\": \"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b
\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 28268840,\n      \"digest
\": \"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f
\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 35369152,\n      \"digest
\": \"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 155,\n      \"digest\":
\"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85\"\n
    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",,\n
      \"size\": 28737,\n      \"digest\":
\"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4\"\n
    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",,
```

```

\n      \ "size\ ": 190,\n      \ "digest\ ":
\n      \ "sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7\ "\n      },
\n      {\n      \ "mediaType\ ": \ "application/vnd.docker.image.rootfs.diff.tar.gzip\ ",
\n      \ "size\ ": 28748,\n      \ "digest\ ":
\n      \ "sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee\ "\n
\n      }\n      ]\n      }\n      }
}

```

- API-Einheiten finden Sie [PutImage](#) in der AWS CLI Befehlsreferenz.

put-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `put-lifecycle-policy`.

AWS CLI

Um eine Lebenszyklusrichtlinie zu erstellen

Im folgenden `put-lifecycle-policy` Beispiel wird eine Lebenszyklusrichtlinie für das angegebene Repository in der Standardregistrierung für ein Konto erstellt.

```

aws ecr put-lifecycle-policy \
  --repository-name "project-a/amazon-ecs-sample" \
  --lifecycle-policy-text "file://policy.json"

```

Inhalt von `policy.json`:

```

{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "untagged",
        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 14
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}

```

```
    }
  ]
}
```

Ausgabe:

```
{
  "registryId": "<aws_account_id>",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\"rules\": [{\"rulePriority\": 1, \"description\": \"Expire images older than 14 days\", \"selection\": {\"tagStatus\": \"untagged\", \"countType\": \"sinceImagePushed\", \"countUnit\": \"days\", \"countNumber\": 14}, \"action\": {\"type\": \"expire\"}}]}"
}
```

Weitere Informationen finden Sie unter [Lebenszyklusrichtlinien](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [PutLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

set-repository-policy

Das folgende Codebeispiel zeigt die Verwendung `set-repository-policy`.

AWS CLI

Um die Repository-Richtlinie für ein Repository festzulegen

Im folgenden `set-repository-policy` Beispiel wird eine in einer Datei enthaltene Repository-Richtlinie an das `cluster-autoscaler` Repository angehängt.

```
aws ecr set-repository-policy \
  --repository-name cluster-autoscaler \
  --policy-text file://my-policy.json
```

Inhalt von `my-policy.json`:

```
{
  "Version" : "2008-10-17",
  "Statement" : [
    {
      "Sid" : "allow public pull",
```

```

    "Effect" : "Allow",
    "Principal" : "*",
    "Action" : [
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
    ]
  }
]
}

```

Ausgabe:

```

{
  "registryId": "012345678910",
  "repositoryName": "cluster-autoscaler",
  "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n\n    \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :\n    \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\n    \", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}

```

- API-Einheiten finden Sie [SetRepositoryPolicy](#) in der AWS CLI Befehlsreferenz.

start-image-scan

Das folgende Codebeispiel zeigt die Verwendung `start-image-scan`.

AWS CLI

Um einen Schwachstellenscan für ein Bild zu starten

Im folgenden `start-image-scan` Beispiel wird ein Image-Scan für den Image-Digest im angegebenen Repository gestartet und von diesem spezifiziert.

```

aws ecr start-image-scan \
  --repository-name sample-repo \
  --image-id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6

```

Ausgabe:

```
{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageId": {
    "imageDigest":
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
  },
  "imageScanStatus": {
    "status": "IN_PROGRESS"
  }
}
```

Weitere Informationen finden Sie unter [Scannen von Bildern](#) im ECR Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [StartImageScan](#) in der AWS CLI Befehlsreferenz.

start-lifecycle-policy-preview

Das folgende Codebeispiel zeigt die Verwendung `start-lifecycle-policy-preview`.

AWS CLI

Um eine Lifecycle-Richtlinievorschau zu erstellen

Im folgenden `start-lifecycle-policy-preview` Beispiel wird eine Lifecycle-Policy-Vorschau erstellt, die durch eine JSON-Datei für das angegebene Repository definiert wird.

```
aws ecr start-lifecycle-policy-preview \
  --repository-name "project-a/amazon-ecs-sample" \
  --lifecycle-policy-text "file://policy.json"
```

Inhalt von `policy.json`:

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "untagged",
        "countType": "sinceImagePushed",
        "countUnit": "days",

```



```

        "countNumber": 14
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}

```

Ausgabe:

```

{
  "registryId": "012345678910",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n\n      \"rulePriority\": 1,\n      \"description\": \"Expire images older than 14\n      days\",\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\n\": \"days\",\n        \"countNumber\": 14\n      },\n      \"action\": {\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}",
  "status": "IN_PROGRESS"
}

```

- API-Einheiten finden Sie [StartLifecyclePolicyPreview](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Repository zu taggen

Im folgenden `tag-resource` Beispiel wird ein Tag mit Schlüssel `Stage` und Wert `Integ` für das `hello-world` Repository festgelegt.

```

aws ecr tag-resource \
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
  --tags Key=Stage,Value=Integ

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung eines Repositorys aufzuheben

Im folgenden `untag-resource` Beispiel wird das Tag mit dem Schlüssel `Stage` aus dem `hello-world` Repository entfernt.

```
aws ecr untag-resource \  
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \  
  --tag-keys Stage
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

upload-layer-part

Das folgende Codebeispiel zeigt die Verwendung `upload-layer-part`.

AWS CLI

Um ein Layer-Teil hochzuladen

Im Folgenden wird ein Teil der Bildebene in das `layer-test` Repository `upload-layer-part` hochgeladen.

```
aws ecr upload-layer-part \  
  --repository-name layer-test \  
  --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \  
  --part-first-byte 0 \  
  --part-last-byte 8323314 \  
  --layer-part-blob file:///var/lib/docker/image/overlay2/layerdb/sha256/ff986b10a018b48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e/layer.b64
```

Ausgabe:

```
{
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
  "registryId": "012345678910",
  "lastByteReceived": 8323314,
  "repositoryName": "layer-test"
}
```

- API-Einheiten finden Sie [UploadLayerPart](#) in der AWS CLI Befehlsreferenz.

Amazon ECR Public-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon ECR Public Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-repository

Das folgende Codebeispiel zeigt die Verwendung `create-repository`.

AWS CLI

Beispiel 1: Um ein Repository in einer öffentlichen Registrierung zu erstellen

Das folgende `create-repository` Beispiel erstellt ein Repository, das `project-a/nginx-web-app` in einer öffentlichen Registrierung benannt ist.

```
aws ecr-public create-repository \
  --repository-name project-a/nginx-web-app
```

Ausgabe:

```
{
  "repository": {
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-app",
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/nginx-web-app",
    "createdAt": "2024-07-01T21:08:55.131000+00:00"
  },
  "catalogData": {}
}
```

Weitere Informationen finden Sie unter [Erstellen eines öffentlichen Repositorys](#) im Amazon ECR Public User Guide.

Beispiel 2: So erstellen Sie ein Repository in einer öffentlichen Registrierung mit einer kurzen Beschreibung des Inhalts des Repositorys, der System- und Betriebsarchitektur, mit der die Images im Repository kompatibel sind

Im folgenden `create-repository` Beispiel wird ein `project-a/nginx-web-app` in einer öffentlichen Registrierung benanntes Repository mit einer kurzen Beschreibung des Inhalts des Repositorys, der System- und Betriebsarchitektur erstellt, mit der die Images im Repository kompatibel sind.

```
aws ecr-public create-repository \
  --repository-name project-a/nginx-web-app \
  --catalog-data 'description=My project-a ECR Public Repository, architectures=ARM,ARM 64,x86,x86-64, operatingSystems=Linux'
```

Ausgabe:

```
{
  "repository": {
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-app",
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/nginx-web-app",
  }
}
```

```
    "createdAt": "2024-07-01T21:23:20.455000+00:00"
  },
  "catalogData": {
    "description": "My project-a ECR Public Repository",
    "architectures": [
      "ARM",
      "ARM 64",
      "x86",
      "x86-64"
    ],
    "operatingSystems": [
      "Linux"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Erstellen eines öffentlichen Repositorys](#) im Amazon ECR Public User Guide.

Beispiel 3: So erstellen Sie ein Repository in einer öffentlichen Registrierung zusammen mit Informationen zu `logImageBlobaboutText`, `usageText` und `Tags`

Im folgenden `create-repository` Beispiel wird ein Repository mit dem Namen `project-a/nginx-web-app` in einer öffentlichen Registrierung zusammen mit, `logImageBlobaboutText`, `usageText` und Tag-Informationen erstellt.

```
aws ecr-public create-repository \
  --cli-input-json file://myfile.json
```

Inhalt von `myfile.json`:

```
{
  "repositoryName": "project-a/nginx-web-app",
  "catalogData": {
    "description": "My project-a ECR Public Repository",
    "architectures": [
      "ARM",
      "ARM 64",
      "x86",
      "x86-64"
    ],
    "operatingSystems": [
```

```

    "Linux"
  ],
  "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",
  "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
  Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
  [2.0.20200722.0], [2], [latest](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafb0a875e78/Dockerfile)\n\n* [2.0.20200722.0-
with-sources], [2-with-sources], [with-sources](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n\n* [2018.03.0.20200602.1], [2018.03], [1](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n\n* [2018.03.0.20200602.1-with-sources], [2018.03-with-sources],
  [1-with-sources](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
  Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
  to provide a stable, secure, and high-performance execution environment for
  applications running on Amazon EC2. The full distribution includes packages that
  enable easy integration with AWS, including launch configuration tools and many
  popular AWS libraries and tools. AWS provides ongoing security and maintenance
  updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
  contains a minimal set of packages. To install additional packages, [use `yum`]  

  (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
  provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
  \n\nFor information on security updates for Amazon Linux, please refer to [Amazon
  Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
  Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
  Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
  which does not reflect the state of backported patches for vulnerabilities.\n
  \n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
  container images in any Docker based environment. Examples include, your laptop,
  in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
  available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
  are available under their own licenses; run rpm -qi [package name] or check
  /usr/share/doc/[package name]-* and /usr/share/licenses/[package name]-* for
  details.\n\nAs with all Docker images, these likely also contain other software
  which may be under other licenses (such as Bash, etc from the base distribution,
  along with any direct or indirect dependencies of the primary software being
  contained).\n\nSome additional license information which was able to be auto-
  detected might be found in [the `repo-info` repository's `amazonlinux/` directory]

```

```
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
Security\n\nFor information on security updates for Amazon Linux, please refer
to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
that Docker Hub's vulnerability scanning for Amazon Linux is currently based
on RPM versions, which does not reflect the state of backported patches for
vulnerabilities.",
  "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
images in any Docker based environment. Examples include, your laptop, in Amazon
EC2 instances, and ECS clusters.\n\n## How do I install a software package from
Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
`amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
get ongoing updates from Amazon in the form of security updates, bug fix updates,
and other enhancements. Security bulletins for Amazon Linux are available at
https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
Linux going forward?\n\nYes; in order to avoid any disruption to your existing
applications and to facilitate migration to Amazon Linux 2, AWS will provide
regular security updates for Amazon Linux 2018.03 AMI and container image for 2
years after the final LTS build is announced. You can also use all your existing
support channels such as AWS Support and Amazon Linux Discussion Forum to continue
to submit support requests."
},
  "tags": [
    {
      "Key": "Name",
      "Value": "project-a/nginx-web-app"
    },
    {
      "Key": "Environment",
      "Value": "Prod"
    }
  ]
}
}
```

Ausgabe:

```
{
  "repository": {
```

```
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
    "createdAt": "2024-07-01T21:53:05.749000+00:00"
  },
  "catalogData": {
    "description": "My project-a ECR Public Repository",
    "architectures": [
      "ARM",
      "ARM 64",
      "x86",
      "x86-64"
    ],
    "operatingSystems": [
      "Linux"
    ],
    "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/
logo/23861450-4b9b-403c-9a4c-7aa0ef140bb8/2f9bf5a7-a32f-45b4-b5cd-c5770a35e6d7.png",
    "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
  Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
  tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
  [2.0.20200722.0], [2], [latest](https://github.com/amazonlinux/container-images/
  blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n\n* [2.0.20200722.0-
  with-sources], [2-with-sources], [with-sources](https://github.com/
  amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
  Dockerfile)\n\n* [2018.03.0.20200602.1], [2018.03], [1](https://github.com/
  amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
  Dockerfile)\n\n* [2018.03.0.20200602.1-with-sources], [2018.03-with-sources],
  [1-with-sources](https://github.com/amazonlinux/container-images/
  blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
  Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
  to provide a stable, secure, and high-performance execution environment for
  applications running on Amazon EC2. The full distribution includes packages that
  enable easy integration with AWS, including launch configuration tools and many
  popular AWS libraries and tools. AWS provides ongoing security and maintenance
  updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
  contains a minimal set of packages. To install additional packages, [use `yum`]
  (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
  provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
```



```
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).\n\nFor information on security updates for Amazon Linux, please refer to [Amazon\nLinux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon\nLinux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker\nHub's vulnerability scanning for Amazon Linux is currently based on RPM versions,\nwhich does not reflect the state of backported patches for vulnerabilities.\n\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux\ncontainer images in any Docker based environment. Examples include, your laptop,\nin Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is\navailable under the [GNU General Public License, version 2.0](https://github.com/\naws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages\nare available under their own licenses; run `rpm -qi [package name]` or check\n`/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for\ndetails.\n\nAs with all Docker images, these likely also contain other software\nwhich may be under other licenses (such as Bash, etc from the base distribution,\nalong with any direct or indirect dependencies of the primary software being\ncontained).\n\nSome additional license information which was able to be auto-\ndetected might be found in [the `repo-info` repository's `amazonlinux/` directory]\n(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n## Security\n\nFor information on security updates for Amazon Linux, please refer\nto [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)\nand [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note\nthat Docker Hub's vulnerability scanning for Amazon Linux is currently based\non RPM versions, which does not reflect the state of backported patches for\nvulnerabilities.",\n\n  "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where\ncan I run Amazon Linux container images?\n\nYou can run Amazon Linux container\nimages in any Docker based environment. Examples include, your laptop, in Amazon\nEC2 instances, and ECS clusters.\n\n## How do I install a software package from\nExtras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the\n`amazon-linux-extras` command. Packages can be installed with the `amazon-linux-\nextras install <package>` command. Example: `amazon-linux-extras install rust1`\n\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon\nLinux images for Amazon EC2 and on-premises use, Amazon Linux container images will\nget ongoing updates from Amazon in the form of security updates, bug fix updates,\nand other enhancements. Security bulletins for Amazon Linux are available at\nhttps://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon\nLinux going forward?\n\nYes; in order to avoid any disruption to your existing\napplications and to facilitate migration to Amazon Linux 2, AWS will provide\nregular security updates for Amazon Linux 2018.03 AMI and container image for 2\nyears after the final LTS build is announced. You can also use all your existing\nsupport channels such as AWS Support and Amazon Linux Discussion Forum to continue\nto submit support requests." ]
```

```
}
```

Weitere Informationen finden Sie unter [Erstellen eines öffentlichen Repositorys](#) im Amazon ECR Public User Guide und [Repository-Katalogdaten](#) im Amazon ECR Public User Guide.

- API-Einheiten finden Sie [CreateRepository](#) in der AWS CLI Befehlsreferenz.

delete-repository

Das folgende Codebeispiel zeigt die Verwendung `delete-repository`.

AWS CLI

Um ein Repository in einer öffentlichen Registrierung zu löschen

Im folgenden `delete-repository` Beispiel wird ein `project-a/nginx-web-app` aus Ihrer öffentlichen Registrierung benanntes Repository gelöscht.

```
aws ecr-public delete-repository \  
  --repository-name project-a/nginx-web-app
```

Ausgabe:

```
{  
  "repository": {  
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/  
nginx-web-app",  
    "registryId": "123456789012",  
    "repositoryName": "project-a/nginx-web-app",  
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/  
nginx-web-app",  
    "createdAt": "2024-07-01T22:14:50.103000+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen eines öffentlichen Repositorys](#) im Amazon ECR Public User Guide.

- API-Einheiten finden Sie [DeleteRepository](#) in der AWS CLI Befehlsreferenz.

ECS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden ECS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-capacity-provider

Das folgende Codebeispiel zeigt die Verwendung `create-capacity-provider`.

AWS CLI

Um einen Kapazitätsanbieter zu erstellen

Im folgenden `create-capacity-provider` Beispiel wird ein Kapazitätsanbieter erstellt, der eine Auto Scaling Scaling-Gruppe namens `My` verwendet `ASG`, für die verwaltete Skalierung und der verwaltete Kündigungsschutz aktiviert sind. Diese Konfiguration wird für Amazon ECS Cluster Auto Scaling verwendet.

```
aws ecs create-capacity-provider \  
  --name "MyCapacityProvider" \  
  --auto-scaling-group-provider "autoScalingGroupArn=arn:aws:autoscaling:us-  
east-1:123456789012:autoScalingGroup:57ffcb94-11f0-4d6d-  
bf60-3bac5EXAMPLE:autoScalingGroupName/  
MyASG,managedScaling={status=ENABLED,targetCapacity=100},managedTerminationProtection=ENABLED"
```

Ausgabe:

```
{  
  "capacityProvider": {
```

```

    "capacityProviderArn": "arn:aws:ecs:us-east-1:123456789012:capacity-provider/
MyCapacityProvider",
    "name": "MyCapacityProvider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
        "autoScalingGroupArn": "arn:aws:autoscaling:us-
east-1:132456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/MyASG",
        "managedScaling": {
            "status": "ENABLED",
            "targetCapacity": 100,
            "minimumScalingStepSize": 1,
            "maximumScalingStepSize": 10000,
            "instanceWarmupPeriod": 300
        },
        "managedTerminationProtection": "ENABLED"
    },
    "tags": []
}

```

Weitere Informationen finden Sie unter [Amazon ECS Cluster Auto Scaling](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [CreateCapacityProvider](#) in der AWS CLI Befehlsreferenz.

create-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-cluster`.

AWS CLI

Beispiel 1: Um einen neuen Cluster zu erstellen

Das folgende `create-cluster` Beispiel erstellt einen Cluster.

```

aws ecs create-cluster \
  --cluster-name MyCluster

```

Ausgabe:

```

{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",

```

```
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": []
  }
}
```

Weitere Informationen finden Sie unter [Creating a Cluster](#) im Amazon ECS Developer Guide.

Beispiel 2: So erstellen Sie einen neuen Cluster mithilfe von Kapazitätsanbietern

Im folgenden `create-cluster` Beispiel wird ein Cluster erstellt und ihm zwei bestehende Kapazitätsanbieter zugeordnet. Der `create-capacity-provider` Befehl wird verwendet, um einen Kapazitätsanbieter zu erstellen. Die Angabe einer Standardstrategie für Kapazitätsanbieter ist optional, wird jedoch empfohlen. In diesem Beispiel erstellen wir einen Cluster mit dem Namen `MyCluster` und ordnen ihm die Kapazitätsanbieter `MyCapacityProvider1` und die `MyCapacityProvider2` Kapazitätsanbieter zu. Es wird eine Standardstrategie für Kapazitätsanbieter angegeben, bei der die Aufgaben gleichmäßig auf beide Kapazitätsanbieter verteilt werden.

```
aws ecs create-cluster --cluster-name MyCluster --capacity-providers MyCapacityProvider 1
MyCapacityProvider 2 -- default-capacity-provider-strategy capacityProvider = 1, weight=1 = 2,
weight=1 MyCapacityProvider capacityProvider MyCapacityProvider
```

Ausgabe:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "PROVISIONING",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "settings": [
      {
```

```

        "name": "containerInsights",
        "value": "enabled"
    }
],
"capacityProviders": [
    "MyCapacityProvider1",
    "MyCapacityProvider2"
],
"defaultCapacityProviderStrategy": [
    {
        "capacityProvider": "MyCapacityProvider1",
        "weight": 1,
        "base": 0
    },
    {
        "capacityProvider": "MyCapacityProvider2",
        "weight": 1,
        "base": 0
    }
],
"attachments": [
    {
        "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
        "type": "asp",
        "status": "PRECREATED",
        "details": [
            {
                "name": "capacityProviderName",
                "value": "MyCapacityProvider1"
            },
            {
                "name": "scalingPlanName",
                "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
            }
        ]
    },
    {
        "id": "ae592060-2382-4663-9476-b015c685593c",
        "type": "asp",
        "status": "PRECREATED",
        "details": [
            {
                "name": "capacityProviderName",

```

```

        "value": "MyCapacityProvider2"
      },
      {
        "name": "scalingPlanName",
        "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
      }
    ]
  },
  "attachmentsStatus": "UPDATE_IN_PROGRESS"
}

```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

Beispiel 3: Um einen neuen Cluster mit mehreren Tags zu erstellen

Das folgende `create-cluster` Beispiel erstellt einen Cluster mit mehreren Tags. Weitere Informationen zum Hinzufügen von Tags mithilfe von Kurzsyntax finden Sie unter [Verwenden von Kurzsyntax mit der AWS Befehlszeilenschnittstelle im Benutzerhandbuch](#).AWS CLI

```

aws ecs create-cluster \
  --cluster-name MyCluster \
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3

```

Ausgabe:

```

{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [
      {
        "key": "key1",
        "value": "value1"
      }
    ]
  }
}

```

```

    },
    {
      "key": "key2",
      "value": "value2"
    },
    {
      "key": "key3",
      "value": "value3"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Creating a Cluster](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [CreateCluster](#) in der AWS CLI Befehlsreferenz.

create-service

Das folgende Codebeispiel zeigt die Verwendung `create-service`.

AWS CLI

Beispiel 1: So erstellen Sie einen Service mit einer Fargate-Aufgabe

Das folgende `create-service` Beispiel zeigt, wie ein Service mithilfe einer Fargate-Aufgabe erstellt wird.

```

aws ecs create-service \
  --cluster MyCluster \
  --service-name MyService \
  --task-definition sample-fargate:1 \
  --desired-count 2 \
  --launch-type FARGATE \
  --platform-version LATEST \
  --network-
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],a
  \
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3

```

Ausgabe:

```
{
```



```
"service": {
  "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/
MyService",
  "serviceName": "MyService",
  "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
  "loadBalancers": [],
  "serviceRegistries": [],
  "status": "ACTIVE",
  "desiredCount": 2,
  "runningCount": 0,
  "pendingCount": 0,
  "launchType": "FARGATE",
  "platformVersion": "LATEST",
  "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:1",
  "deploymentConfiguration": {
    "maximumPercent": 200,
    "minimumHealthyPercent": 100
  },
  "deployments": [
    {
      "id": "ecs-svc/1234567890123456789",
      "status": "PRIMARY",
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sample-fargate:1",
      "desiredCount": 2,
      "pendingCount": 0,
      "runningCount": 0,
      "createdAt": 1557119253.821,
      "updatedAt": 1557119253.821,
      "launchType": "FARGATE",
      "platformVersion": "1.3.0",
      "networkConfiguration": {
        "awsvpcConfiguration": {
          "subnets": [
            "subnet-12344321"
          ],
          "securityGroups": [
            "sg-12344321"
          ],
          "assignPublicIp": "ENABLED"
        }
      }
    }
  ]
}
```

```
    ],
    "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
    "events": [],
    "createdAt": 1557119253.821,
    "placementConstraints": [],
    "placementStrategy": [],
    "networkConfiguration": {
      "awsvpcConfiguration": {
        "subnets": [
          "subnet-12344321"
        ],
        "securityGroups": [
          "sg-12344321"
        ],
        "assignPublicIp": "ENABLED"
      }
    },
    "schedulingStrategy": "REPLICA",
    "tags": [
      {
        "key": "key1",
        "value": "value1"
      },
      {
        "key": "key2",
        "value": "value2"
      },
      {
        "key": "key3",
        "value": "value3"
      }
    ],
    "enableECSTags": false,
    "propagateTags": "NONE"
  }
}
```

Beispiel 2: Um einen Dienst mit dem EC2 Starttyp zu erstellen

Das folgende `create-service` Beispiel zeigt, wie Sie einen Dienst erstellen, der `ecs-simple-service` mit einer Aufgabe aufgerufen wird, die den EC2 Starttyp verwendet. Der Dienst verwendet die `sleep360` Aufgabendefinition und verwaltet eine Instanziierung der Aufgabe.

```
aws ecs create-service \  
  --cluster MyCluster \  
  --service-name ecs-simple-service \  
  --task-definition sleep360:2 \  
  --desired-count 1
```

Ausgabe:

```
{  
  "service": {  
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/ecs-  
simple-service",  
    "serviceName": "ecs-simple-service",  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "status": "ACTIVE",  
    "desiredCount": 1,  
    "runningCount": 0,  
    "pendingCount": 0,  
    "launchType": "EC2",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sleep360:2",  
    "deploymentConfiguration": {  
      "maximumPercent": 200,  
      "minimumHealthyPercent": 100  
    },  
    "deployments": [  
      {  
        "id": "ecs-svc/1234567890123456789",  
        "status": "PRIMARY",  
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-  
definition/sleep360:2",  
        "desiredCount": 1,  
        "pendingCount": 0,  
        "runningCount": 0,  
        "createdAt": 1557206498.798,  
        "updatedAt": 1557206498.798,  
        "launchType": "EC2"  
      }  
    ],  
    "events": [],  
    "createdAt": 1557206498.798,  
  }  
}
```

```

    "placementConstraints": [],
    "placementStrategy": [],
    "schedulingStrategy": "REPLICA",
    "enableECSManagedTags": false,
    "propagateTags": "NONE"
  }
}

```

Beispiel 3: Um einen Dienst zu erstellen, der einen externen Deployment Controller verwendet

Im folgenden `create-service` Beispiel wird ein Dienst erstellt, der einen externen Deployment Controller verwendet.

```

aws ecs create-service \
  --cluster MyCluster \
  --service-name MyService \
  --deployment-controller type=EXTERNAL \
  --desired-count 1

```

Ausgabe:

```

{
  "service": {
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",
    "serviceName": "MyService",
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "loadBalancers": [],
    "serviceRegistries": [],
    "status": "ACTIVE",
    "desiredCount": 1,
    "runningCount": 0,
    "pendingCount": 0,
    "launchType": "EC2",
    "deploymentConfiguration": {
      "maximumPercent": 200,
      "minimumHealthyPercent": 100
    },
    "taskSets": [],
    "deployments": [],
    "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/ecs.amazonaws.com/AWSServiceRoleForECS",
  }
}

```

```

    "events": [],
    "createdAt": 1557128207.101,
    "placementConstraints": [],
    "placementStrategy": [],
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
      "type": "EXTERNAL"
    },
    "enableECSTags": false,
    "propagateTags": "NONE"
  }
}

```

Beispiel 4: Um einen neuen Dienst hinter einem Load Balancer zu erstellen

Das folgende `create-service` Beispiel zeigt, wie Sie einen Dienst erstellen, der sich hinter einem Load Balancer befindet. Sie müssen einen Load Balancer in derselben Region wie Ihre Container-Instance konfiguriert haben. In diesem Beispiel werden die `--cli-input-json` Option und eine JSON Eingabedatei `ecs-simple-service-elb.json` mit dem folgenden Inhalt aufgerufen:

```

{
  "serviceName": "ecs-simple-service-elb",
  "taskDefinition": "ecs-demo",
  "loadBalancers": [
    {
      "loadBalancerName": "EC2Contai-EcsElast-123456789012",
      "containerName": "simple-demo",
      "containerPort": 80
    }
  ],
  "desiredCount": 10,
  "role": "ecsServiceRole"
}

```

Befehl:

```

aws ecs create-service \
  --cluster MyCluster \
  --service-name ecs-simple-service-elb \
  --cli-input-json file://ecs-simple-service-elb.json

```

Ausgabe:

```
{
  "service": {
    "status": "ACTIVE",
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/ecs-
demo:1",
    "pendingCount": 0,
    "loadBalancers": [
      {
        "containerName": "ecs-demo",
        "containerPort": 80,
        "loadBalancerName": "EC2Contai-EcsElast-123456789012"
      }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/ecsServiceRole",
    "desiredCount": 10,
    "serviceName": "ecs-simple-service-elb",
    "clusterArn": "arn:aws:ecs:<us-west-2:123456789012:cluster/MyCluster",
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/ecs-simple-
service-elb",
    "deployments": [
      {
        "status": "PRIMARY",
        "pendingCount": 0,
        "createdAt": 1428100239.123,
        "desiredCount": 10,
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/ecs-demo:1",
        "updatedAt": 1428100239.123,
        "id": "ecs-svc/1234567890123456789",
        "runningCount": 0
      }
    ],
    "events": [],
    "runningCount": 0
  }
}
```

Weitere Informationen finden Sie unter [Creating a Service](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [CreateService](#) in der AWS CLI Befehlsreferenz.

create-task-set

Das folgende Codebeispiel zeigt die Verwendung `create-task-set`.

AWS CLI

Um einen Tasksatz zu erstellen

Im folgenden `create-task-set` Beispiel wird ein Taskset in einem Dienst erstellt, der einen externen Deployment Controller verwendet.

```
aws ecs create-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-definition MyTaskDefinition:2 \  
  --network-  
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

Ausgabe:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "ACTIVE",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
MyTaskDefinition:2",  
    "computedDesiredCount": 0,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557128360.711,  
    "updatedAt": 1557128360.711,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12344321"  
        ],  
        "securityGroups": [  
          "sg-12344321"  
        ],  
        "assignPublicIp": "DISABLED"  
      }  
    }  
  }  
}
```

```

    }
  },
  "loadBalancers": [],
  "serviceRegistries": [],
  "scale": {
    "value": 0.0,
    "unit": "PERCENT"
  },
  "stabilityStatus": "STABILIZING",
  "stabilityStatusAt": 1557128360.711
}
}

```

- API-Einheiten finden Sie [CreateTaskSet](#) unter AWS CLI Befehlsreferenz.

delete-account-setting

Das folgende Codebeispiel zeigt die Verwendung `delete-account-setting`.

AWS CLI

Um die Kontoeinstellungen für einen bestimmten IAM Benutzer oder eine bestimmte IAM Rolle zu löschen

Im folgenden Beispiel `delete-account-setting` werden die Kontoeinstellungen für den bestimmten IAM Benutzer oder die IAM Rolle gelöscht.

```

aws ecs delete-account-setting \
  --name serviceLongArnFormat \
  --principal-arn arn:aws:iam::123456789012:user/MyUser

```

Ausgabe:

```

{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"
  }
}

```


Weitere Informationen finden Sie unter [Amazon Resource Names \(ARNs\) und IDs](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DeleteAccountSetting](#) in der AWS CLI Befehlsreferenz.

delete-attributes

Das folgende Codebeispiel zeigt die Verwendung `delete-attributes`.

AWS CLI

Um ein oder mehrere benutzerdefinierte Attribute aus einer ECS Amazon-Ressource zu löschen

Im Folgenden `delete-attributes` wird ein Attribut mit dem Namen `stack` aus einer Container-Instance gelöscht.

```
aws ecs delete-attributes \  
  --attributes name=stack, targetId=arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Ausgabe:

```
{  
  "attributes": [  
    {  
      "name": "stack",  
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",  
      "value": "production"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [DeleteAttributes AWS CLI](#) Befehlsreferenz.

delete-capacity-provider

Das folgende Codebeispiel zeigt die Verwendung `delete-capacity-provider`.

AWS CLI

Beispiel 1: So löschen Sie einen Kapazitätsanbieter mithilfe des Amazon-Ressourcennamens (ARN)

Im folgenden `delete-capacity-provider` Beispiel wird ein Kapazitätsanbieter gelöscht, indem der Amazon-Ressourcenname (ARN) des Kapazitätsanbieters angegeben wird. Sowohl der Status ARN als auch der Status der Löschung des Kapazitätsanbieters können mit dem `describe-capacity-providers` Befehl abgerufen werden.

```
aws ecs delete-capacity-provider \  
  --capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/  
ExampleCapacityProvider
```

Ausgabe:

```
{  
  "capacityProvider": {  
    "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-  
provider/ExampleCapacityProvider",  
    "name": "ExampleCapacityProvider",  
    "status": "ACTIVE",  
    "autoScalingGroupProvider": {  
      "autoScalingGroupArn": "arn:aws:autoscaling:us-  
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",  
      "managedScaling": {  
        "status": "ENABLED",  
        "targetCapacity": 100,  
        "minimumScalingStepSize": 1,  
        "maximumScalingStepSize": 10000  
      },  
      "managedTerminationProtection": "DISABLED"  
    },  
    "updateStatus": "DELETE_IN_PROGRESS",  
    "tags": []  
  }  
}
```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

Beispiel 2: Um einen Kapazitätsanbieter unter Verwendung des Namens zu löschen

Im folgenden `delete-capacity-provider` Beispiel wird ein Kapazitätsanbieter gelöscht, indem der Kurzname des Kapazitätsanbieters angegeben wird. Der Kurzname sowie der Status der Löschung des Kapazitätsanbieters können mit dem `describe-capacity-providers` Befehl abgerufen werden.

```
aws ecs delete-capacity-provider \  
  --capacity-provider ExampleCapacityProvider
```

Ausgabe:

```
{  
  "capacityProvider": {  
    "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-  
provider/ExampleCapacityProvider",  
    "name": "ExampleCapacityProvider",  
    "status": "ACTIVE",  
    "autoScalingGroupProvider": {  
      "autoScalingGroupArn": "arn:aws:autoscaling:us-  
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",  
      "managedScaling": {  
        "status": "ENABLED",  
        "targetCapacity": 100,  
        "minimumScalingStepSize": 1,  
        "maximumScalingStepSize": 10000  
      },  
      "managedTerminationProtection": "DISABLED"  
    },  
    "updateStatus": "DELETE_IN_PROGRESS",  
    "tags": []  
  }  
}
```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DeleteCapacityProvider](#) in der AWS CLI Befehlsreferenz.

delete-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster`.

AWS CLI

Um einen leeren Cluster zu löschen

Im folgenden `delete-cluster` Beispiel wird der angegebene leere Cluster gelöscht.

```
aws ecs delete-cluster --cluster MyCluster
```

Ausgabe:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "status": "INACTIVE",
    "clusterName": "MyCluster",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0
    "statistics": [],
    "tags": []
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines Clusters](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DeleteCluster](#) in der AWS CLI Befehlsreferenz.

delete-service

Das folgende Codebeispiel zeigt die Verwendung `delete-service`.

AWS CLI

Um einen Dienst zu löschen

Im folgenden `ecs delete-service` Beispiel wird der angegebene Dienst aus einem Cluster gelöscht. Sie können den `--force` Parameter angeben, um einen Dienst auch dann zu löschen, wenn er nicht auf Null Aufgaben skaliert wurde.

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

Weitere Informationen finden Sie unter [Löschen eines Service](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DeleteService](#) in der AWS CLI Befehlsreferenz.

delete-task-definitions

Das folgende Codebeispiel zeigt die Verwendung `delete-task-definitions`.

AWS CLI

Um eine Aufgabendefinition zu löschen

Im folgenden `delete-task-definitions` Beispiel wird eine INACTIVE Aufgabendefinition gelöscht.

```
aws ecs delete-task-definitions \  
  --task-definition curltest:1
```

Ausgabe:

```
{  
  "taskDefinitions": [  
    {  
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/  
curltest:1",  
      "containerDefinitions": [  
        {  
          "name": "ctest",  
          "image": "mreferre/eksutils",  
          "cpu": 0,  
          "portMappings": [],  
          "essential": true,  
          "entryPoint": [  
            "sh",  
            "-c"  
          ],  
          "command": [  
            "curl ${ECS_CONTAINER_METADATA_URI_V4}/task"  
          ],  
        }  
      ]  
    }  
  ]  
}
```

```
        "environment": [],
        "mountPoints": [],
        "volumesFrom": [],
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
                "awslogs-create-group": "true",
                "awslogs-group": "/ecs/curltest",
                "awslogs-region": "us-east-1",
                "awslogs-stream-prefix": "ecs"
            }
        }
    },
    ],
    "family": "curltest",
    "taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
    "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
    "networkMode": "awsvpc",
    "revision": 1,
    "volumes": [],
    "status": "DELETE_IN_PROGRESS",
    "compatibilities": [
        "EC2",
        "FARGATE"
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    ],
    "cpu": "256",
    "memory": "512",
    "registeredAt": "2021-09-10T12:56:24.704000+00:00",
    "deregisteredAt": "2023-03-14T15:20:59.419000+00:00",
    "registeredBy": "arn:aws:sts::123456789012:assumed-role/Admin/jdoe"
    }
],
"failures": []
}
```

Weitere Informationen finden Sie unter [ECS Amazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DeleteTaskDefinitions](#) in der AWS CLI Befehlsreferenz.

delete-task-set

Das folgende Codebeispiel zeigt die Verwendung `delete-task-set`.

AWS CLI

Um einen Tasksatz zu löschen

Das folgende `delete-task-set` Beispiel zeigt, wie ein Task-Set gelöscht wird. Sie können den `--force` Parameter hinzufügen, um eine Aufgabengruppe zu löschen, auch wenn sie nicht auf Null skaliert wurde.

```
aws ecs delete-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-  
svc/1234567890123456789 \  
  --force
```

Ausgabe:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "DRAINING",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
    "computedDesiredCount": 0,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557130260.276,  
    "updatedAt": 1557130290.707,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12345678"  
        ],  
        "securityGroups": [  
          "sg-12345678"  
        ],  
      }  
    }  
  }  
}
```

```

        "assignPublicIp": "DISABLED"
      }
    },
    "loadBalancers": [],
    "serviceRegistries": [],
    "scale": {
      "value": 0.0,
      "unit": "PERCENT"
    },
    "stabilityStatus": "STABILIZING",
    "stabilityStatusAt": 1557130290.707
  }
}

```

- API-Einheiten finden Sie unter [DeleteTaskSet AWS CLI-Befehlsreferenz](#).

deregister-container-instance

Das folgende Codebeispiel zeigt die Verwendung `deregister-container-instance`.

AWS CLI

Um eine Container-Instance von einem Cluster abzumelden

Im folgenden `deregister-container-instance` Beispiel wird die Registrierung einer Container-Instance aus dem angegebenen Cluster aufgehoben. Wenn in der Container-Instance noch Aufgaben ausgeführt werden, müssen Sie diese Aufgaben entweder beenden, bevor Sie die Registrierung aufheben, oder die Option verwenden. `--force`

```

aws ecs deregister-container-instance \
  --cluster arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
  --container-instance arn:aws:ecs:us-west-2:123456789012:container-instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --force

```

Ausgabe:

```

{
  "containerInstance": {
    "remainingResources": [
      {
        "integerValue": 1024,

```



```
        "doubleValue": 0.0,
        "type": "INTEGER",
        "longValue": 0,
        "name": "CPU"
    },
    {
        "integerValue": 985,
        "doubleValue": 0.0,
        "type": "INTEGER",
        "longValue": 0,
        "name": "MEMORY"
    },
    {
        "type": "STRINGSET",
        "integerValue": 0,
        "name": "PORTS",
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
        ],
        "longValue": 0,
        "doubleValue": 0.0
    },
    {
        "type": "STRINGSET",
        "integerValue": 0,
        "name": "PORTS_UDP",
        "stringSetValue": [],
        "longValue": 0,
        "doubleValue": 0.0
    }
],
"agentConnected": true,
"attributes": [
    {
        "name": "ecs.capability.secrets.asm.environment-variables"
    },
    {
        "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
    }
]
```

```
    "value": "ami-01a82c3fce2c3ba58",
    "name": "ecs.ami-id"
  },
  {
    "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
  },
  {
    "name": "com.amazonaws.ecs.capability.logging-driver.none"
  },
  {
    "name": "ecs.capability.ecr-endpoint"
  },
  {
    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
  },
  {
    "value": "vpc-1234567890123467",
    "name": "ecs.vpc-id"
  },
  {
    "name": "ecs.capability.execution-role-awslogs"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
  },
  {
    "name": "ecs.capability.docker-plugin.local"
  },
  {
    "name": "ecs.capability.task-eni"
  },
  {
    "name": "ecs.capability.task-cpu-mem-limit"
  },
  {
    "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
  },
  {
```

```
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
  },
  {
    "name": "ecs.capability.execution-role-ecr-pull"
  },
  {
    "name": "ecs.capability.container-health-check"
  },
  {
    "value": "subnet-1234567890123467",
    "name": "ecs.subnet-id"
  },
  {
    "value": "us-west-2a",
    "name": "ecs.availability-zone"
  },
  {
    "value": "t2.micro",
    "name": "ecs.instance-type"
  },
  {
    "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
  },
  {
    "name": "ecs.capability.aws-appmesh"
  },
  {
    "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
  },
  },
```

```
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
},
{
  "name": "com.amazonaws.ecs.capability.privileged-container"
},
{
  "name": "ecs.capability.container-ordering"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
},
{
  "value": "x86_64",
  "name": "ecs.cpu-architecture"
},
{
  "value": "93f43776-2018.10.0",
  "name": "ecs.capability.cni-plugin-version"
},
{
  "name": "ecs.capability.secrets.ssm.environment-variables"
},
{
  "name": "ecs.capability.pid-ipc-namespace-sharing"
},
{
  "name": "com.amazonaws.ecs.capability.ecr-auth"
},
{
  "value": "linux",
  "name": "ecs.os-type"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
```

```
    },
    {
      "name": "ecs.capability.task-eia"
    },
    {
      "name": "ecs.capability.private-registry-
authentication.secretsmanager"
    },
    {
      "name": "com.amazonaws.ecs.capability.task-iam-role"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
    }
  ],
  "pendingTasksCount": 0,
  "tags": [],
  "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "registeredResources": [
    {
      "integerValue": 1024,
      "doubleValue": 0.0,
      "type": "INTEGER",
      "longValue": 0,
      "name": "CPU"
    },
    {
      "integerValue": 985,
      "doubleValue": 0.0,
      "type": "INTEGER",
      "longValue": 0,
      "name": "MEMORY"
    },
    {
      "type": "STRINGSET",
      "integerValue": 0,
      "name": "PORTS",
      "stringSetValue": [
        "22",
        "2376",
        "2375",
        "51678",
        "51679"
      ]
    }
  ]
}
```

```
    ],
    "longValue": 0,
    "doubleValue": 0.0
  },
  {
    "type": "STRINGSET",
    "integerValue": 0,
    "name": "PORTS_UDP",
    "stringSetValue": [],
    "longValue": 0,
    "doubleValue": 0.0
  }
],
"status": "INACTIVE",
"registeredAt": 1557768075.681,
"version": 4,
"versionInfo": {
  "agentVersion": "1.27.0",
  "agentHash": "aabe65ee",
  "dockerVersion": "DockerVersion: 18.06.1-ce"
},
"attachments": [],
"runningTasksCount": 0,
"ec2InstanceId": "i-12345678901234678"
}
}
```

Weitere Informationen finden Sie unter [Deregister a Container Instance](#) im Developer Guide. ECS

- API-Einheiten finden Sie [DeregisterContainerInstance](#) in der AWS CLI Befehlsreferenz.

deregister-task-definition

Das folgende Codebeispiel zeigt die Verwendung `deregister-task-definition`.

AWS CLI

Um die Registrierung einer Aufgabendefinition aufzuheben

Im folgenden `deregister-task-definition` Beispiel wird die Registrierung der ersten Version der `curler` Aufgabendefinition in Ihrer Standardregion aufgehoben.

```
aws ecs deregister-task-definition --task-definition curler:1
```

Beachten Sie, dass in der resultierenden Ausgabe der Status der Aufgabendefinition wie folgt angezeigt wird: INACTIVE

```
{
  "taskDefinition": {
    "status": "INACTIVE",
    "family": "curler",
    "volumes": [],
    "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/
curler:1",
    "containerDefinitions": [
      {
        "environment": [],
        "name": "curler",
        "mountPoints": [],
        "image": "curl:latest",
        "cpu": 100,
        "portMappings": [],
        "entryPoint": [],
        "memory": 256,
        "command": [
          "curl -v http://example.com/"
        ],
        "essential": true,
        "volumesFrom": []
      }
    ],
    "revision": 1
  }
}
```

Weitere Informationen finden Sie unter [ECSAmazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DeregisterTaskDefinition](#) in der AWS CLI Befehlsreferenz.

describe-capacity-providers

Das folgende Codebeispiel zeigt die Verwendung `describe-capacity-providers`.

AWS CLI

Beispiel 1: Um alle Kapazitätsanbieter zu beschreiben

Im folgenden `describe-capacity-providers` Beispiel werden Details zu allen Kapazitätsanbietern abgerufen.

```
aws ecs describe-capacity-providers
```

Ausgabe:

```
{
  "capacityProviders": [
    {
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/MyCapacityProvider",
      "name": "MyCapacityProvider",
      "status": "ACTIVE",
      "autoScalingGroupProvider": {
        "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
        "managedScaling": {
          "status": "ENABLED",
          "targetCapacity": 100,
          "minimumScalingStepSize": 1,
          "maximumScalingStepSize": 1000
        },
        "managedTerminationProtection": "ENABLED"
      },
      "tags": []
    },
    {
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE",
      "name": "FARGATE",
      "status": "ACTIVE",
      "tags": []
    },
    {
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE_SPOT",
      "name": "FARGATE_SPOT",
      "status": "ACTIVE",
      "tags": []
    }
  ]
}
```



```
}
```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

Beispiel 2: Um einen bestimmten Kapazitätsanbieter zu beschreiben

Im folgenden `describe-capacity-providers` Beispiel werden Details zu einem bestimmten Kapazitätsanbieter abgerufen. Durch die Verwendung des `--include TAGS` Parameters werden der Ausgabe die mit dem Kapazitätsanbieter verknüpften Tags hinzugefügt.

```
aws ecs describe-capacity-providers \  
  --capacity-providers MyCapacityProvider \  
  --include TAGS
```

Ausgabe:

```
{  
  "capacityProviders": [  
    {  
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-  
provider/MyCapacityProvider",  
      "name": "MyCapacityProvider",  
      "status": "ACTIVE",  
      "autoScalingGroupProvider": {  
        "autoScalingGroupArn": "arn:aws:autoscaling:us-  
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",  
        "managedScaling": {  
          "status": "ENABLED",  
          "targetCapacity": 100,  
          "minimumScalingStepSize": 1,  
          "maximumScalingStepSize": 1000  
        },  
        "managedTerminationProtection": "ENABLED"  
      },  
      "tags": [  
        {  
          "key": "environment",  
          "value": "production"  
        }  
      ]  
    }  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DescribeCapacityProviders](#) in der AWS CLI Befehlsreferenz.

describe-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-clusters`.

AWS CLI

Beispiel 1: Um einen Cluster zu beschreiben

Im folgenden `describe-clusters` Beispiel werden Details zum angegebenen Cluster abgerufen.

```
aws ecs describe-clusters \
  --cluster default
```

Ausgabe:

```
{
  "clusters": [
    {
      "status": "ACTIVE",
      "clusterName": "default",
      "registeredContainerInstancesCount": 0,
      "pendingTasksCount": 0,
      "runningTasksCount": 0,
      "activeServicesCount": 1,
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"
    }
  ],
  "failures": []
}
```

Weitere Informationen finden Sie unter [Amazon ECS Clusters](#) im Amazon ECS Developer Guide.

Beispiel 2: Um einen Cluster mit der Anhangsoption zu beschreiben

Das folgende `describe-clusters` Beispiel spezifiziert die `ATTACHMENTS` Option. Es ruft Details über den angegebenen Cluster und eine Liste der Ressourcen ab, die dem Cluster in Form von Anhängen zugeordnet sind. Wenn Sie einen Kapazitätsanbieter mit einem Cluster verwenden, werden die Ressourcen, entweder AutoScaling Pläne oder Skalierungsrichtlinien, als `asp` oder `ATTACHMENTS as_policy` dargestellt.

```
aws ecs describe-clusters \  
  --include ATTACHMENTS \  
  --clusters sampleCluster
```

Ausgabe:

```
{  
  "clusters": [  
    {  
      "clusterArn": "arn:aws:ecs:af-south-1:123456789222:cluster/  
sampleCluster",  
      "clusterName": "sampleCluster",  
      "status": "ACTIVE",  
      "registeredContainerInstancesCount": 0,  
      "runningTasksCount": 0,  
      "pendingTasksCount": 0,  
      "activeServicesCount": 0,  
      "statistics": [],  
      "tags": [],  
      "settings": [],  
      "capacityProviders": [  
        "sampleCapacityProvider"  
      ],  
      "defaultCapacityProviderStrategy": [],  
      "attachments": [  
        {  
          "id": "a1b2c3d4-5678-901b-cdef-EXAMPLE22222",  
          "type": "as_policy",  
          "status": "CREATED",  
          "details": [  
            {  
              "name": "capacityProviderName",  
              "value": "sampleCapacityProvider"  
            },  
            {  
              "name": "scalingPolicyName",
```

```

        "value": "ECSManagedAutoScalingPolicy-3048e262-
fe39-4eaf-826d-6f975d303188"
      }
    ]
  },
  "attachmentsStatus": "UPDATE_COMPLETE"
}
],
"failures": []
}

```

Weitere Informationen finden Sie unter [Amazon ECS Clusters](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DescribeClusters](#) in der AWS CLI Befehlsreferenz.

describe-container-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-container-instances`.

AWS CLI

Um die Container-Instance zu beschreiben

Im folgenden `describe-container-instances` Beispiel werden Details für eine Container-Instance im `update` Cluster abgerufen, wobei die Container-Instance UUID als ID verwendet wird.

```

aws ecs describe-container-instances \
  --cluster update \
  --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE

```

Ausgabe:

```

{
  "failures": [],
  "containerInstances": [
    {
      "status": "ACTIVE",
      "registeredResources": [
        {
          "integerValue": 2048,
          "longValue": 0,

```

```
        "type": "INTEGER",
        "name": "CPU",
        "doubleValue": 0.0
    },
    {
        "integerValue": 3955,
        "longValue": 0,
        "type": "INTEGER",
        "name": "MEMORY",
        "doubleValue": 0.0
    },
    {
        "name": "PORTS",
        "longValue": 0,
        "doubleValue": 0.0,
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678"
        ],
        "type": "STRINGSET",
        "integerValue": 0
    }
],
"ec2InstanceId": "i-A1B2C3D4",
"agentConnected": true,
"containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
"pendingTasksCount": 0,
"remainingResources": [
    {
        "integerValue": 2048,
        "longValue": 0,
        "type": "INTEGER",
        "name": "CPU",
        "doubleValue": 0.0
    },
    {
        "integerValue": 3955,
        "longValue": 0,
        "type": "INTEGER",
        "name": "MEMORY",
        "doubleValue": 0.0
    }
]
```

```

    },
    {
      "name": "PORTS",
      "longValue": 0,
      "doubleValue": 0.0,
      "stringSetValue": [
        "22",
        "2376",
        "2375",
        "51678"
      ],
      "type": "STRINGSET",
      "integerValue": 0
    }
  ],
  "runningTasksCount": 0,
  "versionInfo": {
    "agentVersion": "1.0.0",
    "agentHash": "4023248",
    "dockerVersion": "DockerVersion: 1.5.0"
  }
}
]
}

```

Weitere Informationen finden Sie unter [Amazon ECS Container Instances](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DescribeContainerInstances](#) in der AWS CLI Befehlsreferenz.

describe-services

Das folgende Codebeispiel zeigt die Verwendung `describe-services`.

AWS CLI

Um einen Dienst zu beschreiben

Im folgenden `describe-services` Beispiel werden Details für den `my-http-service` Dienst im Standardcluster abgerufen.

```
aws ecs describe-services --services my-http-service
```

Ausgabe:

```
{
  "services": [
    {
      "status": "ACTIVE",
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:1",
      "pendingCount": 0,
      "loadBalancers": [],
      "desiredCount": 10,
      "createdAt": 1466801808.595,
      "serviceName": "my-http-service",
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
      "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/my-http-
service",
      "deployments": [
        {
          "status": "PRIMARY",
          "pendingCount": 0,
          "createdAt": 1466801808.595,
          "desiredCount": 10,
          "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:1",
          "updatedAt": 1428326312.703,
          "id": "ecs-svc/1234567890123456789",
          "runningCount": 10
        }
      ],
      "events": [
        {
          "message": "(service my-http-service) has reached a steady
state.",
          "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
          "createdAt": 1466801812.435
        }
      ],
      "runningCount": 10
    }
  ],
  "failures": []
}
```

Weitere Informationen finden Sie unter [Services](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DescribeServices](#) in der AWS CLI Befehlsreferenz.

describe-task-definition

Das folgende Codebeispiel zeigt die Verwendung `describe-task-definition`.

AWS CLI

Um eine Aufgabendefinition zu beschreiben

Im folgenden `describe-task-definition` Beispiel werden die Details einer Aufgabendefinition abgerufen.

```
aws ecs describe-task-definition \  
--task-definition hello_world:8
```

Ausgabe:

```
{  
  "taskDefinition": {  
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/  
hello_world:8",  
    "containerDefinitions": [  
      {  
        "cpu": 10,  
        "environment": [],  
        "essential": true,  
        "image": "wordpress",  
        "links": [  
          "mysql"  
        ] ,  
        "memory": 500,  
        "mountPoints": [],  
        "name": "wordpress",  
        "portMappings": [  
          {  
            "containerPort": 80,  
            "hostPort": 80  
          }  
        ],  
      }  
    ],  
  }  
}
```



```
        "volumesFrom": []
      },
      {
        "cpu": 10,
        "environment": [
          {
            "name": "MYSQL_ROOT_PASSWORD",
            "value": "password"
          }
        ],
        "essential": true,
        "image": "mysql",
        "memory": 500,
        "mountPoints": [],
        "name": "mysql",
        "portMappings": [],
        "volumesFrom": []
      }
    ],
    "family": "hello_world",
    "revision": 8,
    "volumes": [],
    "status": "ACTIVE",
    "placementConstraints": [],
    "compatibilities": [
      "EXTERNAL",
      "EC2"
    ],
    "registeredAt": "2024-06-21T11:15:12.669000-05:00",
    "registeredBy": "arn:aws:sts::012345678910:assumed-role/demo-role/jane-doe"
  },
  "tags": []
}
```

Weitere Informationen finden Sie unter [ECSAmazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DescribeTaskDefinition](#) in der AWS CLI Befehlsreferenz.

describe-task-sets

Das folgende Codebeispiel zeigt die Verwendung `describe-task-sets`.

AWS CLI

Um einen Tasksatz zu beschreiben

Das folgende `describe-task-sets` Beispiel beschreibt einen Tasksatz in einem Service, der einen externen Deployer verwendet.

```
aws ecs describe-task-sets \  
  --cluster MyCluster \  
  --service MyService \  
  --task-sets arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-  
svc/1234567890123456789
```

Ausgabe:

```
{  
  "taskSets": [  
    {  
      "id": "ecs-svc/1234567890123456789",  
      "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
      "status": "ACTIVE",  
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
      "computedDesiredCount": 0,  
      "pendingCount": 0,  
      "runningCount": 0,  
      "createdAt": 1557207715.195,  
      "updatedAt": 1557207740.014,  
      "launchType": "EC2",  
      "networkConfiguration": {  
        "awsvpcConfiguration": {  
          "subnets": [  
            "subnet-12344321"  
          ],  
          "securityGroups": [  
            "sg-1234431"  
          ],  
          "assignPublicIp": "DISABLED"  
        }  
      },  
      "loadBalancers": [],  
      "serviceRegistries": [],
```

```

        "scale": {
            "value": 0.0,
            "unit": "PERCENT"
        },
        "stabilityStatus": "STEADY_STATE",
        "stabilityStatusAt": 1557207740.014
    }
],
"failures": []
}

```

- API-Einheiten finden Sie unter [DescribeTaskSets AWS CLI-Befehlsreferenz](#).

describe-tasks

Das folgende Codebeispiel zeigt die Verwendung von `describe-tasks`.

AWS CLI

Beispiel 1: Um eine einzelne Aufgabe zu beschreiben (Aufgaben)

Im folgenden `describe-tasks` Beispiel werden die Details einer Aufgabe in einem Cluster abgerufen. Sie können die Aufgabe angeben, indem Sie entweder die ID oder den vollständigen Teil ARN der Aufgabe verwenden. In diesem Beispiel wird der gesamte ARN Teil der Aufgabe verwendet.

```

aws ecs describe-tasks \
  --cluster MyCluster \
  --tasks arn:aws:ecs:us-east-1:123456789012:task/MyCluster/4d590253bb114126b7afa7b58EXAMPLE

```

Ausgabe:

```

{
  "tasks": [
    {
      "attachments": [],
      "attributes": [
        {
          "name": "ecs.cpu-architecture",
          "value": "x86_64"
        }
      ]
    }
  ]
}

```

```
    ],
    "availabilityZone": "us-east-1b",
    "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
    "connectivity": "CONNECTED",
    "connectivityAt": "2021-08-11T12:21:26.681000-04:00",
    "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/test/025c7e2c5e054a6790a29fc1fEXAMPLE",
    "containers": [
      {
        "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/4d590253bb114126b7afa7b58eea9221/a992d1cc-ea46-474a-b6e8-24688EXAMPLE",
        "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE",
        "name": "simple-app",
        "image": "httpd:2.4",
        "runtimeId":
"91251eed27db90006ad67b1a08187290869f216557717dd5c39b37c94EXAMPLE",
        "lastStatus": "RUNNING",
        "networkBindings": [
          {
            "bindIP": "0.0.0.0",
            "containerPort": 80,
            "hostPort": 80,
            "protocol": "tcp"
          }
        ],
        "networkInterfaces": [],
        "healthStatus": "UNKNOWN",
        "cpu": "10",
        "memory": "300"
      }
    ],
    "cpu": "10",
    "createdAt": "2021-08-11T12:21:26.681000-04:00",
    "desiredStatus": "RUNNING",
    "enableExecuteCommand": false,
    "group": "service:testupdate",
    "healthStatus": "UNKNOWN",
    "lastStatus": "RUNNING",
    "launchType": "EC2",
    "memory": "300",
    "overrides": {
      "containerOverrides": [
        {
```

```

        "name": "simple-app"
      }
    ],
    "inferenceAcceleratorOverrides": [],
  },
  "pullStartedAt": "2021-08-11T12:21:28.234000-04:00",
  "pullStoppedAt": "2021-08-11T12:21:33.793000-04:00",
  "startedAt": "2021-08-11T12:21:34.945000-04:00",
  "startedBy": "ecs-svc/968695068243EXAMPLE",
  "tags": [],
  "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/4d590253bb114126b7afa7b58eea9221",
  "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/console-sample-app-static2:1",
  "version": 2
}
],
"failures": []
}

```

Weitere Informationen finden Sie unter [ECS Amazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

Beispiel 2: Um mehrere Aufgaben zu beschreiben

Im folgenden `describe-tasks` Beispiel werden die Details mehrerer Aufgaben in einem Cluster abgerufen. Sie können die Aufgabe angeben, indem Sie entweder die ID oder den vollständigen Teil ARN der Aufgabe verwenden. In diesem Beispiel werden alle IDs Aufgaben verwendet.

```

aws ecs describe-tasks \
  --cluster MyCluster \
  --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"

```

Ausgabe:

```

{
  "tasks": [
    {
      "attachments": [
        {
          "id": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
          "type": "ElasticNetworkInterface",
          "status": "ATTACHED",

```

```
    "details": [
      {
        "name": "subnetId",
        "value": "subnet-0d0eab1bb3EXAMPLE"
      },
      {
        "name": "networkInterfaceId",
        "value": "eni-0fa40520aeEXAMPLE"
      },
      {
        "name": "macAddress",
        "value": "0e:89:76:28:07:b3"
      },
      {
        "name": "privateDnsName",
        "value": "ip-10-0-1-184.ec2.internal"
      },
      {
        "name": "privateIPv4Address",
        "value": "10.0.1.184"
      }
    ]
  },
  "attributes": [
    {
      "name": "ecs.cpu-architecture",
      "value": "x86_64"
    }
  ],
  "availabilityZone": "us-east-1b",
  "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
  "connectivity": "CONNECTED",
  "connectivityAt": "2021-12-20T12:13:37.875000-05:00",
  "containers": [
    {
      "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/MyCluster/74de0355a10a4f979ac495c14EXAMPLE/aad3ba00-83b3-4dac-84d4-11f8cEXAMPLE",
      "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
      "name": "web",
      "image": "nginx",
      "runtimeId": "74de0355a10a4f979ac495c14EXAMPLE-265927825",
      "lastStatus": "RUNNING",
```

```
        "networkBindings": [],
        "networkInterfaces": [
            {
                "attachmentId": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
                "privateIpv4Address": "10.0.1.184"
            }
        ],
        "healthStatus": "UNKNOWN",
        "cpu": "99",
        "memory": "100"
    }
],
"cpu": "256",
"createdAt": "2021-12-20T12:13:20.226000-05:00",
"desiredStatus": "RUNNING",
"enableExecuteCommand": false,
"group": "service:tdsevicetag",
"healthStatus": "UNKNOWN",
"lastStatus": "RUNNING",
"launchType": "FARGATE",
"memory": "512",
"overrides": {
    "containerOverrides": [
        {
            "name": "web"
        }
    ],
    "inferenceAcceleratorOverrides": []
},
"platformVersion": "1.4.0",
"platformFamily": "Linux",
"pullStartedAt": "2021-12-20T12:13:42.665000-05:00",
"pullStoppedAt": "2021-12-20T12:13:46.543000-05:00",
"startedAt": "2021-12-20T12:13:48.086000-05:00",
"startedBy": "ecs-svc/988401040018EXAMPLE",
"tags": [],
"taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
"taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
"version": 3,
"ephemeralStorage": {
    "sizeInGiB": 20
}
}
```

```
  },
  {
    "attachments": [
      {
        "id": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
        "type": "ElasticNetworkInterface",
        "status": "ATTACHED",
        "details": [
          {
            "name": "subnetId",
            "value": "subnet-0d0eab1bb3EXAMPLE"
          },
          {
            "name": "networkInterfaceId",
            "value": "eni-064c7766daEXAMPLE"
          },
          {
            "name": "macAddress",
            "value": "0e:76:83:01:17:a9"
          },
          {
            "name": "privateDnsName",
            "value": "ip-10-0-1-41.ec2.internal"
          },
          {
            "name": "privateIPv4Address",
            "value": "10.0.1.41"
          }
        ]
      }
    ],
    "attributes": [
      {
        "name": "ecs.cpu-architecture",
        "value": "x86_64"
      }
    ],
    "availabilityZone": "us-east-1b",
    "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
    "connectivity": "CONNECTED",
    "connectivityAt": "2021-12-20T12:13:35.243000-05:00",
    "containers": [
      {
```



```
        "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/d789e94343414c25b9f6bd59eEXAMPLE/9afef792-609b-43a5-bb6a-3efdbEXAMPLE",
        "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
        "name": "web",
        "image": "nginx",
        "runtimeId": "d789e94343414c25b9f6bd59eEXAMPLE-265927825",
        "lastStatus": "RUNNING",
        "networkBindings": [],
        "networkInterfaces": [
            {
                "attachmentId": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
                "privateIpv4Address": "10.0.1.41"
            }
        ],
        "healthStatus": "UNKNOWN",
        "cpu": "99",
        "memory": "100"
    }
],
"cpu": "256",
"createdAt": "2021-12-20T12:13:20.226000-05:00",
"desiredStatus": "RUNNING",
"enableExecuteCommand": false,
"group": "service:tdsevicetag",
"healthStatus": "UNKNOWN",
"lastStatus": "RUNNING",
"launchType": "FARGATE",
"memory": "512",
"overrides": {
    "containerOverrides": [
        {
            "name": "web"
        }
    ],
    "inferenceAcceleratorOverrides": []
},
"platformVersion": "1.4.0",
"platformFamily": "Linux",
"pullStartedAt": "2021-12-20T12:13:44.611000-05:00",
"pullStoppedAt": "2021-12-20T12:13:48.251000-05:00",
"startedAt": "2021-12-20T12:13:49.326000-05:00",
"startedBy": "ecs-svc/988401040018EXAMPLE",
"tags": [],
```

```

        "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
        "version": 3,
        "ephemeralStorage": {
            "sizeInGiB": 20
        }
    },
    "failures": []
}

```

Weitere Informationen finden Sie unter [ECSAmazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [DescribeTasks](#) in der AWS CLI Befehlsreferenz.

execute-command

Das folgende Codebeispiel zeigt die Verwendung `execute-command`.

AWS CLI

Um einen interaktiven `/bin/sh` Befehl auszuführen

Im folgenden `execute-command` Beispiel wird ein interaktiver `/bin/sh` Befehl `MyContainer` für einen Container ausgeführt, der nach einer Aufgabe mit der ID `arn:aws:ecs:us-east-1:123456789012:task/MyCluster/d789e94343414c25b9f6bd59eEXAMPLE`.

```

aws ecs execute-command \
  --cluster MyCluster \
  --task arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE \
  --container MyContainer \
  --interactive \
  --command "/bin/sh"

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von Amazon ECS Exec zum Debuggen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie unter [ExecuteCommand AWS CLI Befehlsreferenz](#).

list-account-settings

Das folgende Codebeispiel zeigt die Verwendung `list-account-settings`.

AWS CLI

Beispiel 1: Um die Kontoeinstellungen für ein Konto anzuzeigen

Das folgende `list-account-settings` Beispiel zeigt die effektiven Kontoeinstellungen für ein Konto.

```
aws ecs list-account-settings --effective-settings
```

Ausgabe:

```
{
  "settings": [
    {
      "name": "containerInstanceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    },
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    },
    {
      "name": "taskLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    }
  ]
}
```

Beispiel 2: Um die Kontoeinstellungen für einen bestimmten IAM Benutzer oder eine bestimmte IAM Rolle anzuzeigen

Im folgenden `list-account-settings` Beispiel werden die Kontoeinstellungen für den angegebenen IAM Benutzer oder die angegebene IAM Rolle angezeigt.

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Ausgabe:

```
{
  "settings": [
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon Resource Names \(ARNs\) und IDs](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListAccountSettings](#) in der AWS CLI Befehlsreferenz.

list-attributes

Das folgende Codebeispiel zeigt die Verwendung `list-attributes`.

AWS CLI

Um die Container-Instances aufzulisten, die ein bestimmtes Attribut enthalten

Im folgenden Beispiel werden die Attribute für Container-Instances aufgeführt, die das `stack=production` Attribut im Standardcluster haben.

```
aws ecs list-attributes \
  --target-type container-instance \
  --attribute-name stack \
  --attribute-value production \
  --cluster default
```

Ausgabe:

```
{
  "attributes": [
    {
```

```
        "name": "stack",
        "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
        "value": "production"
    }
]
}
```

Weitere Informationen finden Sie unter [Amazon ECS Container Agent Configuration](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListAttributes](#) in der AWS CLI Befehlsreferenz.

list-clusters

Das folgende Codebeispiel zeigt die Verwendung `list-clusters`.

AWS CLI

Um Ihre verfügbaren Cluster aufzulisten

Das folgende `list-clusters` Beispiel listet alle verfügbaren Cluster auf.

```
aws ecs list-clusters
```

Ausgabe:

```
{
  "clusterArns": [
    "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
    "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
  ]
}
```

Weitere Informationen finden Sie unter [Amazon ECS Clusters](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListClusters](#) in der AWS CLI Befehlsreferenz.

list-container-instances

Das folgende Codebeispiel zeigt die Verwendung `list-container-instances`.

AWS CLI

Um die Container-Instances in einem Cluster aufzulisten

Das folgende `list-container-instances` Beispiel listet alle verfügbaren Container-Instances in einem Cluster auf.

```
aws ecs list-container-instances --cluster MyCluster
```

Ausgabe:

```
{
  "containerInstanceArns": [
    "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
  ]
}
```

Weitere Informationen finden Sie unter [Amazon ECS Container Instances](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListContainerInstances](#) in der AWS CLI Befehlsreferenz.

list-services-by-namespace

Das folgende Codebeispiel zeigt die Verwendung `list-services-by-namespace`.

AWS CLI

Um die Dienste in einem Namespace aufzulisten

Das folgende `list-services-by-namespace` Beispiel listet alle Dienste auf, die für den angegebenen Namespace in Ihrer Standardregion konfiguriert sind.

```
aws ecs list-services-by-namespace \
  --namespace service-connect
```

Ausgabe:

```
{
```

```
"serviceArns": [
  "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",
  "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-
service"
]
```

Weitere Informationen finden Sie unter [Service Connect](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListServicesByNamespace](#) in der AWS CLI Befehlsreferenz.

list-services

Das folgende Codebeispiel zeigt die Verwendung `list-services`.

AWS CLI

Um die Dienste in einem Cluster aufzulisten

Das folgende `list-services` Beispiel zeigt, wie die Dienste aufgelistet werden, die in einem Cluster ausgeführt werden.

```
aws ecs list-services --cluster MyCluster
```

Ausgabe:

```
{
  "serviceArns": [
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"
  ]
}
```

Weitere Informationen finden Sie unter [Services](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListServices](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für einen bestimmten Cluster auf.

```
aws ecs list-tags-for-resource \
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

Ausgabe:

```
{
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value2"
    },
    {
      "key": "key3",
      "value": "value3"
    }
  ]
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-task-definition-families

Das folgende Codebeispiel zeigt die Verwendung `list-task-definition-families`.

AWS CLI

Beispiel 1: Um die registrierten Aufgabendefinitionsfamilien aufzulisten

Im folgenden `list-task-definition-families` Beispiel werden alle registrierten Aufgabendefinitionsfamilien aufgeführt.

```
aws ecs list-task-definition-families
```

Ausgabe:


```
{
  "families": [
    "node-js-app",
    "web-timer",
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

Beispiel 2: Um die registrierten Aufgabendefinitionsfamilien zu filtern

Im folgenden `list-task-definition-families` Beispiel werden die Versionen der Aufgabendefinitionen aufgeführt, die mit „hpcc“ beginnen.

```
aws ecs list-task-definition-families --family-prefix hpcc
```

Ausgabe:

```
{
  "families": [
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

Weitere Informationen finden Sie unter [Aufgabendefinitionsparameter](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListTaskDefinitionFamilies](#) in der AWS CLI Befehlsreferenz.

list-task-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-task-definitions`.

AWS CLI

Beispiel 1: Um die registrierten Aufgabendefinitionen aufzulisten

Das folgende `list-task-definitions` Beispiel listet alle registrierten Aufgabendefinitionen auf.

```
aws ecs list-task-definitions
```

Ausgabe:

```
{
  "taskDefinitionArns": [
    "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep300:2",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep360:1",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
  ]
}
```

Beispiel 2: Um die registrierten Aufgabendefinitionen in einer Familie aufzulisten

Im folgenden list-task-definitions Beispiel werden die Versionen der Aufgabendefinitionen einer angegebenen Familie aufgeführt.

```
aws ecs list-task-definitions --family-prefix wordpress
```

Ausgabe:

```
{
  "taskDefinitionArns": [
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
  ]
}
```

Weitere Informationen finden Sie unter [ECSAmazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListTaskDefinitions](#) in der AWS CLI Befehlsreferenz.

list-tasks

Das folgende Codebeispiel zeigt die Verwendung list-tasks.

AWS CLI

Beispiel 1: Um die Aufgaben in einem Cluster aufzulisten

Das folgende `list-tasks` Beispiel listet alle Aufgaben in einem Cluster auf.

```
aws ecs list-tasks --cluster default
```

Ausgabe:

```
{
  "taskArns": [
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-11111EXAMPLE",
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE"
  ]
}
```

Beispiel 2: Um die Aufgaben auf einer bestimmten Container-Instance aufzulisten

Das folgende `list-tasks` Beispiel listet die Aufgaben auf einer Container-Instance auf, wobei die Container-Instance UUID als Filter verwendet wird.

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

Ausgabe:

```
{
  "taskArns": [
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE"
  ]
}
```

Weitere Informationen finden Sie unter [ECSAmazon-Aufgabendefinitionen](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [ListTasks](#) in der AWS CLI Befehlsreferenz.

put-account-setting-default

Das folgende Codebeispiel zeigt die Verwendung `put-account-setting-default`.

AWS CLI

Um die Standardkontoeinstellungen zu ändern

Im folgenden `put-account-setting-default` Beispiel wird die Standardkontoeinstellung für alle IAM Benutzer oder Rollen in Ihrem Konto geändert. Diese Änderungen gelten für das gesamte AWS Konto, sofern ein IAM Benutzer oder eine Rolle diese Einstellungen nicht ausdrücklich für sich selbst außer Kraft setzt.

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

Ausgabe:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:root"
  }
}
```

Weitere Informationen finden Sie unter [Amazon Resource Names \(ARNs\) und IDs](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [PutAccountSettingDefault](#) in der AWS CLI Befehlsreferenz.

put-account-setting

Das folgende Codebeispiel zeigt die Verwendung `put-account-setting`.

AWS CLI

Um die Kontoeinstellungen für Ihr IAM Benutzerkonto zu ändern

Das folgende `put-account-setting` Beispiel aktiviert die `serviceLongArnFormat` Kontoeinstellungen für Ihr IAM Benutzerkonto.

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled
```

Ausgabe:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::130757420319:user/your_username"
  }
}
```

Weitere Informationen finden Sie unter [Kontoeinstellungen ändern](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [PutAccountSetting](#) in der AWS CLI Befehlsreferenz.

put-account-settings

Das folgende Codebeispiel zeigt die Verwendung `put-account-settings`.

AWS CLI

Um die Kontoeinstellungen für einen IAM Benutzer oder eine IAM Rolle zu ändern

Im folgenden `put-account-setting` Beispiel werden die Kontoeinstellungen für den angegebenen IAM Benutzer oder die angegebene IAM Rolle geändert.

```
aws ecs put-account-setting \
  --name serviceLongArnFormat \
  --value enabled \
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Ausgabe:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"
  }
}
```

- API-Einheiten finden Sie unter [PutAccountSettings AWS CLI](#) Befehlsreferenz.

put-attributes

Das folgende Codebeispiel zeigt die Verwendung `put-attributes`.

AWS CLI

Um ein Attribut zu erstellen und es mit einer ECS Amazon-Ressource zu verknüpfen

Im Folgenden wird ein Attribut mit dem Namen `Stack` und dem Wert `Production` auf eine Container-Instance `put-attributes` angewendet.

```
aws ecs put-attributes \  
  --attributes name=stack,value=production,targetId=arn:aws:ecs:us-  
west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Ausgabe:

```
{  
  "attributes": [  
    {  
      "name": "stack",  
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",  
      "value": "production"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [PutAttributes](#) unter AWS CLI Befehlsreferenz.

put-cluster-capacity-providers

Das folgende Codebeispiel zeigt die Verwendung `put-cluster-capacity-providers`.

AWS CLI

Beispiel 1: Um einen vorhandenen Kapazitätsanbieter zu einem Cluster hinzuzufügen

Im folgenden `put-cluster-capacity-providers` Beispiel wird einem Cluster ein vorhandener Kapazitätsanbieter hinzugefügt. Der `create-capacity-provider` Befehl wird verwendet, um einen Kapazitätsanbieter zu erstellen. Der `describe-clusters`

Befehl wird verwendet, um die aktuellen Kapazitätsanbieter und die mit einem Cluster verknüpfte Standardstrategie für Kapazitätsanbieter zu beschreiben. Wenn Sie einem Cluster einen neuen Kapazitätsanbieter hinzufügen, müssen Sie zusätzlich zu dem neuen Kapazitätsanbieter, den Sie dem Cluster zuordnen möchten, alle vorhandenen Kapazitätsanbieter angeben. Sie müssen auch die Standardstrategie für den Kapazitätsanbieter angeben, der dem Cluster zugeordnet werden soll. In diesem Beispiel ist dem `MyCluster` Cluster der `MyCapacityProvider1` Kapazitätsanbieter zugeordnet, und Sie möchten den `MyCapacityProvider2` Kapazitätsanbieter hinzufügen und ihn in die Standardstrategie für den Kapazitätsanbieter aufnehmen, sodass die Aufgaben gleichmäßig auf beide Kapazitätsanbieter verteilt werden.

```
aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
  --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

Ausgabe:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [
      "MyCapacityProvider1",
      "MyCapacityProvider2"
    ],
    "defaultCapacityProviderStrategy": [
```

```
    {
      "capacityProvider": "MyCapacityProvider1",
      "weight": 1,
      "base": 0
    },
    {
      "capacityProvider": "MyCapacityProvider2",
      "weight": 1,
      "base": 0
    }
  ],
  "attachments": [
    {
      "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
      "type": "as_policy",
      "status": "ACTIVE",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider1"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
        }
      ]
    },
    {
      "id": "ae592060-2382-4663-9476-b015c685593c",
      "type": "as_policy",
      "status": "ACTIVE",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider2"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
        }
      ]
    }
  ]
}
```



```

    ],
    "attachmentsStatus": "UPDATE_IN_PROGRESS"
  }
}

```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

Beispiel 2: Um einen Kapazitätsanbieter aus einem Cluster zu entfernen

Im folgenden `put-cluster-capacity-providers` Beispiel wird ein Kapazitätsanbieter aus einem Cluster entfernt. Der `describe-clusters` Befehl wird verwendet, um die aktuellen Kapazitätsanbieter zu beschreiben, die einem Cluster zugeordnet sind. Wenn Sie einen Kapazitätsanbieter aus einem Cluster entfernen, müssen Sie die Kapazitätsanbieter angeben, die dem Cluster zugeordnet bleiben sollen, sowie die Standardstrategie für den Kapazitätsanbieter, die dem Cluster zugeordnet werden sollen. In diesem Beispiel sind dem Cluster die `MyCapacityProvider1` und die `MyCapacityProvider2` Kapazitätsanbieter zugeordnet, und Sie möchten den `MyCapacityProvider2` Kapazitätsanbieter entfernen, sodass Sie ihn nur `MyCapacityProvider1` im Befehl zusammen mit der aktualisierten Standardstrategie für Kapazitätsanbieter angeben.

```

aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers MyCapacityProvider1 \
  --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0

```

Ausgabe:

```

{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [

```

```
    {
      "name": "containerInsights",
      "value": "enabled"
    }
  ],
  "capacityProviders": [
    "MyCapacityProvider1"
  ],
  "defaultCapacityProviderStrategy": [
    {
      "capacityProvider": "MyCapacityProvider1",
      "weight": 1,
      "base": 0
    }
  ],
  "attachments": [
    {
      "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
      "type": "as_policy",
      "status": "ACTIVE",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider1"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
        }
      ]
    },
    {
      "id": "ae592060-2382-4663-9476-b015c685593c",
      "type": "as_policy",
      "status": "DELETING",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider2"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
        }
      ]
    }
  ]
}
```

```

    ]
  }
],
"attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

Beispiel 3: Um alle Kapazitätsanbieter aus einem Cluster zu entfernen

Im folgenden `put-cluster-capacity-providers` Beispiel werden alle vorhandenen Kapazitätsanbieter aus dem Cluster entfernt.

```

aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers [] \
  --default-capacity-provider-strategy []

```

Ausgabe:

```

{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [],
    "defaultCapacityProviderStrategy": [],
    "attachments": [

```

```

    {
      "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
      "type": "as_policy",
      "status": "DELETING",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider1"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE111111"
        }
      ]
    },
    {
      "id": "ae592060-2382-4663-9476-b015c685593c",
      "type": "as_policy",
      "status": "DELETING",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider2"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE222222"
        }
      ]
    }
  ],
  "attachmentsStatus": "UPDATE_IN_PROGRESS"
}

```

Weitere Informationen finden Sie unter [Cluster-Kapazitätsanbieter](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [PutClusterCapacityProviders](#) in der AWS CLI Befehlsreferenz.

register-task-definition

Das folgende Codebeispiel zeigt die Verwendung `register-task-definition`.

AWS CLI

Beispiel 1: Um eine Aufgabendefinition in einer JSON Datei zu registrieren

Im folgenden `register-task-definition` Beispiel wird eine Aufgabendefinition für die angegebene Familie registriert. Die Containerdefinitionen werden im JSON Format am angegebenen Dateispeicherort gespeichert.

```
aws ecs register-task-definition \  
  --cli-input-json file://<path_to_json_file>/sleep360.json
```

Inhalt von `sleep360.json`:

```
{  
  "containerDefinitions": [  
    {  
      "name": "sleep",  
      "image": "busybox",  
      "cpu": 10,  
      "command": [  
        "sleep",  
        "360"  
      ],  
      "memory": 10,  
      "essential": true  
    }  
  ],  
  "family": "sleep360"  
}
```

Ausgabe:

```
{  
  "taskDefinition": {  
    "status": "ACTIVE",  
    "family": "sleep360",  
    "placementConstraints": [],  
    "compatibilities": [  

```

```

        "EXTERNAL",
        "EC2"
    ],
    "volumes": [],
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
sleep360:1",
    "containerDefinitions": [
        {
            "environment": [],
            "name": "sleep",
            "mountPoints": [],
            "image": "busybox",
            "cpu": 10,
            "portMappings": [],
            "command": [
                "sleep",
                "360"
            ],
            "memory": 10,
            "essential": true,
            "volumesFrom": []
        }
    ],
    "revision": 1
}

```

Weitere Informationen finden Sie unter [Beispielaufgabendefinitionen](#) im Amazon ECS Developer Guide.

Beispiel 2: Um eine Aufgabendefinition mit einem JSON Zeichenkettenparameter zu registrieren

Im folgenden `register-task-definition` Beispiel wird eine Aufgabendefinition mithilfe von Containerdefinitionen registriert, die als JSON Zeichenkettenparameter mit maskierten doppelten Anführungszeichen bereitgestellt werden.

```

aws ecs register-task-definition \
  --family sleep360 \
  --container-definitions "[{\\"name\\":\\"sleep\\",\\"image\\":\\"busybox\\",\\"cpu\\":10,
\\\"command\\":[\\"sleep\\",\\"360\\"],\\"memory\\":10,\\"essential\\":true}]"

```

Die Ausgabe ist identisch mit dem vorherigen Beispiel.

Weitere Informationen finden Sie unter [Erstellen einer Aufgabendefinition](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [RegisterTaskDefinition](#) in der AWS CLI Befehlsreferenz.

run-task

Das folgende Codebeispiel zeigt die Verwendung `run-task`.

AWS CLI

Um eine Aufgabe auf Ihrem Standardcluster auszuführen

Im folgenden `run-task` Beispiel wird eine Aufgabe auf dem Standardcluster ausgeführt und ein Client-Token verwendet.

```
aws ecs run-task \  
  --cluster default \  
  --task-definition sleep360:1 \  
  --client-token 550e8400-e29b-41d4-a716-446655440000
```

Ausgabe:

```
{  
  "tasks": [  
    {  
      "attachments": [],  
      "attributes": [  
        {  
          "name": "ecs.cpu-architecture",  
          "value": "x86_64"  
        }  
      ],  
      "availabilityZone": "us-east-1b",  
      "capacityProviderName": "example-capacity-provider",  
      "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/default",  
      "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-instance/default/bc4d2ec611d04bb7bb97e83ceEXAMPLE",  
      "containers": [  
        {  
          "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/default/d6f51cc5bbc94a47969c92035e9f66f8/75853d2d-711e-458a-8362-0f0aEXAMPLE",
```

```
        "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
        "name": "sleep",
        "image": "busybox",
        "lastStatus": "PENDING",
        "networkInterfaces": [],
        "cpu": "10",
        "memory": "10"
    }
],
"cpu": "10",
"createdAt": "2023-11-21T16:59:34.403000-05:00",
"desiredStatus": "RUNNING",
"enableExecuteCommand": false,
"group": "family:sleep360",
"lastStatus": "PENDING",
"launchType": "EC2",
"memory": "10",
"overrides": {
    "containerOverrides": [
        {
            "name": "sleep"
        }
    ],
    "inferenceAcceleratorOverrides": []
},
"tags": [],
"taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
"taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/sleep360:1",
"version": 1
}
],
"failures": []
}
```

Weitere Informationen finden Sie unter [Running Tasks](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [RunTask](#) in der AWS CLI Befehlsreferenz.

start-task

Das folgende Codebeispiel zeigt die Verwendung `start-task`.

AWS CLI

Um eine neue Aufgabe zu starten

Im Folgenden `start-task` wird eine Aufgabe mit der neuesten Version der `sleep360` Aufgabendefinition auf der angegebenen Container-Instance im Standardcluster gestartet.

```
aws ecs start-task \  
  --task-definition sleep360 \  
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

Ausgabe:

```
{  
  "tasks": [  
    {  
      "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/  
default/666fdccc2e2d4b6894dd422f4eeee8f8",  
      "clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",  
      "taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-  
definition/sleep360:3",  
      "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/default/765936fadbdd46b5991a4bd70c2a43d4",  
      "overrides": {  
        "containerOverrides": [  
          {  
            "name": "sleep"  
          }  
        ]  
      },  
      "lastStatus": "PENDING",  
      "desiredStatus": "RUNNING",  
      "cpu": "128",  
      "memory": "128",  
      "containers": [  
        {  
          "containerArn": "arn:aws:ecs:us-  
west-2:130757420319:container/75f11ed4-8a3d-4f26-a33b-ad1db9e02d41",
```

```

        "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
        "name": "sleep",
        "lastStatus": "PENDING",
        "networkInterfaces": [],
        "cpu": "10",
        "memory": "10"
    }
],
"version": 1,
"createdAt": 1563421494.186,
"group": "family:sleep360",
"launchType": "EC2",
"attachments": [],
"tags": []
}
],
"failures": []
}

```

- API-Einheiten finden Sie [StartTask](#) unter AWS CLI Befehlsreferenz.

stop-task

Das folgende Codebeispiel zeigt die Verwendung `stop-task`.

AWS CLI

So beenden Sie eine Aufgabe

Im Folgenden `stop-task` wird verhindert, dass die angegebene Aufgabe im Standardcluster ausgeführt wird.

```

aws ecs stop-task \
  --task 666fdccc2e2d4b6894dd422f4eeee8f8

```

Ausgabe:

```

{
  "task": {
    "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",

```

```
"clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",
"taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-definition/
sleep360:3",
"containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
"overrides": {
  "containerOverrides": []
},
"lastStatus": "STOPPED",
"desiredStatus": "STOPPED",
"cpu": "128",
"memory": "128",
"containers": [],
"version": 2,
"stoppedReason": "Taskfailedtostart",
"stopCode": "TaskFailedToStart",
"connectivity": "CONNECTED",
"connectivityAt": 1563421494.186,
"pullStartedAt": 1563421494.252,
"pullStoppedAt": 1563421496.252,
"executionStoppedAt": 1563421497,
"createdAt": 1563421494.186,
"stoppingAt": 1563421497.252,
"stoppedAt": 1563421497.252,
"group": "family:sleep360",
"launchType": "EC2",
"attachments": [],
"tags": []
}
}
```

- API-Einheiten finden Sie [StopTask](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird der angegebenen Ressource ein einzelnes Tag hinzugefügt.

```
aws ecs tag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tags key=key1,value=value1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Um einer Ressource mehrere Tags hinzuzufügen

Das folgende `tag-resource` Beispiel fügt der angegebenen Ressource mehrere Tags hinzu.

```
aws ecs tag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden die aufgelisteten Tags aus der angegebenen Ressource entfernt.

```
aws ecs untag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tag-keys key1,key2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-cluster-settings

Das folgende Codebeispiel zeigt die Verwendung `update-cluster-settings`.

AWS CLI

Um die Einstellungen für Ihren Cluster zu ändern

Das folgende `update-cluster-settings` Beispiel aktiviert CloudWatch Container Insights für den `default` Cluster.

```
aws ecs update-cluster-settings \  
  --cluster default \  
  --settings name=containerInsights,value=enabled
```

Ausgabe:

```
{  
  "cluster": {  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "clusterName": "default",  
    "status": "ACTIVE",  
    "registeredContainerInstancesCount": 0,  
    "runningTasksCount": 0,  
    "pendingTasksCount": 0,  
    "activeServicesCount": 0,  
    "statistics": [],  
    "tags": [],  
    "settings": [  
      {  
        "name": "containerInsights",  
        "value": "enabled"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Kontoeinstellungen ändern](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [UpdateClusterSettings](#) in der AWS CLI Befehlsreferenz.

update-container-agent

Das folgende Codebeispiel zeigt die Verwendung `update-container-agent`.

AWS CLI

Um den Container-Agenten auf einer ECS Amazon-Container-Instance zu aktualisieren

Im folgenden `update-container-agent` Beispiel wird der Container-Agent auf der angegebenen Container-Instance im Standard-Cluster aktualisiert.

```
aws ecs update-container-agent --cluster default --container-  
instance a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{  
  "containerInstance": {  
    "status": "ACTIVE",  
    ...  
    "agentUpdateStatus": "PENDING",  
    "versionInfo": {  
      "agentVersion": "1.0.0",  
      "agentHash": "4023248",  
      "dockerVersion": "DockerVersion: 1.5.0"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Updating the Amazon ECS Container Agent](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [UpdateContainerAgent](#) in der AWS CLI Befehlsreferenz.

update-container-instances-state

Das folgende Codebeispiel zeigt die Verwendung `update-container-instances-state`.

AWS CLI

Um den Status einer Container-Instance zu aktualisieren

Im Folgenden wird der Status der angegebenen Container-Instance `update-container-instances-state` aktualisiert, `DRAINING` wodurch sie aus dem Cluster entfernt wird, in dem sie registriert ist.

```
aws ecs update-container-instances-state \  
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \  
  --status DRAINING
```

Ausgabe:

```
{  
  "containerInstances": [  
    {  
      "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/default/765936fadbdd46b5991a4bd70c2a43d4",  
      "ec2InstanceId": "i-013d87ffbb4d513bf",  
      "version": 4390,  
      "versionInfo": {  
        "agentVersion": "1.29.0",  
        "agentHash": "a190a73f",  
        "dockerVersion": "DockerVersion:18.06.1-ce"  
      },  
      "remainingResources": [  
        {  
          "name": "CPU",  
          "type": "INTEGER",  
          "doubleValue": 0,  
          "longValue": 0,  
          "integerValue": 1536  
        },  
        {  
          "name": "MEMORY",  
          "type": "INTEGER",  
          "doubleValue": 0,  
          "longValue": 0,  
          "integerValue": 2681  
        },  
        {  
          "name": "PORTS",  
          "type": "STRINGSET",  
          "doubleValue": 0,  
          "longValue": 0,  
          "integerValue": 0,  
          "stringSetValue": [  
            "22",  
            "2376",  
            "2375",  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```
        "51678",
        "51679"
    ]
},
{
    "name": "PORTS_UDP",
    "type": "STRINGSET",
    "doubleValue": 0,
    "longValue": 0,
    "integerValue": 0,
    "stringSetValue": []
}
],
"registeredResources": [
    {
        "name": "CPU",
        "type": "INTEGER",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 2048
    },
    {
        "name": "MEMORY",
        "type": "INTEGER",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 3705
    },
    {
        "name": "PORTS",
        "type": "STRINGSET",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 0,
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
        ]
    }
},
{
    "name": "PORTS_UDP",
```



```
        "type": "STRINGSET",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 0,
        "stringSetValue": []
    }
],
"status": "DRAINING",
"agentConnected": true,
"runningTasksCount": 2,
"pendingTasksCount": 0,
"attributes": [
    {
        "name": "ecs.capability.secrets.asm.environment-variables"
    },
    {
        "name": "ecs.capability.branch-cni-plugin-version",
        "value": "e0703516-"
    },
    {
        "name": "ecs.ami-id",
        "value": "ami-00e0090ac21971297"
    },
    {
        "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
    },
    {
        "name": "com.amazonaws.ecs.capability.logging-driver.none"
    },
    {
        "name": "ecs.capability.ecr-endpoint"
    },
    {
        "name": "ecs.capability.docker-plugin.local"
    },
    {
        "name": "ecs.capability.task-cpu-mem-limit"
    },
    {
        "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
    },
    {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
    },
    ],
```

```
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
},
{
  "name": "ecs.availability-zone",
  "value": "us-west-2c"
},
{
  "name": "ecs.capability.aws-appmesh"
},
{
  "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
},
{
  "name": "ecs.capability.task-eni-trunking"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
},
{
  "name": "com.amazonaws.ecs.capability.privileged-container"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
},
{
  "name": "ecs.cpu-architecture",
  "value": "x86_64"
},
},
```

```
{
  "name": "com.amazonaws.ecs.capability.ecr-auth"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
},
{
  "name": "ecs.os-type",
  "value": "linux"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
},
{
  "name": "ecs.capability.task-eia"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
},
{
  "name": "ecs.capability.private-registry-
authentication.secretsmanager"
},
{
  "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
},
{
  "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
},
{
  "name": "ecs.capability.execution-role-awslogs"
},
{
  "name": "ecs.vpc-id",
  "value": "vpc-1234"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
```

```
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
    },
    {
      "name": "ecs.capability.task-eni"
    },
    {
      "name": "ecs.capability.execution-role-ecr-pull"
    },
    {
      "name": "ecs.capability.container-health-check"
    },
    {
      "name": "ecs.subnet-id",
      "value": "subnet-1234"
    },
    {
      "name": "ecs.instance-type",
      "value": "c5.large"
    },
    {
      "name": "com.amazonaws.ecs.capability.task-iam-role-network-
host"
    },
    {
      "name": "ecs.capability.container-ordering"
    },
    {
      "name": "ecs.capability.cni-plugin-version",
      "value": "91ccefc8-2019.06.0"
    },
    {
      "name": "ecs.capability.pid-ipc-namespace-sharing"
    },
    {
      "name": "ecs.capability.secrets.ssm.environment-variables"
    },
    {
      "name": "com.amazonaws.ecs.capability.task-iam-role"
    }
  ],
  "registeredAt": 1560788724.507,
  "attachments": [],
```

```

        "tags": []
      }
    ],
    "failures": []
  }

```

- API-Einheiten finden Sie [UpdateContainerInstancesState](#) in der AWS CLI Befehlsreferenz.

update-service-primary-task-set

Das folgende Codebeispiel zeigt die Verwendung `update-service-primary-task-set`.

AWS CLI

Um den primären Tasksatz für einen Service zu aktualisieren

Im folgenden `update-service-primary-task-set` Beispiel wird der primäre Tasksatz für den angegebenen Service aktualisiert.

```

aws ecs update-service-primary-task-set \
  --cluster MyCluster \
  --service MyService \
  --primary-task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-svc/1234567890123456789

```

Ausgabe:

```

{
  "taskSet": {
    "id": "ecs-svc/1234567890123456789",
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-svc/1234567890123456789",
    "status": "PRIMARY",
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/sample-fargate:2",
    "computedDesiredCount": 1,
    "pendingCount": 0,
    "runningCount": 0,
    "createdAt": 1557128360.711,
    "updatedAt": 1557129412.653,
    "launchType": "EC2",
    "networkConfiguration": {

```

```
    "awsvpcConfiguration": {
      "subnets": [
        "subnet-12344321"
      ],
      "securityGroups": [
        "sg-12344312"
      ],
      "assignPublicIp": "DISABLED"
    }
  },
  "loadBalancers": [],
  "serviceRegistries": [],
  "scale": {
    "value": 50.0,
    "unit": "PERCENT"
  },
  "stabilityStatus": "STABILIZING",
  "stabilityStatusAt": 1557129279.914
}
}
```

- API-Einheiten finden Sie [UpdateServicePrimaryTaskSet](#) unter AWS CLI Befehlsreferenz.

update-service

Das folgende Codebeispiel zeigt die Verwendung `update-service`.

AWS CLI

Beispiel 1: Um die in einem Dienst verwendete Aufgabendefinition zu ändern

Im folgenden `update-service` Beispiel wird der `my-http-service` Dienst aktualisiert, sodass er die `amazon-ecs-sample` Aufgabendefinition verwendet.

```
aws ecs update-service --service my-http-service --task-definition amazon-ecs-sample
```

Beispiel 2: Um die Anzahl der Aufgaben in einem Service zu ändern

Im folgenden `update-service` Beispiel wird die gewünschte Anzahl an Aufgaben für den Dienst `my-http-service` auf 3 aktualisiert.

```
aws ecs update-service --service my-http-service --desired-count 3
```

Weitere Informationen finden Sie unter [Aktualisieren eines Service](#) im Amazon ECS Developer Guide.

- API-Einheiten finden Sie [UpdateService](#) in der AWS CLI Befehlsreferenz.

update-task-set

Das folgende Codebeispiel zeigt die Verwendung `update-task-set`.

AWS CLI

Um einen Tasksatz zu aktualisieren

Im folgenden `update-task-set` Beispiel wird ein Task-Set aktualisiert, um den Maßstab anzupassen.

```
aws ecs update-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-  
svc/1234567890123456789 \  
  --scale value=50,unit=PERCENT
```

Ausgabe:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "ACTIVE",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
    "computedDesiredCount": 0,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557128360.711,  
    "updatedAt": 1557129279.914,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12344321"  
        ]  
      }  
    }  
  }  
}
```

```
    ],
    "securityGroups": [
      "sg-12344321"
    ],
    "assignPublicIp": "DISABLED"
  }
},
"loadBalancers": [],
"serviceRegistries": [],
"scale": {
  "value": 50.0,
  "unit": "PERCENT"
},
"stabilityStatus": "STABILIZING",
"stabilityStatusAt": 1557129279.914
}
}
```

- API-Einheiten finden Sie [UpdateTaskSet](#) unter AWS CLI Befehlsreferenz.

EFS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden EFS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-file-system

Das folgende Codebeispiel zeigt die Verwendung `create-file-system`.

AWS CLI

Um ein verschlüsseltes Dateisystem zu erstellen

Im folgenden `create-file-system` Beispiel wird ein verschlüsseltes Dateisystem unter Verwendung der Standardeinstellung erstellt CMK. Außerdem wird das Tag hinzugefügt `Name=my-file-system`.

```
aws efs create-file-system \  
  --performance-mode generalPurpose \  
  --throughput-mode bursting \  
  --encrypted \  
  --tags Key=Name, Value=my-file-system
```

Ausgabe:

```
{  
  "OwnerId": "123456789012",  
  "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",  
  "FileSystemId": "fs-c7a0456e",  
  "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/  
fs-48499b4d",  
  "CreationTime": 1595286880.0,  
  "LifecycleState": "creating",  
  "Name": "my-file-system",  
  "NumberOfMountTargets": 0,  
  "SizeInBytes": {  
    "Value": 0,  
    "ValueInIA": 0,  
    "ValueInStandard": 0  
  },  
  "PerformanceMode": "generalPurpose",  
  "Encrypted": true,  
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-  
adcf-30d92example",  
  "ThroughputMode": "bursting",  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    }  
  ]  
}
```

```
}
```

Weitere Informationen finden Sie unter [Erstellen von EFS Amazon-Dateisystemen](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateFileSystem](#) unter AWS CLI Befehlsreferenz.

create-mount-target

Das folgende Codebeispiel zeigt die Verwendung `create-mount-target`.

AWS CLI

Um ein Mount-Ziel zu erstellen

Das folgende `create-mount-target` Beispiel erstellt ein Mount-Ziel für das angegebene Dateisystem.

```
aws efs create-mount-target \  
  --file-system-id fs-c7a0456e \  
  --subnet-id subnet-02bf4c428bexample \  
  --security-groups sg-068f739363example
```

Ausgabe:

```
{  
  "OwnerId": "123456789012",  
  "MountTargetId": "fsmt-f9a14450",  
  "FileSystemId": "fs-c7a0456e",  
  "SubnetId": "subnet-02bf4c428bexample",  
  "LifeCycleState": "creating",  
  "IpAddress": "10.0.1.24",  
  "NetworkInterfaceId": "eni-02d542216aexample",  
  "AvailabilityZoneId": "use2-az2",  
  "AvailabilityZoneName": "us-east-2b",  
  "VpcId": "vpc-0123456789abcdef0"  
}
```

Weitere Informationen finden Sie unter [Erstellen von Mount-Zielen](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateMountTarget](#) in der AWS CLI Befehlsreferenz.

delete-file-system

Das folgende Codebeispiel zeigt die Verwendung `delete-file-system`.

AWS CLI

Um ein Dateisystem zu löschen

Im folgenden `delete-file-system` Beispiel wird das angegebene Dateisystem gelöscht.

```
aws efs delete-file-system \  
  --file-system-id fs-c7a0456e
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines EFS Amazon-Dateisystems](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteFileSystem](#) unter AWS CLI Befehlsreferenz.

delete-mount-target

Das folgende Codebeispiel zeigt die Verwendung `delete-mount-target`.

AWS CLI

Um ein Mount-Ziel zu löschen

Das folgende `delete-mount-target` Beispiel löscht das angegebene Mount-Ziel.

```
aws efs delete-mount-target \  
  --mount-target-id fsmt-f9a14450
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen von Mount-Zielen](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteMountTarget](#) in der AWS CLI Befehlsreferenz.

describe-file-systems

Das folgende Codebeispiel zeigt die Verwendung `describe-file-systems`.

AWS CLI

Um ein Dateisystem zu beschreiben

Das folgende `describe-file-systems` Beispiel beschreibt das angegebene Dateisystem.

```
aws efs describe-file-systems \  
  --file-system-id fs-c7a0456e
```

Ausgabe:

```
{  
  "FileSystems": [  
    {  
      "OwnerId": "123456789012",  
      "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",  
      "FileSystemId": "fs-c7a0456e",  
      "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-  
system/fs-48499b4d",  
      "CreationTime": 1595286880.0,  
      "LifecycleState": "available",  
      "Name": "my-file-system",  
      "NumberOfMountTargets": 3,  
      "SizeInBytes": {  
        "Value": 6144,  
        "Timestamp": 1600991437.0,  
        "ValueInIA": 0,  
        "ValueInStandard": 6144  
      },  
      "PerformanceMode": "generalPurpose",  
      "Encrypted": true,  
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-  
adcf-30d92example",  
      "ThroughputMode": "bursting",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "my-file-system"  
        }  
      ]  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwaltung von EFS Amazon-Dateisystemen](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeFileSystems](#) in der AWS CLI Befehlsreferenz.

describe-mount-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-mount-targets`.

AWS CLI

Um ein Mount-Ziel zu beschreiben

Das folgende `describe-mount-targets` Beispiel beschreibt das angegebene Mount-Ziel.

```
aws efs describe-mount-targets \  
  --mount-target-id fsmt-f9a14450
```

Ausgabe:

```
{  
  "MountTargets": [  
    {  
      "OwnerId": "123456789012",  
      "MountTargetId": "fsmt-f9a14450",  
      "FileSystemId": "fs-c7a0456e",  
      "SubnetId": "subnet-02bf4c428bexample",  
      "LifeCycleState": "creating",  
      "IpAddress": "10.0.1.24",  
      "NetworkInterfaceId": "eni-02d542216aexample",  
      "AvailabilityZoneId": "use2-az2",  
      "AvailabilityZoneName": "us-east-2b",  
      "VpcId": "vpc-0123456789abcdef0"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Erstellen von Mount-Zielen](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMountTargets](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Um die Tags für ein Dateisystem zu beschreiben

Das folgende `describe-tags` Beispiel beschreibt die Tags für das angegebene Dateisystem.

```
aws efs describe-tags \  
  --file-system-id fs-c7a0456e
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    },  
    {  
      "Key": "Department",  
      "Value": "Business Intelligence"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Dateisystem-Tags](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTags](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Ressource abzurufen

Im folgenden `list-tags-for-resource` Beispiel werden die Tags abgerufen, die dem angegebenen Dateisystem zugeordnet sind.

```
aws efs list-tags-for-resource \  
  --resource-id fs-c7a0456e
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    },  
    {  
      "Key": "Department",  
      "Value": "Business Intelligence"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Dateisystem-Tags](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird das Tag `Department=Business Intelligence` dem angegebenen Dateisystem hinzugefügt.

```
aws efs tag-resource \  
  --resource-id fs-c7a0456e \  
  --tags Key=Department,Value="Business Intelligence"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Dateisystem-Tags](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem `Department` Tag-Schlüssel aus dem angegebenen Dateisystem entfernt.

```
aws efs untag-resource \  
  --resource-id fs-c7a0456e \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Dateisystem-Tags](#) im Amazon Elastic File System-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

EKS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden EKS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-encryption-config

Das folgende Codebeispiel zeigt die Verwendung `associate-encryption-config`.

AWS CLI

Um eine Verschlüsselungskonfiguration einem vorhandenen Cluster zuzuordnen

Das folgende `associate-encryption-config` Beispiel aktiviert die Verschlüsselung auf einem vorhandenen EKS Cluster, für den die Verschlüsselung noch nicht aktiviert ist.

```
aws eks associate-encryption-config \  
  --cluster-name my-eks-cluster \  
  --encryption-config '[{"resources":["secrets"],"provider":  
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

Ausgabe:

```
{  
  "update": {  
    "id": "3141b835-8103-423a-8e68-12c2521ffa4d",  
    "status": "InProgress",  
    "type": "AssociateEncryptionConfig",  
    "params": [  
      {  
        "type": "EncryptionConfig",  
        "value": "[{\\"resources\\":[\"secrets\\\"],\"provider\\\":{\\"keyArn\\\":  
\\\"arn:aws:kms:region-code:account:key/key\\\"}]}"  
      }  
    ],  
    "createdAt": "2024-03-14T11:01:26.297000-04:00",  
    "errors": []  
  }  
}
```

Weitere Informationen finden Sie unter [Enabling Secret Encryption on a existing cluster](#) im EKSAmerican-Benutzerhandbuch.

- API-Einheiten finden Sie [AssociateEncryptionConfig](#) unter AWS CLI Befehlsreferenz.

associate-identity-provider-config

Das folgende Codebeispiel zeigt die Verwendung `associate-identity-provider-config`.

AWS CLI

Ordnen Sie Ihrem EKS Amazon-Cluster einen Identitätsanbieter zu

Das folgende `associate-identity-provider-config` Beispiel ordnet Ihrem EKS Amazon-Cluster einen Identitätsanbieter zu.

```
aws eks associate-identity-provider-config \
  --cluster-name my-eks-cluster \
  --oidc 'identityProviderConfigName=my-identity-provider,issuerUrl=https://oidc.eks.us-east-2.amazonaws.com/id/38D6A4619A0A69E342B113ED7F1A7652,clientId=kubernetes,usernameClaim=email,usernamePrefix=my-username-prefix,groupsClaim=my-claim,groupsPrefix=my-groups-prefix,requiredClaims={Claim1=value1,Claim2=value2}' \
  --tags env=dev
```

Ausgabe:

```
{
  "update": {
    "id": "8c6c1bef-61fe-42ac-a242-89412387b8e7",
    "status": "InProgress",
    "type": "AssociateIdentityProviderConfig",
    "params": [
      {
        "type": "IdentityProviderConfig",
        "value": "[{"type": "oidc", "name": "my-identity-provider"}]"
      }
    ],
    "createdAt": "2024-04-11T13:46:49.648000-04:00",
    "errors": []
  },
  "tags": {
    "env": "dev"
  }
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern für Ihren Cluster von einem OpenID Connect-Identitätsanbieter — Zuordnen eines OIDC Identitätsanbieters](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AssociateIdentityProviderConfig](#) in der AWS CLI Befehlsreferenz.

create-addon

Das folgende Codebeispiel zeigt die Verwendung `create-addon`.

AWS CLI

Beispiel 1: Um ein EKS Amazon-Add-on mit einer kompatiblen Standardversion für die jeweilige EKS Cluster-Version zu erstellen

Der folgende `create-addon` Beispielfehler erstellt ein EKS Amazon-Add-on mit einer kompatiblen Standardversion für die jeweilige EKS Cluster-Version.

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.15.1-eksbuild.1",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/1ec71ee1-b9c2-8915-4e17-e8be0a55a149",  
    "createdAt": "2024-03-14T12:20:03.264000-04:00",  
    "modifiedAt": "2024-03-14T12:20:03.283000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 2: Um ein EKS Amazon-Add-on mit einer bestimmten Add-On-Version zu erstellen

Der folgende `create-addon` Beispielfehl erstellt ein EKS Amazon-Add-on mit einer bestimmten Add-On-Version.

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/34c71ee6-7738-6c8b-c6bd-3921a176b5ff",  
    "createdAt": "2024-03-14T12:30:24.507000-04:00",  
    "modifiedAt": "2024-03-14T12:30:24.521000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 3: Um ein EKS Amazon-Add-on mit benutzerdefinierten Konfigurationswerten zu erstellen und Konfliktdetails zu lösen

Der folgende `create-addon` Beispielfehl erstellt ein EKS Amazon-Add-on mit benutzerdefinierten Konfigurationswerten und behebt Konfliktdetails.

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2 \  
  --configuration-values '{"resources":{"limits":{"cpu":"100m"}}}' \  
  --resolve-conflicts OVERWRITE
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-  
addon/a6c71ee9-0304-9237-1be8-25af1b0f1ffb",  
    "createdAt": "2024-03-14T12:35:58.313000-04:00",  
    "modifiedAt": "2024-03-14T12:35:58.327000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {},  
    "configurationValues": "{\"resources\":{\"limits\":{\"cpu\":\"100m\"}}}"  
  }  
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 4: So erstellen Sie ein EKS Amazon-Add-on mit einer Datei mit benutzerdefinierten JSON Konfigurationswerten

Der folgende `create-addon` Beispielbefehl erstellt ein EKS Amazon-Add-on mit benutzerdefinierten Konfigurationswerten und Details zur Konfliktlösung.

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2 \  
  --configuration-values '{"resources":{"limits":{"cpu":"100m"}}}' \  
  --resolve-conflicts OVERWRITE
```

```
--service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
--addon-version v1.16.4-eksbuild.2 \  
--configuration-values 'file://configuration-values.json' \  
--resolve-conflicts OVERWRITE \  
--tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

Inhalt von `configuration-values.json`:

```
{  
  "resources": {  
    "limits": {  
      "cpu": "150m"  
    }  
  },  
  "env": {  
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"  
  }  
}
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/d8c71ef8-fbd8-07d0-fb32-6a7be19eeced",  
    "createdAt": "2024-03-14T13:10:51.763000-04:00",  
    "modifiedAt": "2024-03-14T13:10:51.777000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {  
      "eks-addon-key-1": "value-1",  
      "eks-addon-key-2": "value-2"  
    },  
    "configurationValues": "{\n  \"resources\": {\n    \"limits\":  
\n      \"cpu\": \"150m\"\n    }\n  },\n  \"env\": {\n    \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n  }\n}"  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [EKSAmazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSAmazon-Benutzerhandbuch.

Beispiel 5: So erstellen Sie ein EKS Amazon-Add-on mit einer Datei mit benutzerdefinierten YAML Konfigurationswerten

Der folgende `create-addon` Beispielbefehl erstellt ein EKS Amazon-Add-on mit benutzerdefinierten Konfigurationswerten und Details zur Konfliktlösung.

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2 \  
  --configuration-values 'file://configuration-values.yaml' \  
  --resolve-conflicts OVERWRITE \  
  --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

Inhalt von `configuration-values.yaml`:

```
resources:  
  limits:  
    cpu: '100m'  
env:  
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/d4c71efb-3909-6f36-a548-402cd4b5d59e",  
    "createdAt": "2024-03-14T13:15:45.220000-04:00",  
  }  
}
```

```

    "modifiedAt": "2024-03-14T13:15:45.237000-04:00",
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {
      "eks-addon-key-3": "value-3",
      "eks-addon-key-4": "value-4"
    },
    "configurationValues": "resources:\n    limits:\n        cpu: '100m'\nenv:\nAWS_VPC_K8S_CNI_LOGLEVEL: 'INFO'"
  }
}

```

Weitere Informationen finden Sie unter [EKSAmerican-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSAmerican-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateAddon](#) in der AWS CLI Befehlsreferenz.

create-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-cluster`.

AWS CLI

Um einen neuen Cluster zu erstellen

Dieser Beispielbefehl erstellt einen Cluster mit `prod` dem Namen Ihrer Standardregion.

Befehl:

```

aws eks create-cluster --name prod \
--role-arn arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J70NKE3BQ4PI \
--resources-vpc-config subnetIds=subnet-6782e71e,subnet-
e7e761ac,securityGroupIds=sg-6979fe18

```

Ausgabe:

```

{
  "cluster": {
    "name": "prod",
    "arn": "arn:aws:eks:us-west-2:012345678910:cluster/prod",
    "createdAt": 1527808069.147,
    "version": "1.10",
  }
}

```



```

    "roleArn": "arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J70NKE3BQ4PI",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-6782e71e",
        "subnet-e7e761ac"
      ],
      "securityGroupIds": [
        "sg-6979fe18"
      ],
      "vpcId": "vpc-950809ec"
    },
    "status": "CREATING",
    "certificateAuthority": {}
  }
}

```

Um einen neuen Cluster mit aktiviertem privaten Endpunktzugriff und aktivierter Protokollierung zu erstellen

Dieser Beispielbefehl erstellt einen Cluster mit dem Namen Ihrer Standardregion, `example` in dem der öffentliche Endpunktzugriff deaktiviert, der private Endpunktzugriff aktiviert und alle Protokollierungstypen aktiviert sind.

Befehl:

```

aws eks create-cluster --name example --kubernetes-version 1.12 \
--role-arn arn:aws:iam::012345678910:role/example-cluster-ServiceRole-1XWBQWYSFRE2Q \
--resources-vpc-
config subnetIds=subnet-0a188dcd2f9a632f,subnet-09290d93da4278664,subnet-0f21dd86e0e91134a, \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'

```

Ausgabe:

```

{
  "cluster": {
    "name": "example",
    "arn": "arn:aws:eks:us-west-2:012345678910:cluster/example",
    "createdAt": 1565804921.901,
    "version": "1.12",

```

```
    "roleArn": "arn:aws:iam::012345678910:role/example-cluster-ServiceRole-1XWBQWYSFRE2Q",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0a188dccd2f9a632f",
        "subnet-09290d93da4278664",
        "subnet-0f21dd86e0e91134a",
        "subnet-0173dead68481a583",
        "subnet-051f70a57ed6fcab6",
        "subnet-01322339c5c7de9b4"
      ],
      "securityGroupIds": [
        "sg-0c5b580845a031c10"
      ],
      "vpcId": "vpc-0f622c01f68d4afec",
      "endpointPublicAccess": false,
      "endpointPrivateAccess": true
    },
    "logging": {
      "clusterLogging": [
        {
          "types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ],
          "enabled": true
        }
      ]
    },
    "status": "CREATING",
    "certificateAuthority": {},
    "platformVersion": "eks.3"
  }
}
```

- API-Einheiten finden Sie [CreateCluster](#) unter AWS CLI Befehlsreferenz.

create-fargate-profile

Das folgende Codebeispiel zeigt die Verwendung `create-fargate-profile`.

AWS CLI

Beispiel 1: Erstellen Sie ein EKS Fargate-Profil für einen Selektor mit einem Namespace

Im folgenden `create-fargate-profile` Beispiel wird ein EKS Fargate-Profil für einen Selektor mit einem Namespace erstellt.

```
aws eks create-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \  
  --fargate-profile-name my-fargate-profile \  
  --selectors '[{"namespace": "default"}]'
```

Ausgabe:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/a2c72bca-318e-abe8-8ed1-27c6d4892e9e",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T12:38:47.368000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default"  
      }  
    ],  
    "status": "CREATING",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [AWS Fargate-Profil — Erstellen eines Fargate-Profiles](#) im EKSAmerican-Benutzerhandbuch.

Beispiel 2: Erstellen Sie ein EKS Fargate-Profil für einen Selektor mit einem Namespace und Labels

Im folgenden `create-fargate-profile` Beispiel wird ein EKS Fargate-Profil für einen Selektor mit einem Namespace und Labels erstellt.

```
aws eks create-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \  
  --fargate-profile-name my-fargate-profile \  
  --selectors '[{"namespace": "default", "labels": {"labelname1":  
  "labelvalue1"}}]'
```

Ausgabe:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/88c72bc7-e8a4-fa34-44e4-2f1397224bb3",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T12:33:48.125000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default",  
        "labels": {  
          "labelname1": "labelvalue1"  
        }  
      }  
    ],  
    "status": "CREATING",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [AWS Fargate-Profil — Erstellen eines Fargate-Profiles](#) im EKSAmerican-Benutzerhandbuch.

Beispiel 3: Erstellen Sie ein EKS Fargate-Profil für einen Selektor mit einem Namespace und Labels sowie Subnetzen, in denen ein IDs Pod gestartet werden soll.

Im folgenden `create-fargate-profile` Beispiel wird ein EKS Fargate-Profil für einen Selektor mit einem Namespace und Labels sowie IDs von Subnetzen erstellt, in denen ein Pod gestartet werden soll.

```
aws eks create-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \  
  --fargate-profile-name my-fargate-profile \  
  --selectors '[{"namespace": "default", "labels": {"labelname1":  
  "labelvalue1"}}]' \  
  --subnets ["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",  
  "subnet-0e2907431c9988b72"]
```

Ausgabe:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/e8c72bc8-e87b-5eb6-57cb-ed4fe57577e3",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T12:35:58.640000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default",  
        "labels": {  
          "labelname1": "labelvalue1"  
        }  
      }  
    ],  
    "status": "CREATING",  
    "tags": {}  
  }  
}
```

}

Weitere Informationen finden Sie unter [AWS Fargate-Profil — Erstellen eines Fargate-Profils](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 4: Erstellen Sie ein EKS Fargate-Profil für einen Selektor mit mehreren Namespaces und Labels sowie Subnetzen, in denen ein IDs Pod gestartet werden soll

Das folgende `create-fargate-profile` Beispiel erstellt ein EKS Fargate-Profil für einen Selektor mit mehreren Namespaces und Labels sowie Subnetzen, in denen ein IDs Pod gestartet werden soll.

```
aws eks create-fargate-profile \
  --cluster-name my-eks-cluster \
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
  --fargate-profile-name my-fargate-profile \
  --selectors '[{"namespace": "default1", "labels": {"labelname1": "labelvalue1",
"labelname2": "labelvalue2"}}, {"namespace": "default2", "labels": {"labelname1":
"labelvalue1", "labelname2": "labelvalue2"}}]' \
  --subnets ["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",
"subnet-0e2907431c9988b72"] \
  --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2":
"value-2"}'
```

Ausgabe:

```
{
  "fargateProfile": {
    "fargateProfileName": "my-fargate-profile",
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/4cc72bbf-b766-8ee6-8d29-e62748feb3cd",
    "clusterName": "my-eks-cluster",
    "createdAt": "2024-03-19T12:15:55.271000-04:00",
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "subnets": [
      "subnet-09d912bb63ef21b9a",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-0e2907431c9988b72"
    ],
    "selectors": [
      {
        "namespace": "default1",
```

```

        "labels": {
            "labelname2": "labelvalue2",
            "labelname1": "labelvalue1"
        }
    },
    {
        "namespace": "default2",
        "labels": {
            "labelname2": "labelvalue2",
            "labelname1": "labelvalue1"
        }
    }
],
"status": "CREATING",
"tags": {
    "eks-fargate-profile-key-2": "value-2",
    "eks-fargate-profile-key-1": "value-1"
}
}
}

```

Weitere Informationen finden Sie unter [AWS Fargate-Profil — Erstellen eines Fargate-Profiles](#) im EKSAmerican-Benutzerhandbuch.

Beispiel 5: Erstellen Sie ein EKS Fargate-Profil mit einem Platzhalter-Selektor für Namespaces und Labels sowie mit Subnetzen, in denen ein Pod gestartet werden soll

Das folgende `create-fargate-profile` Beispiel erstellt ein EKS Fargate-Profil für einen Selektor mit mehreren Namespaces und Labels sowie Subnetzen, in denen ein IDs Pod gestartet werden soll.

```

aws eks create-fargate-profile \
  --cluster-name my-eks-cluster \
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
  --fargate-profile-name my-fargate-profile \
  --selectors '[{"namespace": "prod*", "labels": {"labelname*?": "*value1"}}, {"namespace": "*dev*", "labels": {"labelname*?": "*value*"}}]' \
  --subnets ["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d", "subnet-0e2907431c9988b72"] \
  --tags [{"eks-fargate-profile-key-1": "value-1" , eks-fargate-profile-key-2: "value-2"}]

```

Ausgabe:

```

{
  "fargateProfile": {
    "fargateProfileName": "my-fargate-profile",
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/my-fargate-profile/e8c72bd6-5966-0bfe-b77b-1802893e5a6f",
    "clusterName": "my-eks-cluster",
    "createdAt": "2024-03-19T13:05:20.550000-04:00",
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "subnets": [
      "subnet-09d912bb63ef21b9a",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-0e2907431c9988b72"
    ],
    "selectors": [
      {
        "namespace": "prod*",
        "labels": {
          "labelname*?": "*value1"
        }
      },
      {
        "namespace": "*dev*",
        "labels": {
          "labelname*?": "*value*"
        }
      }
    ],
    "status": "CREATING",
    "tags": {
      "eks-fargate-profile-key-2": "value-2",
      "eks-fargate-profile-key-1": "value-1"
    }
  }
}

```

Weitere Informationen finden Sie unter [AWS Fargate-Profil — Erstellen eines Fargate-Profiles](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateFargateProfile AWS CLI Befehlsreferenz](#).

create-nodegroup

Das folgende Codebeispiel zeigt die Verwendung `create-nodegroup`.

AWS CLI

Beispiel 1: Erstellt eine verwaltete Knotengruppe für einen EKS Amazon-Cluster

Im folgenden `create-nodegroup` Beispiel wird eine verwaltete Knotengruppe für einen EKS Amazon-Cluster erstellt.

```
aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --node-role arn:aws:iam::111122223333:role/role-name \
  --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a" \
  --scaling-config minSize=1,maxSize=3,desiredSize=1 \
  --region us-east-2
```

Ausgabe:

```
{
  "nodegroup": {
    "nodegroupName": "my-eks-nodegroup",
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-nodegroup/bac7550f-b8b8-5fbb-4f3e-7502a931119e",
    "clusterName": "my-eks-cluster",
    "version": "1.26",
    "releaseVersion": "1.26.12-20240329",
    "createdAt": "2024-04-04T13:19:32.260000-04:00",
    "modifiedAt": "2024-04-04T13:19:32.260000-04:00",
    "status": "CREATING",
    "capacityType": "ON_DEMAND",
    "scalingConfig": {
      "minSize": 1,
      "maxSize": 3,
      "desiredSize": 1
    },
    "instanceTypes": [
      "t3.medium"
    ],
    "subnets": [
      "subnet-0e2907431c9988b72", "subnet-04ad87f71c6e5ab4d",
      "subnet-09d912bb63ef21b9a"
    ],
  }
}
```

```

    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",
    "diskSize": 20,
    "health": {
      "issues": []
    },
    "updateConfig": {
      "maxUnavailable": 1
    },
    "tags": {}
  }
}

```

Weitere Informationen finden Sie unter [Erstellen einer verwalteten Knotengruppe](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 2: Erstellt eine verwaltete Knotengruppe für einen EKS Amazon-Cluster mit benutzerdefinierten Instanztypen und Festplattengröße

Das folgende `create-nodegroup` Beispiel erstellt eine verwaltete Knotengruppe für einen EKS Amazon-Cluster mit benutzerdefinierten Instanztypen und Festplattengröße.

```

aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --node-role arn:aws:iam::111122223333:role/role-name \
  --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a" \
  --scaling-config minSize=1,maxSize=3,desiredSize=1 \
  --capacity-type ON_DEMAND \
  --instance-types 'm5.large' \
  --disk-size 50 \
  --region us-east-2

```

Ausgabe:

```

{
  "nodegroup": {
    "nodegroupName": "my-eks-nodegroup",
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-nodegroup/c0c7551b-e4f9-73d9-992c-a450fdb82322",

```

```
"clusterName": "my-eks-cluster",
"version": "1.26",
"releaseVersion": "1.26.12-20240329",
"createdAt": "2024-04-04T13:46:07.595000-04:00",
"modifiedAt": "2024-04-04T13:46:07.595000-04:00",
"status": "CREATING",
"capacityType": "ON_DEMAND",
"scalingConfig": {
  "minSize": 1,
  "maxSize": 3,
  "desiredSize": 1
},
"instanceTypes": [
  "m5.large"
],
"subnets": [
  "subnet-0e2907431c9988b72",
  "subnet-04ad87f71c6e5ab4d",
  "subnet-09d912bb63ef21b9a"
],
"amiType": "AL2_x86_64",
"nodeRole": "arn:aws:iam::111122223333:role/role-name",
"diskSize": 50,
"health": {
  "issues": []
},
"updateConfig": {
  "maxUnavailable": 1
},
"tags": {}
}
```

Weitere Informationen finden Sie unter [Erstellen einer verwalteten Knotengruppe](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 3: Erstellt eine verwaltete Knotengruppe für einen EKS Amazon-Cluster mit benutzerdefinierten Instanztypen, Festplattengröße, AMI-Typ, Kapazitätstyp, Update-Konfiguration, Labels, Taints und Tags.

Das folgende `create-nodegroup` Beispiel erstellt eine verwaltete Knotengruppe für einen EKS Amazon-Cluster mit benutzerdefinierten Instance-Typen, Festplattengröße, AMI-Type, Capacity-Type, Update-Config, Labels, Taints und Tags.

```
aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --node-role arn:aws:iam::111122223333:role/role-name \
  --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"
\
  --scaling-config minSize=1,maxSize=5,desiredSize=4 \
  --instance-types 't3.large' \
  --disk-size 50 \
  --ami-type AL2_x86_64 \
  --capacity-type SPOT \
  --update-config maxUnavailable=2 \
  --labels '{"my-eks-nodegroup-label-1": "value-1" , "my-eks-nodegroup-label-2":
"value-2"}' \
  --taints '{"key": "taint-key-1" , "value": "taint-value-1", "effect":
"NO_EXECUTE"}' \
  --tags '{"my-eks-nodegroup-key-1": "value-1" , "my-eks-nodegroup-key-2":
"value-2"}'
```

Ausgabe:

```
{
  "nodegroup": {
    "nodegroupName": "my-eks-nodegroup",
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/88c75524-97af-0cb9-a9c5-7c0423ab5314",
    "clusterName": "my-eks-cluster",
    "version": "1.26",
    "releaseVersion": "1.26.12-20240329",
    "createdAt": "2024-04-04T14:05:07.940000-04:00",
    "modifiedAt": "2024-04-04T14:05:07.940000-04:00",
    "status": "CREATING",
    "capacityType": "SPOT",
    "scalingConfig": {
      "minSize": 1,
      "maxSize": 5,
      "desiredSize": 4
    },
    "instanceTypes": [
      "t3.large"
    ],
    "subnets": [
```

```
        "subnet-0e2907431c9988b72",
        "subnet-04ad87f71c6e5ab4d",
        "subnet-09d912bb63ef21b9a"
    ],
    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",
    "labels": {
        "my-eks-nodegroup-label-2": "value-2",
        "my-eks-nodegroup-label-1": "value-1"
    },
    "taints": [
        {
            "key": "taint-key-1",
            "value": "taint-value-1",
            "effect": "NO_EXECUTE"
        }
    ],
    "diskSize": 50,
    "health": {
        "issues": []
    },
    "updateConfig": {
        "maxUnavailable": 2
    },
    "tags": {
        "my-eks-nodegroup-key-1": "value-1",
        "my-eks-nodegroup-key-2": "value-2"
    }
}
}
```

Weitere Informationen finden Sie unter [Erstellen einer verwalteten Knotengruppe](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateNodegroup](#) in der AWS CLI Befehlsreferenz.

delete-addon

Das folgende Codebeispiel zeigt die Verwendung `delete-addon`.

AWS CLI

Beispiel 1. Um ein EKS Amazon-Add-on zu löschen, aber die Zusatzsoftware im EKS Cluster beizubehalten

Der folgende `delete-addon` Beispielbefehl löscht ein EKS Amazon-Add-on, behält aber die Zusatzsoftware auf dem EKS Cluster bei.

```
aws eks delete-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --preserve
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "DELETING",  
    "addonVersion": "v1.9.3-eksbuild.7",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/a8c71ed3-944e-898b-9167-c763856af4b8",  
    "createdAt": "2024-03-14T11:49:09.009000-04:00",  
    "modifiedAt": "2024-03-14T12:03:49.776000-04:00",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von EKS Amazon-Add-Ons — Löschen eines Add-ons](#) im Amazon EKS.

Beispiel 2. Um ein EKS Amazon-Add-on zu löschen und auch die Zusatzsoftware aus dem EKS Cluster zu löschen

Der folgende `delete-addon` Beispielbefehl löscht ein EKS Amazon-Add-on und löscht auch die Zusatzsoftware aus dem EKS Cluster.

```
aws eks delete-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --preserve
```

```
--cluster-name my-eks-cluster \  
--addon-name my-eks-addon
```

Ausgabe:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "DELETING",  
    "addonVersion": "v1.15.1-eksbuild.1",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-  
addon/bac71ed1-ec43-3bb6-88ea-f243cdb58954",  
    "createdAt": "2024-03-14T11:45:31.983000-04:00",  
    "modifiedAt": "2024-03-14T11:58:40.136000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von EKS Amazon-Add-Ons — Löschen eines Add-ons](#) im Amazon EKS.

- API-Einheiten finden Sie [DeleteAddon](#) in der AWS CLI Befehlsreferenz.

delete-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster`.

AWS CLI

Löschen Sie eine EKS Amazon-Cluster-Steuerebene

Das folgende `delete-cluster` Beispiel löscht eine EKS Amazon-Cluster-Steuerebene.

```
aws eks delete-cluster \  
--name my-eks-cluster
```

Ausgabe:

```
{
  "cluster": {
    "name": "my-eks-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
    "createdAt": "2024-03-14T11:31:44.348000-04:00",
    "version": "1.27",
    "endpoint": "https://DALSJ343KE23J3RN45653DSKJTT647TYD.yl4.us-
east-2.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0fb75d2d8401716e7",
        "subnet-02184492f67a3d0f9",
        "subnet-04098063527aab776",
        "subnet-0e2907431c9988b72",
        "subnet-04ad87f71c6e5ab4d",
        "subnet-09d912bb63ef21b9a"
      ],
      "securityGroupIds": [
        "sg-0c1327f6270afbb36"
      ],
      "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
      "vpcId": "vpc-0012b8e1cc0abb17d",
      "endpointPublicAccess": true,
      "endpointPrivateAccess": true,
      "publicAccessCidrs": [
        "0.0.0.0/0"
      ]
    },
    "kubernetesNetworkConfig": {
      "serviceIpv4Cidr": "10.100.0.0/16",
      "ipFamily": "ipv4"
    },
    "logging": {
      "clusterLogging": [
        {
          "types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ]
        }
      ]
    }
  }
}
```



```

        ],
        "enabled": true
    }
]
},
"identity": {
    "oidc": {
        "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
DAL5J343KE23J3RN45653DSKJTT647TYD"
    }
},
"status": "DELETING",
"certificateAuthority": {
    "data": "XXX_CA_DATA_XXX"
},
"platformVersion": "eks.16",
"tags": {
    "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
    "alpha.eksctl.io/cluster-name": "my-eks-cluster",
    "karpenter.sh/discovery": "my-eks-cluster",
    "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
    "auto-delete": "no",
    "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
    "EKS-Cluster-Name": "my-eks-cluster",
    "alpha.eksctl.io/cluster-oidc-enabled": "true",
    "aws:cloudformation:logical-id": "ControlPlane",
    "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
    "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
},
"accessConfig": {
    "authenticationMode": "API_AND_CONFIG_MAP"
}
}
}

```

Weitere Informationen finden Sie unter [Löschen eines EKS Amazon-Clusters](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteCluster](#) in der AWS CLI Befehlsreferenz.

delete-fargate-profile

Das folgende Codebeispiel zeigt die Verwendung `delete-fargate-profile`.

AWS CLI

Beispiel 1: Erstellen Sie ein EKS Fargate-Profil für einen Selektor mit einem Namespace

Im folgenden `delete-fargate-profile` Beispiel wird ein EKS Fargate-Profil für einen Selektor mit einem Namespace erstellt.

```
aws eks delete-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --fargate-profile-name my-fargate-profile
```

Ausgabe:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/1ac72bb3-3fc6-2631-f1e1-98bff53bed62",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T11:48:39.975000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default",  
        "labels": {  
          "foo": "bar"  
        }  
      }  
    ],  
    "status": "DELETING",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [AWS Fargate-Profil — Löschen eines Fargate](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteFargateProfile AWS CLI Befehlsreferenz](#).

delete-nodegroup

Das folgende Codebeispiel zeigt die Verwendung `delete-nodegroup`.

AWS CLI

Beispiel 1: Löschen einer verwalteten Knotengruppe für einen EKS Amazon-Cluster

Das folgende `delete-nodegroup` Beispiel löscht eine verwaltete Knotengruppe für einen EKS Amazon-Cluster.

```
aws eks delete-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup
```

Ausgabe:

```
{
  "nodegroup": {
    "nodegroupName": "my-eks-nodegroup",
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-nodegroup/1ec75f5f-0e21-dcc0-b46e-f9c442685cd8",
    "clusterName": "my-eks-cluster",
    "version": "1.26",
    "releaseVersion": "1.26.12-20240329",
    "createdAt": "2024-04-08T13:25:15.033000-04:00",
    "modifiedAt": "2024-04-08T13:25:31.252000-04:00",
    "status": "DELETING",
    "capacityType": "SPOT",
    "scalingConfig": {
      "minSize": 1,
      "maxSize": 5,
      "desiredSize": 4
    },
    "instanceTypes": [
      "t3.large"
    ],
  },
}
```

```
    "subnets": [
      "subnet-0e2907431c9988b72",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-09d912bb63ef21b9a"
    ],
    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",
    "labels": {
      "my-eks-nodegroup-label-2": "value-2",
      "my-eks-nodegroup-label-1": "value-1"
    },
    "taints": [
      {
        "key": "taint-key-1",
        "value": "taint-value-1",
        "effect": "NO_EXECUTE"
      }
    ],
    "diskSize": 50,
    "health": {
      "issues": []
    },
    "updateConfig": {
      "maxUnavailable": 2
    },
    "tags": {
      "my-eks-nodegroup-key-1": "value-1",
      "my-eks-nodegroup-key-2": "value-2"
    }
  }
}
```

- API-Einheiten finden Sie [DeleteNodegroup](#) in der AWS CLI Befehlsreferenz.

deregister-cluster

Das folgende Codebeispiel zeigt die Verwendung `deregister-cluster`.

AWS CLI

Um einen verbundenen Cluster abzumelden, um ihn aus der EKS Amazon-Steuerebene zu entfernen

Im folgenden `deregister-cluster` Beispiel wird die Registrierung eines verbundenen Clusters aufgehoben, um ihn aus der EKS Amazon-Steuerebene zu entfernen.

```
aws eks deregister-cluster \  
  --name my-eks-anywhere-cluster
```

Ausgabe:

```
{  
  "cluster": {  
    "name": "my-eks-anywhere-cluster",  
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",  
    "createdAt": "2024-04-12T12:38:37.561000-04:00",  
    "status": "DELETING",  
    "tags": {},  
    "connectorConfig": {  
      "activationId": "dfb5ad28-13c3-4e26-8a19-5b2457638c74",  
      "activationExpiry": "2024-04-15T12:38:37.082000-04:00",  
      "provider": "EKS_ANYWHERE",  
      "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Deregistrierung eines Clusters](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterCluster](#) in der AWS CLI Befehlsreferenz.

describe-addon-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-addon-configuration`.

AWS CLI

Beispiel 1: Verfügbare Konfigurationsoptionen bei der Erstellung oder Aktualisierung von Amazon vpc-cni AddOns

Das folgende `describe-addon-configuration` Beispiel gibt das gesamte verfügbare Konfigurationsschema zurück, das Sie verwenden, wenn ein Add-on für das vpc-cni-Add-on mit der entsprechenden Version erstellt oder aktualisiert wird.

```
aws eks describe-addon-configuration \
  --addon-name vpc-cni \
  --addon-version v1.15.1-eksbuild.1
```

Ausgabe:

```
{
  "addonName": "vpc-cni",
  "addonVersion": "v1.15.1-eksbuild.1",
  "configurationSchema": "{\ref\": \"#/definitions/VpcCni\", \"schema\": \"http://
json-schema.org/draft-06/schema#\", \"definitions\": {\Affinity\": {\type\":
[\"object\", \"null\"]}, \"EniConfig\": {\additionalProperties\": false, \"properties
\": {\create\": {\type\": \"boolean\"}, \"region\": {\type\": \"string\"}, \"subnets\":
{\additionalProperties\": {\additionalProperties\": false, \"properties\": {\id\":
{\type\": \"string\"}, \"securityGroups\": {\items\": {\type\": \"string\"}, \"type\":
\"array\"}}, \"required\": [\"id\"], \"type\": \"object\"}, \"minProperties\": 1, \"type
\": \"object\"}}, \"required\": [\"create\", \"region\", \"subnets\"], \"type\": \"object
\", \"Env\": {\additionalProperties\": false, \"properties\": {\ADDITIONAL_ENI_TAGS
\": {\type\": \"string\"}, \"ANNOTATE_POD_IP\": {\format\": \"boolean\", \"type\":
\"string\"}, \"AWS_EC2_ENDPOINT\": {\type\": \"string\"}, \"AWS_EXTERNAL_SERVICE_CIDRS
\": {\type\": \"string\"}, \"AWS_MANAGE_ENIS_NON_SCHEDULABLE\": {\format\": \"boolean
\", \"type\": \"string\"}, \"AWS_VPC_CNI_NODE_PORT_SUPPORT\": {\format\": \"boolean
\", \"type\": \"string\"}, \"AWS_VPC_ENI_MTU\": {\format\": \"integer\", \"type\":
\"string\"}, \"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\": {\format\": \"boolean\", \"type
\": \"string\"}, \"AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS\": {\type\": \"string\"},
\"AWS_VPC_K8S_CNI_EXTERNALSNAT\": {\format\": \"boolean\", \"type\": \"string\"},
\"AWS_VPC_K8S_CNI_LOGLEVEL\": {\type\": \"string\"}, \"AWS_VPC_K8S_CNI_LOG_FILE\":
{\type\": \"string\"}, \"AWS_VPC_K8S_CNI_RANDOMIZESNAT\": {\type\": \"string\"},
\"AWS_VPC_K8S_CNI_VETHPREFIX\": {\type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_FILE
\": {\type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\": {\type\": \"string
\"}, \"CLUSTER_ENDPOINT\": {\type\": \"string\"}, \"DISABLE_INTROSPECTION\":
{\format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_LEAKED_ENI_CLEANUP\":
{\format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_METRICS\": {\format
\": \"boolean\", \"type\": \"string\"}, \"DISABLE_NETWORK_RESOURCE_PROVISIONING
\": {\format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_POD_V6\": {\format
\": \"boolean\", \"type\": \"string\"}, \"ENABLE_BANDWIDTH_PLUGIN\": {\format\":
\"boolean\", \"type\": \"string\"}, \"ENABLE_POD_ENI\": {\format\": \"boolean\",
\"type\": \"string\"}, \"ENABLE_PREFIX_DELEGATION\": {\format\": \"boolean\",
\"type\": \"string\"}, \"ENABLE_V4_EGRESS\": {\format\": \"boolean\", \"type\":
\"string\"}, \"ENABLE_V6_EGRESS\": {\format\": \"boolean\", \"type\": \"string\"},
\"ENI_CONFIG_ANNOTATION_DEF\": {\type\": \"string\"}, \"ENI_CONFIG_LABEL_DEF\":
{\type\": \"string\"}, \"INTROSPECTION_BIND_ADDRESS\": {\type\": \"string\"},
\"IP_COOLDOWN_PERIOD\": {\format\": \"integer\", \"type\": \"string\"}, \"MAX_ENI
```

```

\":"format\":"integer\","type\":"string\"},\\"MINIMUM_IP_TARGET\":"format
\":"integer\","type\":"string\"},\\"POD_SECURITY_GROUP_ENFORCING_MODE\":"
{\\"type\":"string\"},\\"WARM_ENI_TARGET\":"format\":"integer\","type\":"
\":"string\"},\\"WARM_IP_TARGET\":"format\":"integer\","type\":"string\"},
\\"WARM_PREFIX_TARGET\":"format\":"integer\","type\":"string\"}},\\"title
\":"Env\","type\":"object\"},\\"Init\":"additionalProperties\":"false,
\\"properties\":"env\":"$ref\":"#/definitions/InitEnv\"}},\\"title\":"Init
\","type\":"object\"},\\"InitEnv\":"additionalProperties\":"false,\\"properties
\":"DISABLE_TCP_EARLY_DEMUX\":"format\":"boolean\","type\":"string\"},
\\"ENABLE_V6_EGRESS\":"format\":"boolean\","type\":"string\"}},\\"title\":"
\\"InitEnv\","type\":"object\"},\\"Limits\":"additionalProperties\":"false,
\\"properties\":"cpu\":"type\":"string\"},\\"memory\":"type\":"string\"}},
\\"title\":"Limits\","type\":"object\"},\\"NodeAgent\":"additionalProperties
\":"false,\\"properties\":"enableCloudWatchLogs\":"format\":"boolean\","
\":"type\":"string\"},\\"enablePolicyEventLogs\":"format\":"boolean\","type\":"
\":"string\"},\\"healthProbeBindAddr\":"format\":"integer\","type\":"string
\"},\\"metricsBindAddr\":"format\":"integer\","type\":"string\"}},\\"title\":"
\\"NodeAgent\","type\":"object\"},\\"Resources\":"additionalProperties\":"false,
\\"properties\":"limits\":"$ref\":"#/definitions/Limits\"},\\"requests\":"
{\\"$ref\":"#/definitions/Limits\"}},\\"title\":"Resources\","type\":"object
\"},\\"Tolerations\":"additionalProperties\":"false,\\"items\":"type\":"object
\"},\\"type\":"array\"},\\"VpcCni\":"additionalProperties\":"false,\\"properties
\":"affinity\":"$ref\":"#/definitions/Affinity\"},\\"enableNetworkPolicy\":"
{\\"format\":"boolean\","type\":"string\"},\\"enableWindowsIpam\":"format\":"
\":"boolean\","type\":"string\"},\\"eniConfig\":"$ref\":"#/definitions/EniConfig
\"},\\"env\":"$ref\":"#/definitions/Env\"},\\"init\":"$ref\":"#/definitions/Init
\"},\\"livenessProbeTimeoutSeconds\":"type\":"integer\"},\\"nodeAgent\":"$ref\":"
\":"#/definitions/NodeAgent\"},\\"readinessProbeTimeoutSeconds\":"type\":"integer
\"},\\"resources\":"$ref\":"#/definitions/Resources\"},\\"tolerations\":"$ref
\":"#/definitions/Tolerations\"}},\\"title\":"VpcCni\","type\":"object\"}},
\\"description\":"vpc-cni\"}
}

```

Beispiel 2: Verfügbare Konfigurationsoptionen beim Erstellen oder Aktualisieren von Amazon- CoreDNS AddOns

Das folgende `describe-addon-configuration` Beispiel gibt das gesamte verfügbare Konfigurationsschema zurück, das Sie verwenden, wenn ein Add-on für das CoreDNS-Add-on mit der entsprechenden Version erstellt oder aktualisiert wird.

```

aws eks describe-addon-configuration \
  --addon-name coredns \

```

```
--addon-version v1.8.7-eksbuild.4
```

Ausgabe:

```
{
  "addonName": "coredns",
  "addonVersion": "v1.8.7-eksbuild.4",
  "configurationSchema": "{\"$ref\":\"#/definitions/CoreDNS\",\"$schema\": \"http://json-schema.org/draft-06/schema#\", \"definitions\": {\"CoreDNS\": {\"additionalProperties\": false, \"properties\": {\"computeType\": {\"type\": \"string\"}, \"corefile\": {\"description\": \"Entire corefile contents to use with installation\", \"type\": \"string\"}, \"nodeSelector\": {\"additionalProperties\": {\"type\": \"string\"}, \"type\": \"object\"}, \"replicaCount\": {\"type\": \"integer\"}, \"resources\": {\"$ref\": \"#/definitions/Resources\"}}, \"title\": \"CoreDNS\", \"type\": \"object\"}, \"Limits\": {\"additionalProperties\": false, \"properties\": {\"cpu\": {\"type\": \"string\"}, \"memory\": {\"type\": \"string\"}}, \"title\": \"Limits\", \"type\": \"object\"}, \"Resources\": {\"additionalProperties\": false, \"properties\": {\"limits\": {\"$ref\": \"#/definitions/Limits\"}, \"requests\": {\"$ref\": \"#/definitions/Limits\"}}, \"title\": \"Resources\", \"type\": \"object\"}}}"
}
```

Weitere Informationen finden Sie unter [Eine kubeconfig-Datei für einen EKS Amazon-Cluster in Amazon erstellen oder aktualisieren](#). EKS

- API-Einheiten finden Sie unter [DescribeAddonConfiguration AWS CLI Befehlsreferenz](#).

describe-addon-versions

Das folgende Codebeispiel zeigt die Verwendung von `describe-addon-versions`.

AWS CLI

Beispiel 1: Listet alle verfügbaren Addons für Cluster auf EKS

Das folgende `describe-addon-versions` Beispiel listet alle verfügbaren AWS Addons auf.

```
aws eks describe-addon-versions \
  --query 'sort_by(addons &owner)[].{publisher: publisher, owner: owner, \
  addonName: addonName, type: type}' \
  --output table
```

Ausgabe:


```

-----
|
|                                     DescribeAddonVersions
|
+-----+-----+-----+
+-----+-----+-----+
|          addonName          |          owner          |          publisher
|          type                |                          |
+-----+-----+-----+
+-----+-----+-----+
| vpc-cni                      | aws                      | eks
|   | networking                |                          |
| snapshot-controller          | aws                      | eks
|   | storage                   |                          |
| kube-proxy                   | aws                      | eks
|   | networking                |                          |
| eks-pod-identity-agent       | aws                      | eks
|   | security                  |                          |
| coredns                      | aws                      | eks
|   | networking                |                          |
| aws-mountpoint-s3-csi-driver | aws                      | s3
|   | storage                   |                          |
| aws-guardduty-agent          | aws                      | eks
|   | security                  |                          |
| aws-efs-csi-driver           | aws                      | eks
|   | storage                   |                          |
| aws-ebs-csi-driver           | aws                      | eks
|   | storage                   |                          |
| amazon-cloudwatch-observability | aws                      | eks
|   | observability             |                          |
| adot                          | aws                      | eks
|   | observability             |                          |
| upwind-security_upwind-operator | aws-marketplace        | Upwind Security
|   | security                  |                          |
| upbound_universal-crossplane  | aws-marketplace        | upbound
|   | infra-management          |                          |
| tetrade-io_istio-distro       | aws-marketplace        | tetrade-io
|   | policy-management         |                          |
| teleport_teleport             | aws-marketplace        | teleport
|   | policy-management         |                          |
| stormforge_optimize-live      | aws-marketplace        | StormForge
|   | cost-management           |                          |
| splunk_splunk-otel-collector-chart | aws-marketplace        | Splunk
|   | monitoring                |

```

```

| solo-io_istio-distro | aws-marketplace | Solo.io
|   service-mesh      |
| rafay-systems_rafay-operator | aws-marketplace | rafay-systems
|   kubernetes-management |
| new-relic_kubernetes-operator | aws-marketplace | New Relic
|   observability      |
| netapp_trident-operator | aws-marketplace | NetApp Inc.
|   storage            |
| leaksignal_leakagent | aws-marketplace | leaksignal
|   monitoring         |
| kubecost_kubecost | aws-marketplace | kubecost
|   cost-management   |
| kong_konnect-ri | aws-marketplace | kong
|   ingress-service-type |
| kasten_k10 | aws-marketplace | Kasten by Veeam
|   data-protection |
| haproxy-technologies_kubernetes-ingress-ee | aws-marketplace | HAProxy
Technologies | ingress-controller |
| groundcover_agent | aws-marketplace | groundcover
|   monitoring      |
| grafana-labs_kubernetes-monitoring | aws-marketplace | Grafana Labs
|   monitoring     |
| factorhouse_kpow | aws-marketplace | factorhouse
|   monitoring     |
| dynatrace_dynatrace-operator | aws-marketplace | dynatrace
|   monitoring     |
| datree_engine-pro | aws-marketplace | datree
|   policy-management |
| datadog_operator | aws-marketplace | Datadog
|   monitoring     |
| cribl_cribledge | aws-marketplace | Cribl
|   observability  |
| calyptia_fluent-bit | aws-marketplace | Calyptia Inc
|   observability  |
| accuknox_kubearmor | aws-marketplace | AccuKnox
|   security       |
+-----+-----+
+-----+-----+

```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 2: Listet alle verfügbaren Addons für die angegebene Kubernetes-Version auf, die unterstützt werden für EKS

Das folgende `describe-addon-versions` Beispiel listet alle verfügbaren Addons für die angegebene Kubernetes-Version auf, für die sie unterstützt werden. EKS

```
aws eks describe-addon-versions \
  --kubernetes-version=1.26 \
  --query 'sort_by(addons &owner)[].{publisher: publisher, owner: owner,
  addonName: addonName, type: type}' \
  --output table
```

Ausgabe:

```
-----
|                                     DescribeAddonVersions
|                                     |
+-----+-----+-----+-----+
|                                     |                                     |
|          addonName                 |          owner          |          publisher
|          | type                    |          |               |
+-----+-----+-----+-----+
| vpc-cni                            | aws                    | eks
|   | networking                    |
| snapshot-controller                | aws                    | eks
|   | storage                       |
| kube-proxy                         | aws                    | eks
|   | networking                    |
| eks-pod-identity-agent              | aws                    | eks
|   | security                      |
| coredns                            | aws                    | eks
|   | networking                    |
| aws-mountpoint-s3-csi-driver        | aws                    | s3
|   | storage                       |
| aws-guardduty-agent                 | aws                    | eks
|   | security                      |
| aws-efs-csi-driver                  | aws                    | eks
|   | storage                       |
| aws-ebs-csi-driver                  | aws                    | eks
|   | storage                       |
| amazon-cloudwatch-observability     | aws                    | eks
|   | observability                 |
-----
```

```

| adot | aws | eks
| | observability |
| upwind-security_upwind-operator | aws-marketplace | Upwind Security
| | security |
| tetrade-io_istio-distro | aws-marketplace | tetrade-io
| | policy-management |
| stormforge_optimize-live | aws-marketplace | StormForge
| | cost-management |
| splunk_splunk-otel-collector-chart | aws-marketplace | Splunk
| | monitoring |
| solo-io_istio-distro | aws-marketplace | Solo.io
| | service-mesh |
| rafay-systems_rafay-operator | aws-marketplace | rafay-systems
| | kubernetes-management |
| new-relic_kubernetes-operator | aws-marketplace | New Relic
| | observability |
| netapp_trident-operator | aws-marketplace | NetApp Inc.
| | storage |
| leaksignal_leakagent | aws-marketplace | leaksignal
| | monitoring |
| kubecost_kubecost | aws-marketplace | kubecost
| | cost-management |
| kong_konnect-ri | aws-marketplace | kong
| | ingress-service-type |
| haproxy-technologies_kubernetes-ingress-ee | aws-marketplace | HAProxy
Technologies | ingress-controller |
| groundcover_agent | aws-marketplace | groundcover
| | monitoring |
| grafana-labs_kubernetes-monitoring | aws-marketplace | Grafana Labs
| | monitoring |
| dynatrace_dynatrace-operator | aws-marketplace | dynatrace
| | monitoring |
| datadog_operator | aws-marketplace | Datadog
| | monitoring |
| cribl_cribledge | aws-marketplace | Cribl
| | observability |
| calyptia_fluent-bit | aws-marketplace | Calyptia Inc
| | observability |
| accuknox_kubearmor | aws-marketplace | AccuKnox
| | security |
+-----+-----+
+-----+-----+

```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 3: Listet alle verfügbaren vpc-cni-Addons-Versionen für die angegebene Kubernetes-Version auf, für die unterstützt wird EKS

Das folgende `describe-addon-versions` Beispiel listet alle verfügbaren vpc-cni-Addons-Versionen für die angegebene Kubernetes-Version auf, für die sie unterstützt werden. EKS

```
aws eks describe-addon-versions \
  --kubernetes-version=1.26 \
  --addon-name=vpc-cni \
  --query='addons[].addonVersions[].addonVersion'
```

Ausgabe:

```
[
  "v1.18.0-eksbuild.1",
  "v1.17.1-eksbuild.1",
  "v1.16.4-eksbuild.2",
  "v1.16.3-eksbuild.2",
  "v1.16.2-eksbuild.1",
  "v1.16.0-eksbuild.1",
  "v1.15.5-eksbuild.1",
  "v1.15.4-eksbuild.1",
  "v1.15.3-eksbuild.1",
  "v1.15.1-eksbuild.1",
  "v1.15.0-eksbuild.2",
  "v1.14.1-eksbuild.1",
  "v1.14.0-eksbuild.3",
  "v1.13.4-eksbuild.1",
  "v1.13.3-eksbuild.1",
  "v1.13.2-eksbuild.1",
  "v1.13.0-eksbuild.1",
  "v1.12.6-eksbuild.2",
  "v1.12.6-eksbuild.1",
  "v1.12.5-eksbuild.2",
  "v1.12.0-eksbuild.2"
]
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on erstellen](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAddonVersions](#) in der AWS CLI Befehlsreferenz.

describe-addon

Das folgende Codebeispiel zeigt die Verwendung von `describe-addon`.

AWS CLI

Beschreiben Sie das aktiv ausgeführte EKS Addon in Ihrem Amazon-Cluster EKS

Im folgenden `describe-addon` Beispiel wird das EKS Addon aktiv in Ihrem EKS Amazon-Cluster ausgeführt.

```
aws eks describe-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni
```

Ausgabe:

```
{
  "addon": {
    "addonName": "vpc-cni",
    "clusterName": "my-eks-cluster",
    "status": "ACTIVE",
    "addonVersion": "v1.16.4-eksbuild.2",
    "health": {
      "issues": []
    },
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f",
    "createdAt": "2024-03-14T13:18:45.417000-04:00",
    "modifiedAt": "2024-03-14T13:18:49.557000-04:00",
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-Yfakrq0C1UTm",
    "tags": {
      "eks-addon-key-3": "value-3",
      "eks-addon-key-4": "value-4"
    },
    "configurationValues": "resources:\n      limits:\n        cpu: '100m'\n      nenv:\n        AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
  }
}
```

- API-Einheiten finden Sie [DescribeAddon](#) in der AWS CLI Befehlsreferenz.

describe-cluster

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster`.

AWS CLI

Beschreiben Sie das aktiv ausgeführte EKS Addon in Ihrem Amazon-Cluster EKS

Im folgenden `describe-cluster` Beispiel wird das EKS Addon aktiv in Ihrem EKS Amazon-Cluster ausgeführt.

```
aws eks describe-cluster \
  --cluster-name my-eks-cluster
```

Ausgabe:

```
{
  "cluster": {
    "name": "my-eks-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
    "createdAt": "2024-03-14T11:31:44.348000-04:00",
    "version": "1.26",
    "endpoint": "https://JSA79429HJDASKJDJ8223829MNDNASW.y14.us-
east-2.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0fb75d2d8401716e7",
        "subnet-02184492f67a3d0f9",
        "subnet-04098063527aab776",
        "subnet-0e2907431c9988b72",
        "subnet-04ad87f71c6e5ab4d",
        "subnet-09d912bb63ef21b9a"
      ],
      "securityGroupIds": [
        "sg-0c1327f6270afbb36"
      ],
      "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
      "vpcId": "vpc-0012b8e1cc0abb17d",
      "endpointPublicAccess": true,
    }
  }
}
```

```
    "endpointPrivateAccess": true,
    "publicAccessCidrs": [
      "22.19.18.2/32"
    ]
  },
  "kubernetesNetworkConfig": {
    "serviceIpv4Cidr": "10.100.0.0/16",
    "ipFamily": "ipv4"
  },
  "logging": {
    "clusterLogging": [
      {
        "types": [
          "api",
          "audit",
          "authenticator",
          "controllerManager",
          "scheduler"
        ],
        "enabled": true
      }
    ]
  },
  "identity": {
    "oidc": {
      "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
JSA79429HJDASKJDJ8223829MNDNASW"
    }
  },
  "status": "ACTIVE",
  "certificateAuthority": {
    "data": "CA_DATA_STRING..."
  },
  "platformVersion": "eks.14",
  "tags": {
    "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
    "alpha.eksctl.io/cluster-name": "my-eks-cluster",
    "karpenter.sh/discovery": "my-eks-cluster",
    "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
    "auto-delete": "no",
    "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
    "EKS-Cluster-Name": "my-eks-cluster",
```



```

        "alpha.eksctl.io/cluster-oidc-enabled": "true",
        "aws:cloudformation:logical-id": "ControlPlane",
        "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
        "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
    },
    "health": {
        "issues": []
    },
    "accessConfig": {
        "authenticationMode": "API_AND_CONFIG_MAP"
    }
}
}

```

- API-Einheiten finden Sie [DescribeCluster](#) in der AWS CLI Befehlsreferenz.

describe-fargate-profile

Das folgende Codebeispiel zeigt die Verwendung `describe-fargate-profile`.

AWS CLI

Beschreiben Sie ein Fargate-Profil

Das folgende `describe-fargate-profile` Beispiel beschreibt ein Fargate-Profil.

```

aws eks describe-fargate-profile \
  --cluster-name my-eks-cluster \
  --fargate-profile-name my-fargate-profile

```

Ausgabe:

```

{
  "fargateProfile": {
    "fargateProfileName": "my-fargate-profile",
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/my-fargate-profile/96c766ce-43d2-f9c9-954c-647334391198",
    "clusterName": "my-eks-cluster",
    "createdAt": "2024-04-11T10:42:52.486000-04:00",
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-farga-FargatePodExecutionRole-1htfAaJdJUE0",
  }
}

```

```

    "subnets": [
      "subnet-09d912bb63ef21b9a",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-0e2907431c9988b72"
    ],
    "selectors": [
      {
        "namespace": "prod*",
        "labels": {
          "labelname*?": "*value1"
        }
      },
      {
        "namespace": "*dev*",
        "labels": {
          "labelname*?": "*value*"
        }
      }
    ],
    "status": "ACTIVE",
    "tags": {
      "eks-fargate-profile-key-2": "value-2",
      "eks-fargate-profile-key-1": "value-1"
    }
  }
}

```

- API-Einheiten finden Sie [DescribeFargateProfile](#) in der AWS CLI Befehlsreferenz.

describe-identity-provider-config

Das folgende Codebeispiel zeigt die Verwendung `describe-identity-provider-config`.

AWS CLI

Beschreiben Sie eine Identitätsanbieter-Konfiguration, die mit Ihrem EKS Amazon-Cluster verknüpft ist.

Das folgende `describe-identity-provider-config` Beispiel beschreibt eine Identitätsanbieter-Konfiguration, die mit Ihrem EKS Amazon-Cluster verknüpft ist.

```

aws eks describe-identity-provider-config \
  --cluster-name my-eks-cluster \

```

```
--identity-provider-config type=oidc,name=my-identity-provider
```

Ausgabe:

```
{
  "identityProviderConfig": {
    "oidc": {
      "identityProviderConfigName": "my-identity-provider",
      "identityProviderConfigArn": "arn:aws:eks:us-
east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-
provider/8ac76722-78e4-cec1-ed76-d49eea058622",
      "clusterName": "my-eks-cluster",
      "issuerUrl": "https://oidc.eks.us-east-2.amazonaws.com/
id/38D6A4619A0A69E342B113ED7F1A7652",
      "clientId": "kubernetes",
      "usernameClaim": "email",
      "usernamePrefix": "my-username-prefix",
      "groupsClaim": "my-claim",
      "groupsPrefix": "my-groups-prefix",
      "requiredClaims": {
        "Claim1": "value1",
        "Claim2": "value2"
      },
      "tags": {
        "env": "dev"
      },
      "status": "ACTIVE"
    }
  }
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern für Ihren Cluster über einen OpenID Connect-Identitätsanbieter](#) im EKSAWS-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeIdentityProviderConfig](#) in der AWS CLI Befehlsreferenz.

describe-nodegroup

Das folgende Codebeispiel zeigt die Verwendung `describe-nodegroup`.

AWS CLI

Beschreiben Sie eine verwaltete Knotengruppe für einen EKS Amazon-Cluster

Das folgende `describe-nodegroup` Beispiel beschreibt eine verwaltete Knotengruppe für einen EKS Amazon-Cluster.

```
aws eks describe-nodegroup \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup
```

Ausgabe:

```
{  
  "nodegroup": {  
    "nodegroupName": "my-eks-nodegroup",  
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-  
cluster/my-eks-nodegroup/a8c75f2f-df78-a72f-4063-4b69af3de5b1",  
    "clusterName": "my-eks-cluster",  
    "version": "1.26",  
    "releaseVersion": "1.26.12-20240329",  
    "createdAt": "2024-04-08T11:42:10.555000-04:00",  
    "modifiedAt": "2024-04-08T11:44:12.402000-04:00",  
    "status": "ACTIVE",  
    "capacityType": "ON_DEMAND",  
    "scalingConfig": {  
      "minSize": 1,  
      "maxSize": 3,  
      "desiredSize": 1  
    },  
    "instanceTypes": [  
      "t3.medium"  
    ],  
    "subnets": [  
      "subnet-0e2907431c9988b72",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-09d912bb63ef21b9a"  
    ],  
    "amiType": "AL2_x86_64",  
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",  
    "labels": {},  
    "resources": {  
      "autoScalingGroups": [  
        {  
          "name": "eks-my-eks-nodegroup-a8c75f2f-df78-  
a72f-4063-4b69af3de5b1"  
        }  
      ]  
    }  
  }  
}
```

```

    ]
  },
  "diskSize": 20,
  "health": {
    "issues": []
  },
  "updateConfig": {
    "maxUnavailable": 1
  },
  "tags": {}
}

```

- API-Einheiten finden Sie [DescribeNodegroup](#) in der AWS CLI Befehlsreferenz.

describe-update

Das folgende Codebeispiel zeigt die Verwendung `describe-update`.

AWS CLI

Beispiel 1: Um ein Update für einen Cluster zu beschreiben

Das folgende `describe-update` Beispiel beschreibt ein Update für einen Cluster mit dem Namen.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161

```

Ausgabe:

```

{
  "update": {
    "id": "10bddb13-a71b-425a-b0a6-71cd03e59161",
    "status": "Successful",
    "type": "EndpointAccessUpdate",
    "params": [
      {
        "type": "EndpointPublicAccess",
        "value": "false"
      }
    ]
  }
}

```

```

    },
    {
      "type": "EndpointPrivateAccess",
      "value": "true"
    }
  ],
  "createdAt": "2024-03-14T10:01:26.297000-04:00",
  "errors": []
}
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer EKS Amazon-Cluster-Kubernetes-Version](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 2: Um ein Update für einen Cluster zu beschreiben

Das folgende `describe-update` Beispiel beschreibt ein Update für einen Cluster mit dem Namen.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id e4994991-4c0f-475a-a040-427e6da52966

```

Ausgabe:

```

{
  "update": {
    "id": "e4994991-4c0f-475a-a040-427e6da52966",
    "status": "Successful",
    "type": "AssociateEncryptionConfig",
    "params": [
      {
        "type": "EncryptionConfig",
        "value": "[{\\"resources\\":[\"secrets\\\"],\\"provider\\":{\\"keyArn\\":
\\\"arn:aws:kms:region-code:account:key/key\\\"}}]"
      }
    ],
    "createdAt": "2024-03-14T11:01:26.297000-04:00",
    "errors": []
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer EKS Amazon-Cluster-Kubernetes-Version](#) im EKSAAmazon-Benutzerhandbuch.

Beispiel 3: Um ein Update für einen Cluster zu beschreiben

Das folgende describe-update Beispiel beschreibt ein Update für einen Cluster mit dem Namen.

```
aws eks describe-update \  
  --name my-eks-cluster \  
  --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

Ausgabe:

```
{  
  "update": {  
    "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",  
    "status": "Successful",  
    "type": "VersionUpdate",  
    "params": [  
      {  
        "type": "Version",  
        "value": "1.29"  
      },  
      {  
        "type": "PlatformVersion",  
        "value": "eks.1"  
      }  
    ],  
    "createdAt": "2024-03-14T12:05:26.297000-04:00",  
    "errors": []  
  }  
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer EKS Amazon-Cluster-Kubernetes-Version](#) im EKSAAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeUpdate](#) in der AWS CLI Befehlsreferenz.

disassociate-identity-provider-config

Das folgende Codebeispiel zeigt die Verwendung `disassociate-identity-provider-config`.

AWS CLI

Trennen Sie den Identitätsanbieter von Ihrem Amazon-Cluster EKS

Im folgenden `disassociate-identity-provider-config` Beispiel wird die Zuordnung eines Identitätsanbieters zu Ihrem EKS Amazon-Cluster aufgehoben.

```
aws eks disassociate-identity-provider-config \  
  --cluster-name my-eks-cluster \  
  --identity-provider-config 'type=oidc,name=my-identity-provider'
```

Ausgabe:

```
{  
  "update": {  
    "id": "5f78d14e-c57b-4857-a3e4-cf664ae20949",  
    "status": "InProgress",  
    "type": "DisassociateIdentityProviderConfig",  
    "params": [  
      {  
        "type": "IdentityProviderConfig",  
        "value": "[]"  
      }  
    ],  
    "createdAt": "2024-04-11T13:53:43.314000-04:00",  
    "errors": []  
  }  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern für Ihren Cluster von einem OpenID Connect-Identitätsanbieter — Trennen Sie einen OIDC Identitätsanbieter von Ihrem Cluster](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateIdentityProviderConfig](#) in AWS CLI der Befehlsreferenz.

get-token

Das folgende Codebeispiel zeigt die Verwendung `get-token`.

AWS CLI

Beispiel 1: Holen Sie sich ein Authentifizierungstoken für einen EKS Amazon-Cluster mit dem Namen `my-eks-cluster`

Im folgenden `get-token` Beispiel wird ein Authentifizierungstoken für einen EKS Amazon-Cluster mit dem Namen `my-eks-cluster` abgerufen.

```
aws eks get-token \
  --cluster-name my-eks-cluster
```

Ausgabe:

```
{
  "kind": "ExecCredential",
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "spec": {},
  "status": {
    "expirationTimestamp": "2024-04-11T20:59:56Z",
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
  }
}
```

Beispiel 2: Ruft ein Authentifizierungstoken für einen EKS Amazon-Cluster mit dem Namen `my-eks-cluster` ab, indem beim Signieren des Tokens diese Rolle ARN für Anmeldeinformationen übernommen wird

Im folgenden `get-token` Beispiel wird ein Authentifizierungstoken für einen EKS Amazon-Cluster abgerufen, der benannt `my-eks-cluster` wird, indem beim Signieren des Tokens diese Rolle ARN für Anmeldeinformationen übernommen wird.

```
aws eks get-token \
  --cluster-name my-eks-cluster \
  --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-
  cluster-ServiceRole-j1k7AfTIQtnM
```

Ausgabe:

```
{
```

```
"kind": "ExecCredential",
"apiVersion": "client.authentication.k8s.io/v1beta1",
"spec": {},
"status": {
  "expirationTimestamp": "2024-04-11T21:05:26Z",
  "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
}
}
```

- API-Einheiten finden Sie [GetToken](#) unter AWS CLI Befehlsreferenz.

list-addons

Das folgende Codebeispiel zeigt die Verwendung `list-addons`.

AWS CLI

Listet alle installierten Add-Ons in Ihrem EKS Amazon-Cluster mit dem Namen `my-eks-cluster` auf

Das folgende `list-addons` Beispiel listet alle installierten Add-Ons in Ihrem EKS Amazon-Cluster mit dem Namen auf `my-eks-cluster`.

```
aws eks list-addons \
  --cluster-name my-eks-cluster
```

Ausgabe:

```
{
  "addons": [
    "kube-proxy",
    "vpc-cni"
  ]
}
```

- API-Einheiten finden Sie [ListAddons](#) in der AWS CLI Befehlsreferenz.

list-clusters

Das folgende Codebeispiel zeigt die Verwendung `list-clusters`.

AWS CLI

Um alle installierten Add-Ons in Ihrem EKS Amazon-Cluster mit dem Namen `my-eks-cluster` aufzulisten

Das folgende `list-clusters` Beispiel listet alle installierten Add-Ons in Ihrem EKS Amazon-Cluster mit dem Namen auf `my-eks-cluster`.

```
aws eks list-clusters
```

Ausgabe:

```
{
  "clusters": [
    "prod",
    "qa",
    "stage",
    "my-eks-cluster"
  ]
}
```

- API-Einheiten finden Sie [ListClusters](#) in der AWS CLI Befehlsreferenz.

`list-fargate-profiles`

Das folgende Codebeispiel zeigt die Verwendung `list-fargate-profiles`.

AWS CLI

Um alle Fargate-Profilen in Ihrem EKS Amazon-Cluster mit dem Namen `my-eks-cluster` aufzulisten

Das folgende `list-fargate-profiles` Beispiel listet alle Fargate-Profilen in Ihrem EKS Amazon-Cluster mit dem Namen `my-eks-cluster` auf.

```
aws eks list-fargate-profiles \
  --cluster-name my-eks-cluster
```

Ausgabe:

```
{
```

```
"fargateProfileNames": [  
    "my-fargate-profile"  
]  
}
```

- API-Einheiten finden Sie [ListFargateProfiles](#) in der AWS CLI Befehlsreferenz.

list-identity-provider-configs

Das folgende Codebeispiel zeigt die Verwendung `list-identity-provider-configs`.

AWS CLI

Identity-Provider auflisten, die einem EKS Amazon-Cluster zugeordnet sind

Das folgende `list-identity-provider-configs` Beispiel listet den Identitätsanbieter auf, der einem EKS Amazon-Cluster zugeordnet ist.

```
aws eks list-identity-provider-configs \  
    --cluster-name my-eks-cluster
```

Ausgabe:

```
{  
  "identityProviderConfigs": [  
    {  
      "type": "oidc",  
      "name": "my-identity-provider"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern für Ihren Cluster über einen OpenID Connect-Identitätsanbieter](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ListIdentityProviderConfigs](#) in der AWS CLI Befehlsreferenz.

list-nodegroups

Das folgende Codebeispiel zeigt die Verwendung `list-nodegroups`.

AWS CLI

Alle Knotengruppen in einem EKS Amazon-Cluster auflisten

Das folgende `list-nodegroups` Beispiel listet alle Knotengruppen in einem EKS Amazon-Cluster auf.

```
aws eks list-nodegroups \  
  --cluster-name my-eks-cluster
```

Ausgabe:

```
{  
  "nodegroups": [  
    "my-eks-managed-node-group",  
    "my-eks-nodegroup"  
  ]  
}
```

- API-Einheiten finden Sie [ListNodegroups](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Beispiel 1: Um alle Tags für einen EKS Amazon-Cluster aufzulisten ARN

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für einen EKS Amazon-Cluster auf ARN.

```
aws eks list-tags-for-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

Ausgabe:

```
{  
  "tags": {  
    "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",  
    "alpha.eksctl.io/cluster-name": "my-eks-cluster",  
  }  
}
```

```

    "karpenter.sh/discovery": "my-eks-cluster",
    "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
    "auto-delete": "no",
    "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
    "EKS-Cluster-Name": "my-eks-cluster",
    "alpha.eksctl.io/cluster-oidc-enabled": "true",
    "aws:cloudformation:logical-id": "ControlPlane",
    "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
    "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
  }
}

```

Beispiel 2: Um alle Tags für eine Amazon EKS Node-Gruppe aufzulisten ARN

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für eine Amazon EKS Node-Gruppe auf ARN.

```

aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-
eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c

```

Ausgabe:

```

{
  "tags": {
    "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-nodegroup-my-eks-
managed-node-group",
    "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-nodegroup-my-eks-managed-node-group/
eaa20310-e219-11ee-b851-0ab9ad8228ff",
    "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
    "EKS-Cluster-Name": "my-eks-cluster",
    "alpha.eksctl.io/nodegroup-type": "managed",
    "NodeGroup Name 1": "my-eks-managed-node-group",
    "k8s.io/cluster-autoscaler/enabled": "true",
    "nodegroup-role": "worker",
    "alpha.eksctl.io/cluster-name": "my-eks-cluster",
    "alpha.eksctl.io/nodegroup-name": "my-eks-managed-node-group",
    "karpenter.sh/discovery": "my-eks-cluster",
    "NodeGroup Name 2": "AmazonLinux-Linux-Managed-NG-v1-26-v1",
  }
}

```

```

    "auto-delete": "no",
    "k8s.io/cluster-autoscaler/my-eks-cluster": "owned",
    "aws:cloudformation:logical-id": "ManagedNodeGroup",
    "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z"
  }
}

```

Beispiel 3: Um alle Tags in einem Amazon EKS Fargate-Profil aufzulisten ARNe

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für ein Amazon EKS Fargate-Profil aufARN.

```

aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/
my-fargate-profile/d6c76780-e541-0725-c816-36754cab734b

```

Ausgabe:

```

{
  "tags": {
    "eks-fargate-profile-key-2": "value-2",
    "eks-fargate-profile-key-1": "value-1"
  }
}

```

Beispiel 4: Um alle Tags für ein EKS Amazon-Add-on aufzulisten ARN

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für ein EKS Amazon-Add-on aufARN.

```

aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-
cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f

```

Ausgabe:

```

{
  "tags": {
    "eks-addon-key-2": "value-2",
    "eks-addon-key-1": "value-1"
  }
}

```

```
}  
}
```

Beispiel 5: Um alle Tags für einen EKS OIDC Amazon-Identitätsanbieter aufzulisten ARN

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für einen EKS OIDC Amazon-Identitätsanbieter auf ARN.

```
aws eks list-tags-for-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:identityproviderconfig/my-eks-  
cluster/oidc/my-identity-provider/8ac76722-78e4-cec1-ed76-d49eea058622
```

Ausgabe:

```
{  
  "tags": {  
    "my-identity-provider": "test"  
  }  
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-update

Das folgende Codebeispiel zeigt die Verwendung `list-update`.

AWS CLI

Beispiel 1: Um die Updates aufzulisten, die mit einem EKS Amazon-Clusternamen verknüpft sind

Das folgende `list-updates` Beispiel listet alle Updates IDs für einen EKS Amazon-Clusternamen auf.

```
aws eks list-updates \  
  --name my-eks-cluster
```

Ausgabe:

```
{  
  "updateIds": [  
    "
```



```
    "5f78d14e-c57b-4857-a3e4-cf664ae20949",  
    "760e5a3f-adad-48c7-88d3-7ac283c09c26",  
    "cd4ec863-bc55-47d5-a377-3971502f529b",  
    "f12657ce-e869-4f17-b158-a82ab8b7d937"  
  ]  
}
```

Beispiel 2: Um alle Updates IDs für eine Amazon EKS Node-Gruppe aufzulisten

Das folgende `list-updates` Beispiel listet alle Updates IDs für eine Amazon EKS Node-Gruppe auf.

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --nodegroup-name my-eks-managed-node-group
```

Ausgabe:

```
{  
  "updateIds": [  
    "8c6c1bef-61fe-42ac-a242-89412387b8e7"  
  ]  
}
```

Beispiel 3: Um alle Updates auf einem IDs EKS Amazon-Add-on aufzulisten

Das folgende `list-updates` Beispiel listet alle Updates IDs für ein EKS Amazon-Add-on auf.

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --addon-name vpc-cni
```

Ausgabe:

```
{  
  "updateIds": [  
    "9cdba8d4-79fb-3c83-afe8-00b508d33268"  
  ]  
}
```

- API-Einheiten finden Sie [ListUpdate](#) in der AWS CLI Befehlsreferenz.

list-updates

Das folgende Codebeispiel zeigt die Verwendung `list-updates`.

AWS CLI

Um die Updates für einen Cluster aufzulisten

Dieser Beispielbefehl listet die aktuellen Updates für einen Cluster auf, der `example` in Ihrer Standardregion benannt ist.

Befehl:

```
aws eks list-updates --name example
```

Ausgabe:

```
{
  "updateIds": [
    "10bddb13-a71b-425a-b0a6-71cd03e59161"
  ]
}
```

- API-Einheiten finden Sie [ListUpdates](#) in der AWS CLI Befehlsreferenz.

register-cluster

Das folgende Codebeispiel zeigt die Verwendung `register-cluster`.

AWS CLI

Beispiel 1: Registrieren Sie einen externen EKS `_ ANYWHERE` Kubernetes-Cluster bei Amazon EKS

Das folgende `register-cluster` Beispiel registriert einen externen EKS `_ ANYWHERE` Kubernetes-Cluster bei Amazon. EKS

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/AmazonEKSClusterAgentRole,provider=EKS_ANYWHERE'
```

Ausgabe:

```
{
  "cluster": {
    "name": "my-eks-anywhere-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
    "createdAt": "2024-04-12T12:38:37.561000-04:00",
    "status": "PENDING",
    "tags": {},
    "connectorConfig": {
      "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
      "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
      "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
      "provider": "EKS_ANYWHERE",
      "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSCollectorAgentRole"
    }
  }
}
```

Weitere Informationen finden Sie unter [Einen externen Cluster verbinden](#) im EKSAmerican-Benutzerhandbuch.

Beispiel 2: Registrieren Sie einen beliebigen externen Kubernetes-Cluster bei Amazon EKS

Das folgende `register-cluster` Beispiel registriert einen externen EKS _ ANYWHERE Kubernetes-Cluster bei Amazon. EKS

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/AmazonEKSCollectorAgentRole,provider=OTHER'
```

Ausgabe:

```
{
  "cluster": {
    "name": "my-onprem-k8s-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-onprem-k8s-cluster",
    "createdAt": "2024-04-12T12:42:10.861000-04:00",
    "status": "PENDING",
    "tags": {},
    "connectorConfig": {
```

```

    "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
    "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
    "activationExpiry": "2024-04-15T12:42:10.339000-04:00",
    "provider": "OTHER",
    "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole"
  }
}
}

```

Weitere Informationen finden Sie unter [Einen externen Cluster verbinden](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterCluster](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Beispiel 1: Um die angegebenen Tags zu einem EKS Amazon-Cluster hinzuzufügen

Das folgende `tag-resource` Beispiel fügt die angegebenen Tags zu einem EKS Amazon-Cluster hinzu.

```

aws eks tag-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \
  --tag 'my-eks-cluster-test-1=test-value-1,my-eks-cluster-dev-1=dev-value-2'

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die angegebenen Tags zu einer Amazon EKS Node-Gruppe hinzuzufügen

Das folgende `tag-resource` Beispiel fügt die angegebenen Tags zu einer Amazon EKS Node-Gruppe hinzu.

```

aws eks tag-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \
  --tag 'my-eks-nodegroup-test-1=test-value-1,my-eks-nodegroup-dev-1=dev-value-2'

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Beispiel 1: Um die angegebenen Tags aus einem EKS Amazon-Cluster zu löschen

Das folgende `untag-resource` Beispiel löscht die angegebenen Tags aus einem EKS Amazon-Cluster.

```
aws eks untag-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \  
  --tag-keys "my-eks-cluster-test-1" "my-eks-cluster-dev-1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die angegebenen Tags aus einer Amazon EKS Node-Gruppe zu löschen

Das folgende `untag-resource` Beispiel löscht die angegebenen Tags aus einer Amazon EKS Node-Gruppe.

```
aws eks untag-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \  
  --tag-keys "my-eks-nodegroup-test-1" "my-eks-nodegroup-dev-1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-addon

Das folgende Codebeispiel zeigt die Verwendung `update-addon`.

AWS CLI

Beispiel 1. Um ein EKS Amazon-Add-on mit Servicekontrolle zu aktualisieren ARN

Der folgende `update-addon` Beispielbefehl aktualisiert ein EKS Amazon-Add-on mit der DienstkontorolleARN.

```
aws eks update-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name vpc-cni \  
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-  
addon-vpc-cni-Role1-Yfakrq0C1UTm
```

Ausgabe:

```
{  
  "update": {  
    "id": "c00d2de2-c2e4-3d30-929e-46b8edec2ce4",  
    "status": "InProgress",  
    "type": "AddonUpdate",  
    "params": [  
      {  
        "type": "ServiceAccountRoleArn",  
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-  
addon-vpc-cni-Role1-Yfakrq0C1UTm"  
      }  
    ],  
    "updatedAt": "2024-04-12T16:04:55.614000-04:00",  
    "errors": []  
  }  
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on aktualisieren](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 2. Um ein EKS Amazon-Add-on mit einer bestimmten Add-On-Version zu aktualisieren

Der folgende `update-addon` Beispielbefehl aktualisiert ein EKS Amazon-Add-on mit einer bestimmten Add-On-Version.

```
aws eks update-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name vpc-cni \  
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-  
addon-vpc-cni-Role1-Yfakrq0C1UTm \  
  --version 1.10.0
```

```
--addon-version v1.16.4-eksbuild.2
```

Ausgabe:

```
{
  "update": {
    "id": "f58dc0b0-2b18-34bd-bc6a-e4abc0011f36",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.16.4-eksbuild.2"
      },
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      }
    ],
    "createdAt": "2024-04-12T16:07:16.550000-04:00",
    "errors": []
  }
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on aktualisieren](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 3. Um ein EKS Amazon-Add-on mit benutzerdefinierten Konfigurationswerten zu aktualisieren und Konfliktdetails zu lösen

Der folgende `update-addon` Beispielbefehl aktualisiert ein EKS Amazon-Add-on mit benutzerdefinierten Konfigurationswerten und Details zur Konfliktlösung.

```
aws eks update-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni \
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm \
  --addon-version v1.16.4-eksbuild.2 \
  --configuration-values '{"resources": {"limits":{"cpu":"100m"}, "requests":
{"cpu":"50m"}}}' \
```

--resolve-conflicts *PRESERVE*

Ausgabe:

```
{
  "update": {
    "id": "cd9f2173-a8d8-3004-a90f-032f14326520",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.16.4-eksbuild.2"
      },
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      },
      {
        "type": "ResolveConflicts",
        "value": "PRESERVE"
      },
      {
        "type": "ConfigurationValues",
        "value": "{\"resources\": {\"limits\": {\"cpu\": \"100m\"}, \"requests
\": {\"cpu\": \"50m\"}}}"
      }
    ],
    "createdAt": "2024-04-12T16:16:27.363000-04:00",
    "errors": []
  }
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on aktualisieren](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 4. Um ein EKS Amazon-Add-on mit einer Datei mit benutzerdefinierten JSON Konfigurationen zu aktualisieren

Der folgende `update-addon` Beispielbefehl aktualisiert ein EKS Amazon-Add-on mit benutzerdefinierten JSON Konfigurationen und Details zur Konfliktlösung.


```
aws eks update-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni \
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm \
  --addon-version v1.17.1-eksbuild.1 \
  --configuration-values 'file://configuration-values.json' \
  --resolve-conflicts PRESERVE
```

Inhalt von `configuration-values.json`:

```
{
  "resources": {
    "limits": {
      "cpu": "100m"
    },
    "requests": {
      "cpu": "50m"
    }
  },
  "env": {
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
  }
}
```

Ausgabe:

```
{
  "update": {
    "id": "6881a437-174f-346b-9a63-6e91763507cc",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.17.1-eksbuild.1"
      },
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      }
    ]
  }
}
```

```

    {
      "type": "ResolveConflicts",
      "value": "PRESERVE"
    },
    {
      "type": "ConfigurationValues",
      "value": "{\n  \"resources\": {\n    \"limits\": {\n\n      \"cpu\": \"100m\"\n    },\n    \"requests\": {\n      \"cpu\": \"50m\"\n    }\n  },\n  \"env\": {\n    \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n  }\n}"
    }
  ],
  "createdAt": "2024-04-12T16:22:55.519000-04:00",
  "errors": []
}
}

```

Weitere Informationen finden Sie unter [EKSAmazon-Add-Ons verwalten — Ein Add-on aktualisieren](#) im EKSAmazon-Benutzerhandbuch.

Beispiel 5. Um ein EKS Amazon-Add-on mit einer Datei mit benutzerdefinierten YAML Konfigurationswerten zu aktualisieren

Der folgende update-addon Beispielbefehl aktualisiert ein EKS Amazon-Add-on mit benutzerdefinierten YAML Konfigurationswerten und Details zur Konfliktlösung.

```

aws eks update-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni \
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-Yfakrq0C1UTm \
  --addon-version v1.18.0-eksbuild.1 \
  --configuration-values 'file://configuration-values.yaml' \
  --resolve-conflicts PRESERVE

```

Inhalt von configuration-values.yaml:

```

resources:
  limits:
    cpu: '100m'
  requests:
    cpu: '50m'

```

```
env:
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

Ausgabe:

```
{
  "update": {
    "id": "a067a4c9-69d0-3769-ace9-d235c5b16701",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.18.0-eksbuild.1"
      },
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      },
      {
        "type": "ResolveConflicts",
        "value": "PRESERVE"
      },
      {
        "type": "ConfigurationValues",
        "value": "resources:\n    limits:\n        cpu: '100m'\n
requests:\n    cpu: '50m'\nenv:\n    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
      }
    ],
    "createdAt": "2024-04-12T16:25:07.212000-04:00",
    "errors": []
  }
}
```

Weitere Informationen finden Sie unter [EKSA Amazon-Add-Ons verwalten — Ein Add-on aktualisieren](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAddon](#) in der AWS CLI Befehlsreferenz.

update-cluster-config

Das folgende Codebeispiel zeigt die Verwendung `update-cluster-config`.

AWS CLI

Um den Cluster-Endpunktzugriff zu aktualisieren

Dieser Beispielbefehl aktualisiert einen Cluster, um den öffentlichen Endpunktzugriff zu deaktivieren und den privaten Endpunktzugriff zu aktivieren.

Befehl:

```
aws eks update-cluster-config --name example \  
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

Ausgabe:

```
{  
  "update": {  
    "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",  
    "status": "InProgress",  
    "type": "EndpointAccessUpdate",  
    "params": [  
      {  
        "type": "EndpointPublicAccess",  
        "value": "false"  
      },  
      {  
        "type": "EndpointPrivateAccess",  
        "value": "true"  
      }  
    ],  
    "createdAt": 1565806986.506,  
    "errors": []  
  }  
}
```

Um die Protokollierung für einen Cluster zu aktivieren

Dieser Beispielbefehl aktiviert alle Protokollierungstypen der Cluster-Steuerungsebene für einen Cluster mit dem Namen *example*.

Befehl:

```
aws eks update-cluster-config --name example \  
--logging-types all
```

```
--logging '{"clusterLogging":[{"types":["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Ausgabe:

```
{
  "update": {
    "id": "7551c64b-1d27-4b1e-9f8e-c45f056eb6fd",
    "status": "InProgress",
    "type": "LoggingUpdate",
    "params": [
      {
        "type": "ClusterLogging",
        "value": "{\"clusterLogging\":{\"types\":[\"api\",\"audit\",\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}}"
      }
    ],
    "createdAt": 1565807210.37,
    "errors": []
  }
}
```

- API-Einheiten finden Sie [UpdateClusterConfig](#) in der AWS CLI Befehlsreferenz.

update-cluster-version

Das folgende Codebeispiel zeigt die Verwendung `update-cluster-version`.

AWS CLI

Um einen EKS Amazon-Cluster mit dem Namen `my-eks-cluster` auf die angegebene Kubernetes-Version zu aktualisieren

Im folgenden `update-cluster-version` Beispiel wird ein EKS Amazon-Cluster auf die angegebene Kubernetes-Version aktualisiert.

```
aws eks update-cluster-version \
  --name my-eks-cluster \
  --kubernetes-version 1.27
```

Ausgabe:

```
{
  "update": {
    "id": "e4091a28-ea14-48fd-a8c7-975aeb469e8a",
    "status": "InProgress",
    "type": "VersionUpdate",
    "params": [
      {
        "type": "Version",
        "value": "1.27"
      },
      {
        "type": "PlatformVersion",
        "value": "eks.16"
      }
    ],
    "createdAt": "2024-04-12T16:56:01.082000-04:00",
    "errors": []
  }
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer EKS Amazon-Cluster-Kubernetes-Version](#) im EKSAAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateClusterVersion](#) in der AWS CLI Befehlsreferenz.

update-kubeconfig

Das folgende Codebeispiel zeigt die Verwendung `update-kubeconfig`.

AWS CLI

Beispiel 1: Konfiguriert Ihr `kubectl`, indem Sie die `kubeconfig` erstellen oder aktualisieren, sodass Sie eine Verbindung zu einem Amazon-Cluster namens `my-eks-cluster` herstellen können.

Im folgenden `update-kubeconfig` Beispiel wird Ihr `kubectl` konfiguriert, indem es die `kubeconfig` erstellt oder aktualisiert, sodass Sie eine Verbindung zu einem Amazon-Cluster mit dem Namen `my-eks-cluster` herstellen können.

```
aws eks update-kubeconfig \
  --name my-eks-cluster
```

Ausgabe:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/xxx/.kube/config
```

Weitere Informationen finden Sie unter [Erstellen oder Aktualisieren einer kubeconfig-Datei für einen EKS Amazon-Cluster](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 2: Konfiguriert Ihr kubectl, indem Sie die kubeconfig erstellen oder aktualisieren (mit der Option `role-arn`, um eine Rolle für die Cluster-Authentifizierung zu übernehmen), sodass Sie eine Verbindung zu einem Amazon-Cluster namens `my-eks-cluster` herstellen können.

Im folgenden `update-kubeconfig` Beispiel wird Ihr kubectl konfiguriert, indem es die kubeconfig erstellt oder aktualisiert (mit der Option `role-arn`, um eine Rolle für die Cluster-Authentifizierung zu übernehmen), sodass Sie eine Verbindung zu einem Amazon-Cluster mit dem Namen `my-eks-cluster` herstellen können.

```
aws eks update-kubeconfig \
  --name my-eks-cluster \
  --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-Cluster-ServiceRole-j1k7AfTIQtnM
```

Ausgabe:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/xxx/.kube/config
```

Weitere Informationen finden Sie unter [Erstellen oder Aktualisieren einer kubeconfig-Datei für einen EKS Amazon-Cluster](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 3: Konfiguriert Ihr kubectl, indem Sie die kubeconfig erstellen oder aktualisieren (mit der Option `role-arn`, um eine Rolle für die Cluster-Authentifizierung zusammen mit einem benutzerdefinierten Cluster-Alias und Benutzer-Alias zu übernehmen), sodass Sie eine Verbindung zu einem Amazon-Cluster namens `my-eks-cluster` herstellen können.

Im folgenden `update-kubeconfig` Beispiel wird Ihr kubectl konfiguriert, indem es die kubeconfig erstellt oder aktualisiert (mit der Option `role-arn`, um eine Rolle für die Cluster-Authentifizierung zusammen mit einem benutzerdefinierten Cluster-Alias und Benutzer-Alias zu übernehmen), sodass Sie eine Verbindung zu einem Amazon-Cluster mit dem Namen `my-eks-cluster` herstellen können.

```
aws eks update-kubeconfig \
```

```
--name my-eks-cluster \  
--role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-  
cluster-ServiceRole-j1k7AfTIQtnM \  
--alias stage-eks-cluster \  
--user-alias john
```

Ausgabe:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

Weitere Informationen finden Sie unter [Erstellen oder Aktualisieren einer kubeconfig-Datei für einen EKS Amazon-Cluster](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 4: Drückt Einträge in der kubeconfig-Datei zur Überprüfung aus und konfiguriert Ihr kubectl so, dass Sie eine Verbindung zu einem Amazon-Cluster namens `` herstellen können EKS my-eks-cluster

Im folgenden update-kubeconfig Beispiel wird Ihr kubectl konfiguriert, indem es die kubeconfig erstellt oder aktualisiert (mit der Option role-arn, um eine Rolle für die Clusterauthentifizierung zusammen mit einem benutzerdefinierten Cluster-Alias und Benutzer-Alias zu übernehmen), sodass Sie eine Verbindung zu einem Amazon-Cluster mit dem Namen herstellen können. EKS my-eks-cluster

```
aws eks update-kubeconfig \  
--name my-eks-cluster \  
--role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-  
cluster-ServiceRole-j1k7AfTIQtnM \  
--alias stage-eks-cluster \  
--user-alias john \  
--verbose
```

Ausgabe:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config  
Entries:  
  
context:  
cluster: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster  
user: john  
name: stage-eks-cluster
```



```
name: john
user:
exec:
  apiVersion: client.authentication.k8s.io/v1beta1
  args:
  - --region
  - us-east-2
  - eks
  - get-token
  - --cluster-name
  - my-eks-cluster
  - --output
  - json
  - --role
  - arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-
ServiceRole-j1k7AfTIQtnM
  command: aws

cluster:
certificate-authority-data: xxx_CA_DATA_xxx
server: https://DALSJ343KE23J3RN45653DSKJTT647TYD.y14.us-east-2.eks.amazonaws.com
name: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

Weitere Informationen finden Sie unter [Erstellen oder Aktualisieren einer kubeconfig-Datei für einen EKS Amazon-Cluster](#) im EKSA Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateKubeconfig](#) in der AWS CLI Befehlsreferenz.

update-nodegroup-config

Das folgende Codebeispiel zeigt die Verwendung `update-nodegroup-config`.

AWS CLI

Beispiel 1: Aktualisieren Sie eine verwaltete Knotengruppe, um dem EKS Worker-Knoten für einen EKS Amazon-Cluster neue Labels und Taint hinzuzufügen

Im folgenden `update-nodegroup-config` Beispiel wird eine verwaltete Knotengruppe aktualisiert, um dem EKS Worker-Knoten für einen EKS Amazon-Cluster neue Labels und Taint hinzuzufügen.

```
aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \
```

```

--nodegroup-name my-eks-nodegroup \
--labels 'addOrUpdateLabels={my-eks-nodegroup-label-1=value-1,my-eks-nodegroup-label-2=value-2}' \
--taints 'addOrUpdateTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'

```

Ausgabe:

```

{
  "update": {
    "id": "e66d21d3-bd8b-3ad1-a5aa-b196dc08c7c1",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [
      {
        "type": "LabelsToAdd",
        "value": "{\"my-eks-nodegroup-label-2\":\"value-2\",\"my-eks-nodegroup-label-1\":\"value-1\"}"
      },
      {
        "type": "TaintsToAdd",
        "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",\"key\":\"taint-key-1\"}]"
      }
    ],
    "createdAt": "2024-04-08T12:05:19.161000-04:00",
    "errors": []
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer verwalteten Knotengruppe](#) im EKSAmazon-Benutzerhandbuch.

Beispiel 2: Aktualisieren Sie eine verwaltete Knotengruppe, um Labels und Makel für den EKS Worker-Knoten eines EKS Amazon-Clusters zu entfernen

Im folgenden update-nodegroup-config Beispiel wird eine verwaltete Knotengruppe aktualisiert, um Labels und Makel für den EKS Worker-Knoten eines EKS Amazon-Clusters zu entfernen.

```

aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \

```

```

--nodegroup-name my-eks-nodegroup \
--labels 'removeLabels=my-eks-nodegroup-label-1, my-eks-nodegroup-label-2' \
--taints 'removeTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'

```

Ausgabe:

```

{
  "update": {
    "id": "67a08692-9e59-3ace-a916-13929f44cec3",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [
      {
        "type": "LabelsToRemove",
        "value": "[\"my-eks-nodegroup-label-1\", \"my-eks-nodegroup-label-2\"]"
      },
      {
        "type": "TaintsToRemove",
        "value": "[{\"effect\": \"NO_EXECUTE\", \"value\": \"taint-value-1\", \"key\": \"taint-key-1\"}]"
      }
    ],
    "createdAt": "2024-04-08T12:17:31.817000-04:00",
    "errors": []
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer verwalteten Knotengruppe](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 3: Aktualisieren Sie eine verwaltete Knotengruppe, um Labels und Taint für den EKS Worker-Knoten für einen EKS Amazon-Cluster zu entfernen und hinzuzufügen

Im folgenden `update-nodegroup-config` Beispiel wird eine verwaltete Knotengruppe aktualisiert, sodass Labels und Taint für den EKS Worker-Knoten eines EKS Amazon-Clusters entfernt und hinzugefügt werden.

```

aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \

```

```
--labels 'addOrUpdateLabels={my-eks-nodegroup-new-label-1=new-value-1,my-eks-nodegroup-new-label-2=new-value-2},removeLabels=my-eks-nodegroup-label-1, my-eks-nodegroup-label-2' \
--taints 'addOrUpdateTaints=[{key=taint-new-key-1,value=taint-new-value-1,effect=PREFER_NO_SCHEDULE}],removeTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'
```

Ausgabe:

```
{
  "update": {
    "id": "4a9c8c45-6ac7-3115-be71-d6412a2339b7",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [
      {
        "type": "LabelsToAdd",
        "value": "{\"my-eks-nodegroup-new-label-1\":\"new-value-1\",\"my-eks-nodegroup-new-label-2\":\"new-value-2\"}"
      },
      {
        "type": "LabelsToRemove",
        "value": "[\"my-eks-nodegroup-label-1\",\"my-eks-nodegroup-label-2\"]"
      },
      {
        "type": "TaintsToAdd",
        "value": "[{\"effect\":\"PREFER_NO_SCHEDULE\",\"value\":\"taint-new-value-1\",\"key\":\"taint-new-key-1\"}]"
      },
      {
        "type": "TaintsToRemove",
        "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\",\"key\":\"taint-key-1\"}]"
      }
    ],
    "createdAt": "2024-04-08T12:30:55.486000-04:00",
    "errors": []
  }
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer verwalteten Knotengruppe](#) im EKSA Amazon-Benutzerhandbuch.

Beispiel 4: Aktualisieren Sie eine verwaltete Knotengruppe, um die Skalierungskonfiguration und die Aktualisierungskonfiguration für den EKS Worker-Knoten für einen Amazon-Cluster zu aktualisieren EKS

Das folgende `update-nodegroup-config` Beispiel aktualisiert eine verwaltete Knotengruppe, um `scaling-config` und `update-config` für den EKS Worker-Knoten für einen Amazon-Cluster zu aktualisieren. EKS

```
aws eks update-nodegroup-config \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --scaling-config minSize=1,maxSize=5,desiredSize=2 \  
  --update-config maxUnavailable=2
```

Ausgabe:

```
{  
  "update": {  
    "id": "a977160f-59bf-3023-805d-c9826e460aea",  
    "status": "InProgress",  
    "type": "ConfigUpdate",  
    "params": [  
      {  
        "type": "MinSize",  
        "value": "1"  
      },  
      {  
        "type": "MaxSize",  
        "value": "5"  
      },  
      {  
        "type": "DesiredSize",  
        "value": "2"  
      },  
      {  
        "type": "MaxUnavailable",  
        "value": "2"  
      }  
    ],  
    "createdAt": "2024-04-08T12:35:17.036000-04:00",  
    "errors": []  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer verwalteten Knotengruppe](#) im EKSAWS-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateNodegroupConfig](#) in der AWS CLI Befehlsreferenz.

update-nodegroup-version

Das folgende Codebeispiel zeigt die Verwendung `update-nodegroup-version`.

AWS CLI

Beispiel 1: Aktualisieren Sie die Kubernetes-Version oder AMI-Version einer von Amazon EKS verwalteten Knotengruppe

Im folgenden `update-nodegroup-version` Beispiel wird die Kubernetes-Version oder AMI-Version einer von Amazon EKS verwalteten Knotengruppe auf die neueste verfügbare Version für Ihren Kubernetes-Cluster aktualisiert.

```
aws eks update-nodegroup-version \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --no-force
```

Ausgabe:

```
{  
  "update": {  
    "id": "a94ebfc3-6bf8-307a-89e6-7dbaa36421f7",  
    "status": "InProgress",  
    "type": "VersionUpdate",  
    "params": [  
      {  
        "type": "Version",  
        "value": "1.26"  
      },  
      {  
        "type": "ReleaseVersion",  
        "value": "1.26.12-20240329"  
      }  
    ],  
  },  
}
```

```
    "createdAt": "2024-04-08T13:16:00.724000-04:00",  
    "errors": []  
  }  
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer verwalteten Knotengruppe](#) im EKSAmazon-Benutzerhandbuch.

Beispiel 2: Aktualisieren Sie die Kubernetes-Version oder AMI Version einer von Amazon EKS verwalteten Knotengruppe

Im folgenden `update-nodegroup-version` Beispiel wird die Kubernetes-Version oder AMI Version einer von Amazon EKS verwalteten Knotengruppe auf die angegebene AMI Release-Version aktualisiert.

```
aws eks update-nodegroup-version \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --kubernetes-version '1.26' \  
  --release-version '1.26.12-20240307' \  
  --no-force
```

Ausgabe:

```
{  
  "update": {  
    "id": "4db06fe1-088d-336b-bdcd-3fdb94995fb7",  
    "status": "InProgress",  
    "type": "VersionUpdate",  
    "params": [  
      {  
        "type": "Version",  
        "value": "1.26"  
      },  
      {  
        "type": "ReleaseVersion",  
        "value": "1.26.12-20240307"  
      }  
    ],  
    "createdAt": "2024-04-08T13:13:58.595000-04:00",  
    "errors": []  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter Aktualisieren einer verwalteten Knotengruppe - <<https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html>>` im EKSAmerican-Benutzerhandbuch.

- APIEinzelheiten finden Sie [UpdateNodegroupVersion](#) in AWS CLI der Befehlsreferenz.

Elastic Beanstalk Beanstalk-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Elastic Beanstalk Aktionen ausführen und gängige Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

abort-environment-update

Das folgende Codebeispiel zeigt die Verwendung `abort-environment-update`.

AWS CLI

Um eine Bereitstellung abzubrechen

Mit dem folgenden Befehl wird die Bereitstellung einer laufenden Anwendungsversion für eine Umgebung mit dem Namen abgebrochen: `my-env`

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- APIEinzelheiten finden Sie [AbortEnvironmentUpdate](#) in der AWS CLI Befehlsreferenz.

check-dns-availability

Das folgende Codebeispiel zeigt die Verwendung `check-dns-availability`.

AWS CLI

Um die Verfügbarkeit eines zu überprüfen CNAME

Der folgende Befehl überprüft die Verfügbarkeit der Subdomain `my-cname.elasticbeanstalk.com`:

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

Ausgabe:

```
{
  "Available": true,
  "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- API-Einheiten finden Sie [CheckDnsAvailability](#) in der AWS CLI Befehlsreferenz.

create-application-version

Das folgende Codebeispiel zeigt die Verwendung `create-application-version`.

AWS CLI

Um eine neue Anwendungsversion zu erstellen

Der folgende Befehl erstellt eine neue Version, „v1“, einer Anwendung mit dem Namen „MyApp“:

```
aws elasticbeanstalk create-application-version --application-name MyApp
--version-label v1 --description MyAppv1 --source-bundle S3Bucket="my-
bucket",S3Key="sample.war" --auto-create-application
```

Die Anwendung wird aufgrund der `auto-create-application` Option automatisch erstellt, falls sie noch nicht existiert. Das Quellpaket ist eine WAR-Datei, die in einem S3-Bucket namens „my-bucket“ gespeichert ist und die Apache Tomcat-Beispielanwendung enthält.

Ausgabe:

```
{
  "ApplicationVersion": {
    "ApplicationName": "MyApp",
    "VersionLabel": "v1",
    "Description": "MyAppv1",
    "DateCreated": "2015-02-03T23:01:25.412Z",
    "DateUpdated": "2015-02-03T23:01:25.412Z",
    "SourceBundle": {
      "S3Bucket": "my-bucket",
      "S3Key": "sample.war"
    }
  }
}
```

- API-Einheiten finden Sie [CreateApplicationVersion](#) in AWS CLI der Befehlsreferenz.

create-application

Das folgende Codebeispiel zeigt die Verwendung `create-application`.

AWS CLI

Um eine neue Anwendung zu erstellen

Der folgende Befehl erstellt eine neue Anwendung mit dem Namen "MyApp,":

```
aws elasticbeanstalk create-application --application-name MyApp --description "my application"
```

Der `create-application` Befehl konfiguriert nur den Namen und die Beschreibung der Anwendung. Um den Quellcode für die Anwendung hochzuladen, erstellen Sie eine erste Version der Anwendung mit `create-application-version`. `create-application-version` hat auch eine `auto-create-application` Option, mit der Sie die Anwendung und die Anwendungsversion in einem Schritt erstellen können.

Ausgabe:

```
{
  "Application": {
    "ApplicationName": "MyApp",
    "ConfigurationTemplates": [],
```

```
"DateUpdated": "2015-02-12T18:32:21.181Z",  
"Description": "my application",  
"DateCreated": "2015-02-12T18:32:21.181Z"  
}  
}
```

- API-Einheiten finden Sie [CreateApplication](#) in der AWS CLI Befehlsreferenz.

create-configuration-template

Das folgende Codebeispiel zeigt die Verwendung `create-configuration-template`.

AWS CLI

Um eine Konfigurationsvorlage zu erstellen

Der folgende Befehl erstellt eine Konfigurationsvorlage, die `my-app-v1` nach den Einstellungen benannt wird, die auf eine Umgebung mit der ID angewendet wurde `e-rpqsewtp2j`:

```
aws elasticbeanstalk create-configuration-template --application-name my-app --  
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

Ausgabe:

```
{  
  "ApplicationName": "my-app",  
  "TemplateName": "my-app-v1",  
  "DateCreated": "2015-08-12T18:40:39Z",  
  "DateUpdated": "2015-08-12T18:40:39Z",  
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"  
}
```

- API-Einheiten finden Sie [CreateConfigurationTemplate](#) in der AWS CLI Befehlsreferenz.

create-environment

Das folgende Codebeispiel zeigt die Verwendung `create-environment`.

AWS CLI

Um eine neue Umgebung für eine Anwendung zu erstellen

Der folgende Befehl erstellt eine neue Umgebung für Version „v1“ einer Java-Anwendung mit dem Namen „my-app“:

```
aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --cname-prefix my-app --version-label v1 --solution-stack-name "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
```

Ausgabe:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v1",
  "Status": "Launching",
  "EnvironmentId": "e-izqpassy4h",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
  "CNAME": "my-app.elasticbeanstalk.com",
  "Health": "Grey",
  "Tier": {
    "Type": "Standard",
    "Name": "WebServer",
    "Version": " "
  },
  "DateUpdated": "2015-02-03T23:04:54.479Z",
  "DateCreated": "2015-02-03T23:04:54.479Z"
}
```

v1 ist das Label einer Anwendungsversion, mit der zuvor hochgeladen wurde. create-application-version

Um eine JSON Datei zur Definition von Umgebungskonfigurationsoptionen anzugeben

Der folgende create-environment Befehl gibt an, dass eine JSON Datei mit dem Namen verwendet werden myoptions.json soll, um Werte zu überschreiben, die aus dem Lösungstapel oder der Konfigurationsvorlage abgerufen wurden:

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-name sampleapp --option-settings file://myoptions.json
```

myoptions.json ist ein JSON Objekt, das mehrere Einstellungen definiert:

```
[
```

```
{
  "Namespace": "aws:elb:healthcheck",
  "OptionName": "Interval",
  "Value": "15"
},
{
  "Namespace": "aws:elb:healthcheck",
  "OptionName": "Timeout",
  "Value": "8"
},
{
  "Namespace": "aws:elb:healthcheck",
  "OptionName": "HealthyThreshold",
  "Value": "2"
},
{
  "Namespace": "aws:elb:healthcheck",
  "OptionName": "UnhealthyThreshold",
  "Value": "3"
}
]
```

Weitere Informationen finden Sie unter Optionswerte im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie [CreateEnvironment](#) in der AWS CLI Befehlsreferenz.

create-storage-location

Das folgende Codebeispiel zeigt die Verwendung `create-storage-location`.

AWS CLI

Um einen Speicherort zu erstellen

Der folgende Befehl erstellt einen Speicherort in Amazon S3:

```
aws elasticbeanstalk create-storage-location
```

Ausgabe:

```
{
  "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
```

```
}
```

- API-Einheiten finden Sie [CreateStorageLocation](#) in der AWS CLI Befehlsreferenz.

delete-application-version

Das folgende Codebeispiel zeigt die Verwendung `delete-application-version`.

AWS CLI

Um eine Anwendungsversion zu löschen

Der folgende Befehl löscht eine Anwendungsversion, die nach `22a0-stage-150819_182129` einer Anwendung mit dem Namen `my-app` ist:

```
aws elasticbeanstalk delete-application-version --version-label 22a0-stage-150819_182129 --application-name my-app
```

- API-Einheiten finden Sie [DeleteApplicationVersion](#) in der AWS CLI Befehlsreferenz.

delete-application

Das folgende Codebeispiel zeigt die Verwendung `delete-application`.

AWS CLI

So löschen Sie eine Anwendung

Der folgende Befehl löscht eine Anwendung mit dem Namen `my-app`:

```
aws elasticbeanstalk delete-application --application-name my-app
```

- API-Einheiten finden Sie [DeleteApplication](#) in der AWS CLI Befehlsreferenz.

delete-configuration-template

Das folgende Codebeispiel zeigt die Verwendung `delete-configuration-template`.

AWS CLI

Um eine Konfigurationsvorlage zu löschen

Der folgende Befehl löscht eine Konfigurationsvorlage, die `my-template` nach einer Anwendung mit dem Namen `my-app` ist:

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --  
application-name my-app
```

- API-Einheiten finden Sie [DeleteConfigurationTemplate](#) in der AWS CLI Befehlsreferenz.

delete-environment-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-environment-configuration`.

AWS CLI

Um einen Konfigurationsentwurf zu löschen

Der folgende Befehl löscht einen Konfigurationsentwurf für eine Umgebung mit dem Namen `my-env`:

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --  
application-name my-app
```

- API-Einheiten finden Sie [DeleteEnvironmentConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-application-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-application-versions`.

AWS CLI

Um Informationen zu einer Anwendungsversion anzuzeigen

Mit dem folgenden Befehl werden Informationen zu einer Anwendungsversion mit der Bezeichnung `v2` abgerufen:

```
aws elasticbeanstalk describe-application-versions --application-name my-app --  
version-label "v2"
```

Ausgabe:

```
{
```

```
"ApplicationVersions": [  
  {  
    "ApplicationName": "my-app",  
    "VersionLabel": "v2",  
    "Description": "update cover page",  
    "DateCreated": "2015-07-23T01:32:26.079Z",  
    "DateUpdated": "2015-07-23T01:32:26.079Z",  
    "SourceBundle": {  
      "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",  
      "S3Key": "my-app/5026-stage-150723_224258.war"  
    }  
  },  
  {  
    "ApplicationName": "my-app",  
    "VersionLabel": "v1",  
    "Description": "initial version",  
    "DateCreated": "2015-07-23T22:26:10.816Z",  
    "DateUpdated": "2015-07-23T22:26:10.816Z",  
    "SourceBundle": {  
      "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",  
      "S3Key": "my-app/5026-stage-150723_222618.war"  
    }  
  }  
]
```

- API-Einheiten finden Sie [DescribeApplicationVersions](#) in der AWS CLI Befehlsreferenz.

describe-applications

Das folgende Codebeispiel zeigt die Verwendung `describe-applications`.

AWS CLI

Um eine Liste von Anwendungen anzuzeigen

Mit dem folgenden Befehl werden Informationen über Anwendungen in der aktuellen Region abgerufen:

```
aws elasticbeanstalk describe-applications
```

Ausgabe:


```
{
  "Applications": [
    {
      "ApplicationName": "ruby",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-08-13T21:05:44.376Z",
      "Versions": [
        "Sample Application"
      ],
      "DateCreated": "2015-08-13T21:05:44.376Z"
    },
    {
      "ApplicationName": "pythonsample",
      "Description": "Application created from the EB CLI using \"eb init\"",
      "Versions": [
        "Sample Application"
      ],
      "DateCreated": "2015-08-13T19:05:43.637Z",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-08-13T19:05:43.637Z"
    },
    {
      "ApplicationName": "nodejs-example",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-08-06T17:50:02.486Z",
      "Versions": [
        "add elasticache",
        "First Release"
      ],
      "DateCreated": "2015-08-06T17:50:02.486Z"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeApplications](#) in der AWS CLI Befehlsreferenz.

describe-configuration-options

Das folgende Codebeispiel zeigt die Verwendung `describe-configuration-options`.

AWS CLI

Um die Konfigurationsoptionen für eine Umgebung anzuzeigen

Mit dem folgenden Befehl werden Beschreibungen aller verfügbaren Konfigurationsoptionen für eine Umgebung mit dem Namen `my-env` abgerufen:

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --  
application-name my-app
```

Ausgabe (abgekürzt):

```
{  
  "Options": [  
    {  
      "Name": "JVMOptions",  
      "UserDefined": false,  
      "DefaultValue": "Xms=256m,Xmx=256m,XX:MaxPermSize=64m,JVM Options=",  
      "ChangeSeverity": "RestartApplicationServer",  
      "Namespace": "aws:cloudformation:template:parameter",  
      "ValueType": "KeyValueList"  
    },  
    {  
      "Name": "Interval",  
      "UserDefined": false,  
      "DefaultValue": "30",  
      "ChangeSeverity": "NoInterruption",  
      "Namespace": "aws:elb:healthcheck",  
      "MaxValue": 300,  
      "MinValue": 5,  
      "ValueType": "Scalar"  
    },  
    ...  
    {  
      "Name": "LowerThreshold",  
      "UserDefined": false,  
      "DefaultValue": "2000000",  
      "ChangeSeverity": "NoInterruption",  
      "Namespace": "aws:autoscaling:trigger",  
      "MinValue": 0,  
      "ValueType": "Scalar"  
    },  
    {  
      "Name": "ListenerEnabled",  
      "UserDefined": false,  
      "DefaultValue": "true",  
      "ChangeSeverity": "Unknown",
```

```

        "Namespace": "aws:elb:listener",
        "ValueType": "Boolean"
      }
    ]
  }

```

Die verfügbaren Konfigurationsoptionen variieren je nach Plattform und Konfigurationsversion. Weitere Informationen zu Namespaces und unterstützten Optionen finden Sie unter Optionswerte im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeConfigurationOptions](#) AWS CLI

describe-configuration-settings

Das folgende Codebeispiel zeigt die Verwendung `describe-configuration-settings`.

AWS CLI

Um Konfigurationseinstellungen für eine Umgebung anzuzeigen

Mit dem folgenden Befehl werden die Konfigurationseinstellungen für eine Umgebung mit dem Namen `my-env` abgerufen:

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --
application-name my-app
```

Ausgabe (abgekürzt):

```

{
  "ConfigurationSettings": [
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Description": "Environment created from the EB CLI using \"eb create
\",
      "DeploymentStatus": "deployed",
      "DateCreated": "2015-08-13T19:16:25Z",
      "OptionSettings": [
        {
          "OptionName": "Availability Zones",
          "ResourceName": "AWSEBAutoScalingGroup",
          "Namespace": "aws:autoscaling:asg",
          "Value": "Any"
        }
      ]
    }
  ]
}

```

```

    },
    {
      "OptionName": "Cooldown",
      "ResourceName": "AWSEBAutoScalingGroup",
      "Namespace": "aws:autoscaling:asg",
      "Value": "360"
    },
    ...
    {
      "OptionName": "ConnectionDrainingTimeout",
      "ResourceName": "AWSEBLoadBalancer",
      "Namespace": "aws:elb:policies",
      "Value": "20"
    },
    {
      "OptionName": "ConnectionSettingIdleTimeout",
      "ResourceName": "AWSEBLoadBalancer",
      "Namespace": "aws:elb:policies",
      "Value": "60"
    }
  ],
  "DateUpdated": "2015-08-13T23:30:07Z",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
Java 8"
}
]
}

```

Weitere Informationen zu Namespaces und unterstützten Optionen finden Sie unter Optionswerte im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [DescribeConfigurationSettings](#) AWS CLI

describe-environment-health

Das folgende Codebeispiel zeigt die Verwendung `describe-environment-health`.

AWS CLI

Um den Zustand der Umwelt zu überprüfen

Mit dem folgenden Befehl werden allgemeine Gesundheitsinformationen für eine Umgebung mit dem Namen `my-env` abgerufen:

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --  
attribute-names All
```

Ausgabe:

```
{  
  "Status": "Ready",  
  "EnvironmentName": "my-env",  
  "Color": "Green",  
  "ApplicationMetrics": {  
    "Duration": 10,  
    "Latency": {  
      "P99": 0.004,  
      "P75": 0.002,  
      "P90": 0.003,  
      "P95": 0.004,  
      "P85": 0.003,  
      "P10": 0.001,  
      "P999": 0.004,  
      "P50": 0.001  
    },  
    "RequestCount": 45,  
    "StatusCodes": {  
      "Status3xx": 0,  
      "Status2xx": 45,  
      "Status5xx": 0,  
      "Status4xx": 0  
    }  
  },  
  "RefreshedAt": "2015-08-20T21:09:18Z",  
  "HealthStatus": "Ok",  
  "InstancesHealth": {  
    "Info": 0,  
    "Ok": 1,  
    "Unknown": 0,  
    "Severe": 0,  
    "Warning": 0,  
    "Degraded": 0,  
    "NoData": 0,  
    "Pending": 0  
  },  
  "Causes": []  
}
```

```
}
```

Gesundheitsinformationen sind nur für Umgebungen verfügbar, in denen erweiterte Gesundheitsberichte aktiviert sind. Weitere Informationen finden Sie unter [Enhanced Health Reporting and Monitoring](#) im [AWS Elastic Beanstalk Developer Guide](#).

- API-Einheiten finden Sie [DescribeEnvironmentHealth](#) in der AWS CLI Befehlsreferenz.

describe-environment-resources

Das folgende Codebeispiel zeigt die Verwendung `describe-environment-resources`.

AWS CLI

Um Informationen zu den AWS Ressourcen in Ihrer Umgebung anzuzeigen

Mit dem folgenden Befehl werden Informationen über Ressourcen in einer Umgebung mit dem Namen `my-env` abgerufen:

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```

Ausgabe:

```
{
  "EnvironmentResources": {
    "EnvironmentName": "my-env",
    "AutoScalingGroups": [
      {
        "Name": "awseb-e-qu3fyyjyjs-stack-AWSEBAutoScalingGroup-
QSB2Z088SXZT"
      }
    ],
    "Triggers": [],
    "LoadBalancers": [
      {
        "Name": "awseb-e-q-AWSEBLoa-1EEPZ0K98BIF0"
      }
    ],
    "Queues": [],
    "Instances": [
      {
        "Id": "i-0c91c786"
      }
    ]
  }
}
```

```

    ],
    "LaunchConfigurations": [
      {
        "Name": "awseb-e-qu3fyyjyjs-stack-
AWSEBAutoScalingLaunchConfiguration-1UUVQIBC96TQ2"
      }
    ]
  }
}

```

- API-Einheiten finden Sie [DescribeEnvironmentResources](#) in der AWS CLI Befehlsreferenz.

describe-environments

Das folgende Codebeispiel zeigt die Verwendung `describe-environments`.

AWS CLI

Um Informationen über eine Umgebung anzuzeigen

Der folgende Befehl ruft Informationen über eine Umgebung mit dem Namen `my-env` ab:

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

Ausgabe:

```

{
  "Environments": [
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "VersionLabel": "7f58-stage-150812_025409",
      "Status": "Ready",
      "EnvironmentId": "e-rpqsewtp2j",
      "EndpointURL": "awseb-e-w-AWSEBLoa-1483140XB0Q4L-109QXY8121.us-
west-2.elb.amazonaws.com",
      "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
Java 8",
      "CNAME": "my-env.elasticbeanstalk.com",
      "Health": "Green",
      "AbortableOperationInProgress": false,
      "Tier": {
        "Version": " ",

```

```

        "Type": "Standard",
        "Name": "WebServer"
    },
    "DateUpdated": "2015-08-12T18:16:55.019Z",
    "DateCreated": "2015-08-07T20:48:49.599Z"
}
]
}

```

- API-Einheiten finden Sie [DescribeEnvironments](#) in der AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um Ereignisse für eine Umgebung anzuzeigen

Der folgende Befehl ruft Ereignisse für eine Umgebung mit dem Namen `my-env` ab:

```
aws elasticbeanstalk describe-events --environment-name my-env
```

Ausgabe (abgekürzt):

```

{
  "Events": [
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Message": "Environment health has transitioned from Info to Ok.",
      "EventDate": "2015-08-20T07:06:53.535Z",
      "Severity": "INFO"
    },
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Severity": "INFO",
      "RequestId": "b7f3960b-4709-11e5-ba1e-07e16200da41",
      "Message": "Environment update completed successfully.",
      "EventDate": "2015-08-20T07:06:02.049Z"
    },
    ...
  ]
}

```



```

    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Severity": "INFO",
      "RequestId": "ca8dfbf6-41ef-11e5-988b-651aa638f46b",
      "Message": "Using elasticbeanstalk-us-west-2-012445113685 as Amazon S3
storage bucket for environment data.",
      "EventDate": "2015-08-13T19:16:27.561Z"
    },
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Severity": "INFO",
      "RequestId": "cdfba8f6-41ef-11e5-988b-65638f41aa6b",
      "Message": "createEnvironment is starting.",
      "EventDate": "2015-08-13T19:16:26.581Z"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

describe-instances-health

Das folgende Codebeispiel zeigt die Verwendung `describe-instances-health`.

AWS CLI

Um den Zustand der Umwelt zu überprüfen

Mit dem folgenden Befehl werden Integritätsinformationen für Instanzen in einer Umgebung mit dem Namen `my-env` abgerufen:

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --
attribute-names ALL
```

Ausgabe:

```

{
  "InstanceHealthList": [
    {
      "InstanceId": "i-08691cc7",

```

```
    "ApplicationMetrics": {
      "Duration": 10,
      "Latency": {
        "P99": 0.006,
        "P75": 0.002,
        "P90": 0.004,
        "P95": 0.005,
        "P85": 0.003,
        "P10": 0.0,
        "P999": 0.006,
        "P50": 0.001
      },
      "RequestCount": 48,
      "StatusCodes": {
        "Status3xx": 0,
        "Status2xx": 47,
        "Status5xx": 0,
        "Status4xx": 1
      }
    },
    "System": {
      "LoadAverage": [
        0.0,
        0.02,
        0.05
      ],
      "CPUUtilization": {
        "SoftIRQ": 0.1,
        "IOWait": 0.2,
        "System": 0.3,
        "Idle": 97.8,
        "User": 1.5,
        "IRQ": 0.0,
        "Nice": 0.1
      }
    },
    "Color": "Green",
    "HealthStatus": "Ok",
    "LaunchedAt": "2015-08-13T19:17:09Z",
    "Causes": []
  }
],
"RefreshedAt": "2015-08-20T21:09:08Z"
```

```
}
```

Gesundheitsinformationen sind nur für Umgebungen verfügbar, in denen erweiterte Gesundheitsberichte aktiviert sind. Weitere Informationen finden Sie unter [Enhanced Health Reporting and Monitoring](#) im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie [DescribeInstancesHealth](#) in der AWS CLI Befehlsreferenz.

list-available-solution-stacks

Das folgende Codebeispiel zeigt die Verwendung `list-available-solution-stacks`.

AWS CLI

Um Lösungstapel anzuzeigen

Der folgende Befehl listet Lösungstapel für alle derzeit verfügbaren Plattformkonfigurationen und alle, die Sie in der Vergangenheit verwendet haben, auf:

```
aws elasticbeanstalk list-available-solution-stacks
```

Ausgabe (abgekürzt):

```
{
  "SolutionStacks": [
    "64bit Amazon Linux 2015.03 v2.0.0 running Node.js",
    "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.6",
    "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.5",
    "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.4",
    "64bit Amazon Linux 2015.03 v2.0.0 running Python 3.4",
    "64bit Amazon Linux 2015.03 v2.0.0 running Python 2.7",
    "64bit Amazon Linux 2015.03 v2.0.0 running Python",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Puma)",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Puma)",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Passenger Standalone)",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Puma)",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Passenger Standalone)",
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 1.9.3",
    "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
    "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 7",
    "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 6",
  ]
}
```

```

    "64bit Windows Server Core 2012 R2 running IIS 8.5",
    "64bit Windows Server 2012 R2 running IIS 8.5",
    "64bit Windows Server 2012 running IIS 8",
    "64bit Windows Server 2008 R2 running IIS 7.5",
    "64bit Amazon Linux 2015.03 v2.0.0 running Docker 1.6.2",
    "64bit Amazon Linux 2015.03 v2.0.0 running Multi-container Docker 1.6.2
(Generic)",
    "64bit Debian jessie v2.0.0 running GlassFish 4.1 Java 8 (Preconfigured -
Docker)",
    "64bit Debian jessie v2.0.0 running GlassFish 4.0 Java 7 (Preconfigured -
Docker)",
    "64bit Debian jessie v2.0.0 running Go 1.4 (Preconfigured - Docker)",
    "64bit Debian jessie v2.0.0 running Go 1.3 (Preconfigured - Docker)",
    "64bit Debian jessie v2.0.0 running Python 3.4 (Preconfigured - Docker)",
  ],
  "SolutionStackDetails": [
    {
      "PermittedFileTypes": [
        "zip"
      ],
      "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Node.js"
    },
    ...
  ]
}

```

- API-Einheiten finden Sie [ListAvailableSolutionStacks](#) in der AWS CLI Befehlsreferenz.

rebuild-environment

Das folgende Codebeispiel zeigt die Verwendung `rebuild-environment`.

AWS CLI

Um eine Umgebung neu aufzubauen

Mit dem folgenden Befehl werden die Ressourcen in einer Umgebung mit dem Namen beendet und neu erstellt: `my-env`

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- API-Einheiten finden Sie [RebuildEnvironment](#) in der AWS CLI Befehlsreferenz.

request-environment-info

Das folgende Codebeispiel zeigt die Verwendung `request-environment-info`.

AWS CLI

Um detaillierte Protokolle anzufordern

Der folgende Befehl fordert Protokolle aus einer Umgebung mit dem Namen `my-env` an:

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-type tail
```

Nachdem Sie Protokolle angefordert haben, rufen Sie deren Speicherort mit `retrieve-environment-info`.

- API-Einheiten finden Sie [RequestEnvironmentInfo](#) in der AWS CLI Befehlsreferenz.

restart-app-server

Das folgende Codebeispiel zeigt die Verwendung `restart-app-server`.

AWS CLI

Um Anwendungsserver neu zu starten

Mit dem folgenden Befehl werden die Anwendungsserver auf allen Instanzen in einer Umgebung mit dem Namen `my-env` neu gestartet:

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- API-Einheiten finden Sie [RestartAppServer](#) in der AWS CLI Befehlsreferenz.

retrieve-environment-info

Das folgende Codebeispiel zeigt die Verwendung `retrieve-environment-info`.

AWS CLI

Um detaillierte Protokolle abzurufen

Der folgende Befehl ruft einen Link zu Protokollen aus einer Umgebung mit dem Namen ab: my-env

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-type tail
```

Ausgabe:

```
{
  "EnvironmentInfo": [
    {
      "SampleTimestamp": "2015-08-20T22:23:17.703Z",
      "Message": "https://elasticbeanstalk-us-west-2-0123456789012.s3.amazonaws.com/resources/environments/logs/tail/e-fyqyju3yjs/i-09c1c867/TailLogs-1440109397703.out?AWSAccessKeyId=AKGPT4J56IAJ2EUBL5CQ&Expires=1440195891&Signature=n%2BEal0V6A2HI0x4Rcfb7LT16bBM%3D",
      "InfoType": "tail",
      "Ec2InstanceId": "i-09c1c867"
    }
  ]
}
```

Zeigen Sie den Link in einem Browser an. Vor dem Abruf müssen Protokolle mit request-environment-info angefordert werden.

- API-Einheiten finden Sie [RetrieveEnvironmentInfo](#) in der AWS CLI Befehlsreferenz.

swap-environment-cnames

Das folgende Codebeispiel zeigt die Verwendung `swap-environment-cnames`.

AWS CLI

Um die Umgebung auszutauschen CNAMEs

Der folgende Befehl tauscht die zugewiesenen Subdomänen zweier Umgebungen aus:

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue --destination-environment-name my-env-green
```

- API-Einheiten finden Sie [SwapEnvironmentCnames](#) in der AWS CLI Befehlsreferenz.

terminate-environment

Das folgende Codebeispiel zeigt die Verwendung `terminate-environment`.

AWS CLI

Um eine Umgebung zu beenden

Der folgende Befehl beendet eine Elastic Beanstalk Beanstalk-Umgebung mit dem Namen: `my-env`

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

Ausgabe:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "Status": "Terminating",
  "EnvironmentId": "e-fh2eravpns",
  "EndpointURL": "awseb-e-f-AWSEBLoa-1I9XUMP4-8492WNUP202574.us-west-2.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "AbortableOperationInProgress": false,
  "Tier": {
    "Version": " ",
    "Type": "Standard",
    "Name": "WebServer"
  },
  "DateUpdated": "2015-08-12T19:05:54.744Z",
  "DateCreated": "2015-08-12T18:52:53.622Z"
}
```

- API-Einheiten finden Sie [TerminateEnvironment](#) in AWS CLI der Befehlsreferenz.

update-application-version

Das folgende Codebeispiel zeigt die Verwendung `update-application-version`.

AWS CLI

Um die Beschreibung einer Anwendungsversion zu ändern

Der folgende Befehl aktualisiert die Beschreibung einer Anwendungsversion mit dem Namen `22a0-stage-150819_185942`:

```
aws elasticbeanstalk update-application-version --version-label 22a0-stage-150819_185942 --application-name my-app --description "new description"
```

Ausgabe:

```
{
  "ApplicationVersion": {
    "ApplicationName": "my-app",
    "VersionLabel": "22a0-stage-150819_185942",
    "Description": "new description",
    "DateCreated": "2015-08-19T18:59:17.646Z",
    "DateUpdated": "2015-08-20T22:53:28.871Z",
    "SourceBundle": {
      "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012",
      "S3Key": "my-app/22a0-stage-150819_185942.war"
    }
  }
}
```

- API-Einheiten finden Sie [UpdateApplicationVersion](#) in der AWS CLI Befehlsreferenz.

update-application

Das folgende Codebeispiel zeigt die Verwendung `update-application`.

AWS CLI

Um die Beschreibung einer Anwendung zu ändern

Der folgende Befehl aktualisiert die Beschreibung einer Anwendung mit dem Namen `my-app`:

```
aws elasticbeanstalk update-application --application-name my-app --description "my Elastic Beanstalk application"
```

Ausgabe:


```
{
  "Application": {
    "ApplicationName": "my-app",
    "Description": "my Elastic Beanstalk application",
    "Versions": [
      "2fba-stage-150819_234450",
      "bf07-stage-150820_214945",
      "93f8",
      "fd7c-stage-150820_000431",
      "22a0-stage-150819_185942"
    ],
    "DateCreated": "2015-08-13T19:15:50.449Z",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-08-20T22:34:56.195Z"
  }
}
```

- API-Einheiten finden Sie [UpdateApplication](#) in der AWS CLI Befehlsreferenz.

update-configuration-template

Das folgende Codebeispiel zeigt die Verwendung `update-configuration-template`.

AWS CLI

Um eine Konfigurationsvorlage zu aktualisieren

Mit dem folgenden Befehl wird die konfigurierte Konfiguration für CloudWatch benutzerdefinierte Integritätsmetriken `ConfigDocument` aus einer gespeicherten Konfigurationsvorlage mit dem Namen `my-template` entfernt:

```
aws elasticbeanstalk update-configuration-template --template-name my-template --application-name my-app --options-to-remove Namespace=aws:elasticbeanstalk:healthreporting:system,OptionName=ConfigDocument
```

Ausgabe:

```
{
  "ApplicationName": "my-app",
  "TemplateName": "my-template",
  "DateCreated": "2015-08-20T22:39:31Z",
  "DateUpdated": "2015-08-20T22:43:11Z",
```

```
"SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
}
```

Weitere Informationen zu Namespaces und unterstützten Optionen finden Sie unter Optionswerte im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [UpdateConfigurationTemplate](#) AWS CLI

update-environment

Das folgende Codebeispiel zeigt die Verwendung `update-environment`.

AWS CLI

Um eine Umgebung auf eine neue Version zu aktualisieren

Mit dem folgenden Befehl wird eine Umgebung mit dem Namen „my-env“ auf Version „v2“ der Anwendung aktualisiert, zu der sie gehört:

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

Dieser Befehl setzt voraus, dass die Umgebung „my-env“ bereits existiert und zu einer Anwendung gehört, die über eine gültige Anwendungsversion mit der Bezeichnung „v2“ verfügt.

Ausgabe:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v2",
  "Status": "Updating",
  "EnvironmentId": "e-szqipays4h",
  "EndpointURL": "awseb-e-i-AWSEBLoa-1RD LX6TC9VUA0-0123456789.us-
west-2.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux running Tomcat 7",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "Tier": {
    "Version": " ",
    "Type": "Standard",
    "Name": "WebServer"
  },
}
```

```
"DateUpdated": "2015-02-03T23:12:29.119Z",  
"DateCreated": "2015-02-03T23:04:54.453Z"  
}
```

Um eine Umgebungsvariable festzulegen

Der folgende Befehl setzt den Wert der Variablen "PARAM1" in der Umgebung „my-env“ auf "ParamValue,“:

```
aws elasticbeanstalk update-environment --environment-name my-env --option-  
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para
```

Der option-settings Parameter benötigt zusätzlich zum Namen und Wert der Variablen einen Namespace. Elastic Beanstalk unterstützt neben Umgebungsvariablen auch mehrere Namespaces für Optionen.

Um Optionseinstellungen aus einer Datei zu konfigurieren

Der folgende Befehl konfiguriert mehrere Optionen im `aws:elb:loadbalancer` Namespace aus einer Datei:

```
aws elasticbeanstalk update-environment --environment-name my-env --option-  
settings file://options.json
```

`options.json` ist ein JSON Objekt, das mehrere Einstellungen definiert:

```
[  
  {  
    "Namespace": "aws:elb:healthcheck",  
    "OptionName": "Interval",  
    "Value": "15"  
  },  
  {  
    "Namespace": "aws:elb:healthcheck",  
    "OptionName": "Timeout",  
    "Value": "8"  
  },  
  {  
    "Namespace": "aws:elb:healthcheck",  
    "OptionName": "HealthyThreshold",  
    "Value": "2"  
  },  
]
```

```
{
  "Namespace": "aws:elb:healthcheck",
  "OptionName": "UnhealthyThreshold",
  "Value": "3"
}
]
```

Ausgabe:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "7f58-stage-150812_025409",
  "Status": "Updating",
  "EnvironmentId": "e-wtp2rqpsej",
  "EndpointURL": "awseb-e-w-AWSEBLoa-14XB83101Q4L-104QXY80921.sa-
east-1.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
8",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "AbortableOperationInProgress": true,
  "Tier": {
    "Version": " ",
    "Type": "Standard",
    "Name": "WebServer"
  },
  "DateUpdated": "2015-08-12T18:15:23.804Z",
  "DateCreated": "2015-08-07T20:48:49.599Z"
}
```

Weitere Informationen zu Namespaces und unterstützten Optionen finden Sie unter Optionswerte im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [UpdateEnvironment](#) AWS CLI

validate-configuration-settings

Das folgende Codebeispiel zeigt die Verwendung `validate-configuration-settings`.

AWS CLI

Um die Konfigurationseinstellungen zu überprüfen

Der folgende Befehl validiert ein Konfigurationsdokument für CloudWatch benutzerdefinierte Metriken:

```
aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings file://options.json
```

`options.json` ist ein JSON Dokument, das eine oder mehrere zu validierende Konfigurationseinstellungen enthält:

```
[
  {
    "Namespace": "aws:elasticbeanstalk:healthreporting:system",
    "OptionName": "ConfigDocument",
    "Value": "{\"CloudWatchMetrics\": {\"Environment\":
  {\"ApplicationLatencyP99.9\": null, \"InstancesSevere\": 60,
  \"ApplicationLatencyP90\": 60, \"ApplicationLatencyP99\": null,
  \"ApplicationLatencyP95\": 60, \"InstancesUnknown\": 60, \"ApplicationLatencyP85\":
  60, \"InstancesInfo\": null, \"ApplicationRequests2xx\": null, \"InstancesDegraded
  \": null, \"InstancesWarning\": 60, \"ApplicationLatencyP50\": 60,
  \"ApplicationRequestsTotal\": null, \"InstancesNoData\": null, \"InstancesPending
  \": 60, \"ApplicationLatencyP10\": null, \"ApplicationRequests5xx\": null,
  \"ApplicationLatencyP75\": null, \"InstancesOk\": 60, \"ApplicationRequests3xx\":
  null, \"ApplicationRequests4xx\": null}, \"Instance\": {\"ApplicationLatencyP99.9\":
  null, \"ApplicationLatencyP90\": 60, \"ApplicationLatencyP99\": null,
  \"ApplicationLatencyP95\": null, \"ApplicationLatencyP85\": null, \"CPUUser\": 60,
  \"ApplicationRequests2xx\": null, \"CPUIdle\": null, \"ApplicationLatencyP50\":
  null, \"ApplicationRequestsTotal\": 60, \"RootFilesystemUtil\": null,
  \"LoadAverage1min\": null, \"CPUIrq\": null, \"CPUNice\": 60, \"CPUiowait\": 60,
  \"ApplicationLatencyP10\": null, \"LoadAverage5min\": null, \"ApplicationRequests5xx
  \": null, \"ApplicationLatencyP75\": 60, \"CPUSystem\": 60, \"ApplicationRequests3xx\":
  60, \"ApplicationRequests4xx\": null, \"InstanceHealth\": null, \"CPUSoftirq\": 60}},
  \"Version\": 1}"
  }
]
```

Wenn die von Ihnen angegebenen Optionen für die angegebene Umgebung gültig sind, gibt Elastic Beanstalk ein leeres Messages-Array zurück:

```
{
  "Messages": []
}
```

Wenn die Validierung fehlschlägt, enthält die Antwort Informationen über den Fehler:

```
{
  "Messages": [
    {
      "OptionName": "ConfigDocumet",
      "Message": "Invalid option specification (Namespace:
'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
Unknown configuration setting.",
      "Namespace": "aws:elasticbeanstalk:healthreporting:system",
      "Severity": "error"
    }
  ]
}
```

Weitere Informationen zu Namespaces und unterstützten Optionen finden Sie unter Optionswerte im AWS Elastic Beanstalk Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [ValidateConfigurationSettings](#) AWS CLI

Elastic Load Balancing — Version 1, Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Elastic Load Balancing — Version 1 Aktionen ausführen und gängige Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags

Das folgende Codebeispiel zeigt die Verwendung `add-tags`.

AWS CLI

Um einem Load Balancer ein Tag hinzuzufügen

In diesem Beispiel werden dem angegebenen Load Balancer Tags hinzugefügt.

Befehl:

```
aws elb add-tags --load-balancer-name my-load-balancer --  
tags "Key=project,Value=Lima" "Key=department,Value=digital-media"
```

- API-Einheiten finden Sie [AddTags](#) in der AWS CLI Befehlsreferenz.

apply-security-groups-to-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `apply-security-groups-to-load-balancer`.

AWS CLI

Um eine Sicherheitsgruppe einem Load Balancer in einem zuzuordnen VPC

In diesem Beispiel wird eine Sicherheitsgruppe dem angegebenen Load Balancer in einem zugeordnet. VPC

Befehl:

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer  
--security-groups sg-fc448899
```

Ausgabe:

```
{  
  "SecurityGroups": [  
    "sg-fc448899"  
  ]  
}
```

- API-Einheiten finden Sie [ApplySecurityGroupsToLoadBalancer](#) in der AWS CLI Befehlsreferenz.

attach-load-balancer-to-subnets

Das folgende Codebeispiel zeigt die Verwendung `attach-load-balancer-to-subnets`.

AWS CLI

Um Subnetze an einen Load Balancer anzuhängen

In diesem Beispiel wird das angegebene Subnetz zu den konfigurierten Subnetzen für den angegebenen Load Balancer hinzugefügt.

Befehl:

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --  
subnets subnet-0ecac448
```

Ausgabe:

```
{  
  "Subnets": [  
    "subnet-15aaab61",  
    "subnet-0ecac448"  
  ]  
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [AttachLoadBalancerToSubnets](#).AWS CLI

configure-health-check

Das folgende Codebeispiel zeigt die Verwendung `configure-health-check`.

AWS CLI

Um die Einstellungen für die Integritätsprüfung für Ihre Back-End-Instanzen EC2 anzugeben

In diesem Beispiel werden die Einstellungen für die Integritätsprüfung angegeben, die zur Bewertung des Zustands Ihrer EC2 Backend-Instanzen verwendet werden.

Befehl:


```
aws elb configure-health-check --load-balancer-name my-load-balancer --health-check Target=HTTP:80/png,Interval=30,UnhealthyThreshold=2,HealthyThreshold=2,Timeout=3
```

Ausgabe:

```
{
  "HealthCheck": {
    "HealthyThreshold": 2,
    "Interval": 30,
    "Target": "HTTP:80/png",
    "Timeout": 3,
    "UnhealthyThreshold": 2
  }
}
```

- API-Einheiten finden Sie [ConfigureHealthCheck](#) in der AWS CLI Befehlsreferenz.

create-app-cookie-stickiness-policy

Das folgende Codebeispiel zeigt die Verwendung `create-app-cookie-stickiness-policy`.

AWS CLI

Um eine Stickiness-Richtlinie für Ihren HTTPS Load Balancer zu generieren

In diesem Beispiel wird eine Stickiness-Richtlinie generiert, die sich an die Lebensdauer des von der Anwendung generierten Cookies für Sticky-Sitzungen hält.

Befehl:

```
aws elb create-app-cookie-stickiness-policy --load-balancer-name my-load-balancer --policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- API-Einheiten finden Sie in der Befehlsreferenz [CreateAppCookieStickinessPolicy](#).AWS CLI

create-lb-cookie-stickiness-policy

Das folgende Codebeispiel zeigt die Verwendung `create-lb-cookie-stickiness-policy`.

AWS CLI

Um eine auf der Dauer basierende Stickiness-Richtlinie für Ihren Load Balancer zu generieren
HTTPS

In diesem Beispiel wird eine Stickiness-Richtlinie generiert, bei der die Lebensdauer von
Sperrsitzen durch den angegebenen Ablaufzeitraum gesteuert wird.

Befehl:

```
aws elb create-lb-cookie-stickiness-policy --load-balancer-name my-load-balancer --  
policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- API-Einheiten finden Sie [CreateLbCookieStickinessPolicy](#) in der AWS CLI Befehlsreferenz.

create-load-balancer-listeners

Das folgende Codebeispiel zeigt die Verwendung `create-load-balancer-listeners`.

AWS CLI

Um HTTP Listener für einen Load Balancer zu erstellen

In diesem Beispiel wird mithilfe des Protokolls ein Listener für Ihren Load Balancer an Port 80
erstellt. HTTP

Befehl:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

Um HTTPS Listener für einen Load Balancer zu erstellen

In diesem Beispiel wird mithilfe des Protokolls ein Listener für Ihren Load Balancer an Port 443
erstellt. HTTPS

Befehl:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- API-Einheiten finden Sie unter [CreateLoadBalancerListeners AWS CLI](#) Befehlsreferenz.

create-load-balancer-policy

Das folgende Codebeispiel zeigt die Verwendung `create-load-balancer-policy`.

AWS CLI

Um eine Richtlinie zu erstellen, die das Proxy-Protokoll auf einem Load Balancer aktiviert

In diesem Beispiel wird eine Richtlinie erstellt, die das Proxy-Protokoll auf dem angegebenen Load Balancer aktiviert.

Befehl:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-ProxyProtocol-policy --policy-type-name ProxyProtocolPolicyType --policy-attributes AttributeName=ProxyProtocol,AttributeValue=true
```

Um eine SSL Verhandlungsrichtlinie unter Verwendung der empfohlenen Sicherheitsrichtlinie zu erstellen

In diesem Beispiel wird mithilfe der empfohlenen Sicherheitsrichtlinie eine SSL Verhandlungsrichtlinie für den angegebenen Load HTTPS Balancer erstellt.

Befehl:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-SSLNegotiation-policy --policy-type-name SSLNegotiationPolicyType --policy-attributes AttributeName=Reference-Security-Policy,AttributeValue=ELBSecurityPolicy-2015-03
```

Um eine SSL Verhandlungsrichtlinie mithilfe einer benutzerdefinierten Sicherheitsrichtlinie zu erstellen

In diesem Beispiel wird mithilfe einer benutzerdefinierten Sicherheitsrichtlinie eine SSL Verhandlungsrichtlinie für Ihren Load HTTPS Balancer erstellt, indem die Protokolle und Chiffren aktiviert werden.

Befehl:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-SSLNegotiation-policy --policy-type-name SSLNegotiationPolicyType --policy-attributes AttributeName=Protocol-SSLv3,AttributeValue=true AttributeName=Protocol-
```

```
TLSv1.1,AttributeValue=true AttributeName=DHE-RSA-AES256-  
SHA256,AttributeValue=true AttributeName=Server-Defined-Cipher-  
Order,AttributeValue=true
```

Um eine Richtlinie für öffentliche Schlüssel zu erstellen

In diesem Beispiel wird eine Richtlinie für öffentliche Schlüssel erstellt.

Befehl:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --  
policy-name my-PublicKey-policy --policy-type-name PublicKeyPolicyType --policy-  
attributes AttributeName=PublicKey,AttributeValue=MIIBIjANBgkqhkiG9w0BAQEFAAAOCAQ8AMIIBCgKCAQ  
+dS74kj//c6x7R0tusUaeQCTgIUkayttRDWchuqo1pHC1u  
+n5xxXnBBE2ejbb2WRsKIQ5rXEeixsjFpFsojpsQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe  
+o4haAWvDBALJU0pkSI1FecPHcs2hwx14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCE0jnYRg7RPvybaGfQ  
+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPubcUkkXDj45vC2s3Z8E  
+Lk7a3uZhvsQHLZnrFuWjBWGWvZ/MhZYgEXAMPLE
```

Um eine Authentifizierungsrichtlinie für Backend-Server zu erstellen

In diesem Beispiel wird eine Backend-Server-Authentifizierungsrichtlinie erstellt, die die Authentifizierung auf Ihrer Back-End-Instance mithilfe einer Public-Key-Richtlinie ermöglicht.

Befehl:

```
aws elb create-load-balancer-policy --load-balancer-name my-  
load-balancer --policy-name my-authentication-policy --policy-  
type-name BackendServerAuthenticationPolicyType --policy-  
attributes AttributeName=PublicKeyPolicyName,AttributeValue=my-PublicKey-policy
```

- API-Einheiten finden Sie unter [CreateLoadBalancerPolicy AWS CLI](#) Befehlsreferenz.

create-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `create-load-balancer`.

AWS CLI

Um einen HTTP Load Balancer zu erstellen

In diesem Beispiel wird ein Load Balancer mit einem HTTP Listener in einem VPC erstellt.

Befehl:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"  
--subnets subnet-15aaab61 --security-groups sg-a61988c3
```

Ausgabe:

```
{  
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"  
}
```

In diesem Beispiel wird ein Load Balancer mit einem HTTP Listener in -Classic erstellt. EC2

Befehl:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"  
--availability-zones us-west-2a us-west-2b
```

Ausgabe:

```
{  
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

Um einen Load Balancer zu erstellen HTTPS

In diesem Beispiel wird ein Load Balancer mit einem HTTPS Listener in einem erstellt. VPC

Befehl:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTPS,InstancePort=443,certificate/my-server-cert" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

Ausgabe:

```
{  
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"  
}
```

In diesem Beispiel wird ein Load Balancer mit einem HTTPS Listener in -Classic erstellt. EC2

Befehl:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTPS,InstancePort=443,certificate/my-server-cert" --availability-zones us-west-2a us-west-2b
```

Ausgabe:

```
{  
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

Um einen internen Load Balancer zu erstellen

In diesem Beispiel wird ein interner Load Balancer mit einem HTTP Listener in einem VPC erstellt.

Befehl:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"  
--scheme internal --subnets subnet-a85db0df --security-groups sg-a61988c3
```

Ausgabe:

```
{  
  "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

- API-Einheiten finden Sie [CreateLoadBalancer](#) in der AWS CLI Befehlsreferenz.

delete-load-balancer-listeners

Das folgende Codebeispiel zeigt die Verwendung `delete-load-balancer-listeners`.

AWS CLI

Um einen Listener aus Ihrem Load Balancer zu löschen

In diesem Beispiel wird der Listener für den angegebenen Port aus dem angegebenen Load Balancer gelöscht.

Befehl:

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-balancer-ports 80
```

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteLoadBalancerListeners](#).AWS CLI

delete-load-balancer-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-load-balancer-policy`.

AWS CLI

Um eine Richtlinie aus Ihrem Load Balancer zu löschen

In diesem Beispiel wird die angegebene Richtlinie aus dem angegebenen Load Balancer gelöscht. Die Richtlinie darf auf keinem Listener aktiviert sein.

Befehl:

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- API-Einheiten finden Sie [DeleteLoadBalancerPolicy](#) in der AWS CLI Befehlsreferenz.

delete-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `delete-load-balancer`.

AWS CLI

Um einen Load Balancer zu löschen

In diesem Beispiel wird der angegebene Load Balancer gelöscht.

Befehl:

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- API-Einheiten finden Sie [DeleteLoadBalancer](#) in der AWS CLI Befehlsreferenz.

deregister-instances-from-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `deregister-instances-from-load-balancer`.

AWS CLI

Um Instances von einem Load Balancer abzumelden

In diesem Beispiel wird die angegebene Instance vom angegebenen Load Balancer deregistriert.

Befehl:

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-balancer --instances i-d6f6fae3
```

Ausgabe:

```
{
  "Instances": [
    {
      "InstanceId": "i-207d9717"
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

- API-Einheiten finden Sie [DeregisterInstancesFromLoadBalancer](#) in AWS CLI der Befehlsreferenz.

describe-account-limits

Das folgende Codebeispiel zeigt die Verwendung `describe-account-limits`.

AWS CLI

Um Ihre Classic Load Balancer Balancer-Limits zu beschreiben

Im folgenden `describe-account-limits` Beispiel werden Details zu den Classic Load Balancer Balancer-Limits für Ihr AWS Konto angezeigt.


```
aws elb describe-account-limits
```

Ausgabe:

```
{
  "Limits": [
    {
      "Name": "classic-load-balancers",
      "Max": "20"
    },
    {
      "Name": "classic-listeners",
      "Max": "100"
    },
    {
      "Name": "classic-registered-instances",
      "Max": "1000"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeAccountLimits](#) in der AWS CLI Befehlsreferenz.

describe-instance-health

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-health`.

AWS CLI

Um den Zustand der Instances für einen Load Balancer zu beschreiben

In diesem Beispiel wird der Zustand der Instances für den angegebenen Load Balancer beschrieben.

Befehl:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

Ausgabe:

```
{
  "InstanceStates": [
```

```
{
  "InstanceId": "i-207d9717",
  "ReasonCode": "N/A",
  "State": "InService",
  "Description": "N/A"
},
{
  "InstanceId": "i-afefb49b",
  "ReasonCode": "N/A",
  "State": "InService",
  "Description": "N/A"
}
]
```

Um den Zustand einer Instance für einen Load Balancer zu beschreiben

In diesem Beispiel wird der Zustand der angegebenen Instance für den angegebenen Load Balancer beschrieben.

Befehl:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --
instances i-7299c809
```

Im Folgenden finden Sie eine Beispielantwort für eine Instance, die sich gerade registriert.

Ausgabe:

```
{
  "InstanceStates": [
    {
      "InstanceId": "i-7299c809",
      "ReasonCode": "ELB",
      "State": "OutOfService",
      "Description": "Instance registration is still in progress."
    }
  ]
}
```

Im Folgenden finden Sie eine Beispielantwort für eine fehlerhafte Instance.

Ausgabe:

```
{
  "InstanceStates": [
    {
      "InstanceId": "i-7299c809",
      "ReasonCode": "Instance",
      "State": "OutOfService",
      "Description": "Instance has failed at least the UnhealthyThreshold number
of health checks consecutively."
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeInstanceHealth](#) in der AWS CLI Befehlsreferenz.

describe-load-balancer-attributes

Das folgende Codebeispiel zeigt die Verwendung von `describe-load-balancer-attributes`.

AWS CLI

Um die Attribute eines Load Balancers zu beschreiben

In diesem Beispiel werden die Attribute des angegebenen Load Balancers beschrieben.

Befehl:

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

Ausgabe:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": false,
      "Timeout": 300
    },
    "CrossZoneLoadBalancing": {
      "Enabled": true
    },
    "ConnectionSettings": {
      "IdleTimeout": 30
    },
  },
}
```

```
    "AccessLog": {
      "Enabled": false
    }
  }
}
```

- API-Einheiten finden Sie [DescribeLoadBalancerAttributes](#) in der AWS CLI Befehlsreferenz.

describe-load-balancer-policies

Das folgende Codebeispiel zeigt die Verwendung `describe-load-balancer-policies`.

AWS CLI

Um alle Richtlinien zu beschreiben, die mit einem Load Balancer verknüpft sind

In diesem Beispiel werden alle Richtlinien beschrieben, die dem angegebenen Load Balancer zugeordnet sind.

Befehl:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

Ausgabe:

```
{
  "PolicyDescriptions": [
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "ProxyProtocol",
          "AttributeValue": "true"
        }
      ],
      "PolicyName": "my-ProxyProtocol-policy",
      "PolicyTypeName": "ProxyProtocolPolicyType"
    },
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "CookieName",
          "AttributeValue": "my-app-cookie"
        }
      ]
    }
  ]
}
```

```

    ],
    "PolicyName": "my-app-cookie-policy",
    "PolicyTypeName": "AppCookieStickinessPolicyType"
  },
  {
    "PolicyAttributeDescriptions": [
      {
        "AttributeName": "CookieExpirationPeriod",
        "AttributeValue": "60"
      }
    ],
    "PolicyName": "my-duration-cookie-policy",
    "PolicyTypeName": "LBCookieStickinessPolicyType"
  },
  .
  .
  .
]
}

```

Um eine bestimmte Richtlinie zu beschreiben, die einem Load Balancer zugeordnet ist

Dieses Beispiel beschreibt die angegebene Richtlinie, die dem angegebenen Load Balancer zugeordnet ist.

Befehl:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --
policy-name my-authentication-policy
```

Ausgabe:

```

{
  "PolicyDescriptions": [
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "PublicKeyPolicyName",
          "AttributeValue": "my-PublicKey-policy"
        }
      ],
      "PolicyName": "my-authentication-policy",
      "PolicyTypeName": "BackendServerAuthenticationPolicyType"
    }
  ]
}

```

```

    }
  ]
}

```

- API-Einheiten finden Sie [DescribeLoadBalancerPolicies](#) in der AWS CLI Befehlsreferenz.

describe-load-balancer-policy-types

Das folgende Codebeispiel zeigt die Verwendung `describe-load-balancer-policy-types`.

AWS CLI

Um die von Elastic Load Balancing definierten Load Balancer-Richtlinientypen zu beschreiben

In diesem Beispiel werden die Load Balancer-Richtlinientypen beschrieben, mit denen Sie Richtlinienkonfigurationen für Ihren Load Balancer erstellen können.

Befehl:

```
aws elb describe-load-balancer-policy-types
```

Ausgabe:

```

{
  "PolicyTypeDescriptions": [
    {
      "PolicyAttributeTypeDescriptions": [
        {
          "Cardinality": "ONE",
          "AttributeName": "ProxyProtocol",
          "AttributeType": "Boolean"
        }
      ],
      "PolicyTypeName": "ProxyProtocolPolicyType",
      "Description": "Policy that controls whether to include the IP address and port of the originating request for TCP messages. This policy operates on TCP/SSL listeners only"
    },
    {
      "PolicyAttributeTypeDescriptions": [
        {
          "Cardinality": "ONE",
          "AttributeName": "PublicKey",

```

```

        "AttributeType": "String"
    }
],
    "PolicyTypeName": "PublicKeyPolicyType",
    "Description": "Policy containing a list of public keys to
accept when authenticating the back-end server(s). This policy cannot be
applied directly to back-end servers or listeners but must be part of a
BackendServerAuthenticationPolicyType."
},
{
    "PolicyAttributeTypeDescriptions": [
        {
            "Cardinality": "ONE",
            "AttributeName": "CookieName",
            "AttributeType": "String"
        }
    ],
    "PolicyTypeName": "AppCookieStickinessPolicyType",
    "Description": "Stickiness policy with session lifetimes controlled by the
lifetime of the application-generated cookie. This policy can be associated only
with HTTP/HTTPS listeners."
},
{
    "PolicyAttributeTypeDescriptions": [
        {
            "Cardinality": "ZERO_OR_ONE",
            "AttributeName": "CookieExpirationPeriod",
            "AttributeType": "Long"
        }
    ],
    "PolicyTypeName": "LBCookieStickinessPolicyType",
    "Description": "Stickiness policy with session lifetimes controlled by
the browser (user-agent) or a specified expiration period. This policy can be
associated only with HTTP/HTTPS listeners."
},
{
    "PolicyAttributeTypeDescriptions": [
        .
        .
        .
    ],
    "PolicyTypeName": "SSLNegotiationPolicyType",

```

```

    "Description": "Listener policy that defines the ciphers and protocols
that will be accepted by the load balancer. This policy can be associated only with
HTTPS/SSL listeners."
  },
  {
    "PolicyAttributeTypeDescriptions": [
      {
        "Cardinality": "ONE_OR_MORE",
        "AttributeName": "PublicKeyPolicyName",
        "AttributeType": "PolicyName"
      }
    ],
    "PolicyTypeName": "BackendServerAuthenticationPolicyType",
    "Description": "Policy that controls authentication to back-end server(s)
and contains one or more policies, such as an instance of a PublicKeyPolicyType.
This policy can be associated only with back-end servers that are using HTTPS/SSL."
  }
]
}

```

- API-Einheiten finden Sie [DescribeLoadBalancerPolicyTypes](#) in der AWS CLI Befehlsreferenz.

describe-load-balancers

Das folgende Codebeispiel zeigt die Verwendung `describe-load-balancers`.

AWS CLI

Um Ihre Load Balancer zu beschreiben

In diesem Beispiel werden alle Ihre Load Balancer beschrieben.

Befehl:

```
aws elb describe-load-balancers
```

Um einen Ihrer Load Balancer zu beschreiben

Dieses Beispiel beschreibt den angegebenen Load Balancer.

Befehl:

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```


Die folgende Beispielantwort bezieht sich auf einen HTTPS Load Balancer in einem VPC

Ausgabe:

```
{
  "LoadBalancerDescriptions": [
    {
      "Subnets": [
        "subnet-15aaab61"
      ],
      "CanonicalHostedZoneNameID": "Z3DZXE0EXAMPLE",
      "CanonicalHostedZoneName": "my-load-balancer-1234567890.us-
west-2.elb.amazonaws.com",
      "ListenerDescriptions": [
        {
          "Listener": {
            "InstancePort": 80,
            "LoadBalancerPort": 80,
            "Protocol": "HTTP",
            "InstanceProtocol": "HTTP"
          },
          "PolicyNames": []
        },
        {
          "Listener": {
            "InstancePort": 443,
            "SSLCertificateId": "arn:aws:iam::123456789012:server-certificate/
my-server-cert",
            "LoadBalancerPort": 443,
            "Protocol": "HTTPS",
            "InstanceProtocol": "HTTPS"
          },
          "PolicyNames": [
            "ELBSecurityPolicy-2015-03"
          ]
        }
      ]
    },
    {
      "HealthCheck": {
        "HealthyThreshold": 2,
        "Interval": 30,
        "Target": "HTTP:80/png",
        "Timeout": 3,
        "UnhealthyThreshold": 2
      },
    },
  ],
}
```

```
"VPCId": "vpc-a01106c2",
"BackendServerDescriptions": [
  {
    "InstancePort": 80,
    "PolicyNames": [
      "my-ProxyProtocol-policy"
    ]
  }
],
"Instances": [
  {
    "InstanceId": "i-207d9717"
  },
  {
    "InstanceId": "i-afefb49b"
  }
],
"DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com",
"SecurityGroups": [
  "sg-a61988c3"
],
"Policies": {
  "LBCookieStickinessPolicies": [
    {
      "PolicyName": "my-duration-cookie-policy",
      "CookieExpirationPeriod": 60
    }
  ],
  "AppCookieStickinessPolicies": [],
  "OtherPolicies": [
    "my-PublicKey-policy",
    "my-authentication-policy",
    "my-SSLNegotiation-policy",
    "my-ProxyProtocol-policy",
    "ELBSecurityPolicy-2015-03"
  ]
},
"LoadBalancerName": "my-load-balancer",
"CreatedTime": "2015-03-19T03:24:02.650Z",
"AvailabilityZones": [
  "us-west-2a"
],
"Scheme": "internet-facing",
"SourceSecurityGroup": {
```

```
        "OwnerAlias": "123456789012",
        "GroupName": "my-elb-sg"
    }
}
]
```

- API-Einheiten finden Sie [DescribeLoadBalancers](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Um die einem Load Balancer zugewiesenen Tags zu beschreiben

In diesem Beispiel werden die Tags beschrieben, die dem angegebenen Load Balancer zugewiesen sind.

Befehl:

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

Ausgabe:

```
{
  "TagDescriptions": [
    {
      "Tags": [
        {
          "Value": "lima",
          "Key": "project"
        },
        {
          "Value": "digital-media",
          "Key": "department"
        }
      ],
      "LoadBalancerName": "my-load-balancer"
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie [DescribeTags](#) in der AWS CLI Befehlsreferenz.

detach-load-balancer-from-subnets

Das folgende Codebeispiel zeigt die Verwendung `detach-load-balancer-from-subnets`.

AWS CLI

Um Load Balancer von Subnetzen zu trennen

In diesem Beispiel wird der angegebene Load Balancer vom angegebenen Subnetz getrennt.

Befehl:

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --  
subnets subnet-0ecac448
```

Ausgabe:

```
{  
  "Subnets": [  
    "subnet-15aaab61"  
  ]  
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [DetachLoadBalancerFromSubnets](#).AWS CLI

disable-availability-zones-for-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `disable-availability-zones-for-load-balancer`.

AWS CLI

Um Availability Zones für einen Load Balancer zu deaktivieren

In diesem Beispiel wird die angegebene Availability Zone aus der Gruppe der Availability Zones für den angegebenen Load Balancer entfernt.

Befehl:

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2a
```

Ausgabe:

```
{
  "AvailabilityZones": [
    "us-west-2b"
  ]
}
```

- API-Einheiten finden Sie unter [DisableAvailabilityZonesForLoadBalancer AWS CLI-Befehlsreferenz](#).

enable-availability-zones-for-load-balancer

Das folgende Codebeispiel zeigt die Verwendung von `enable-availability-zones-for-load-balancer`.

AWS CLI

Um Availability Zones für einen Load Balancer zu aktivieren

In diesem Beispiel wird die angegebene Availability Zone zum angegebenen Load Balancer hinzugefügt.

Befehl:

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2b
```

Ausgabe:

```
{
  "AvailabilityZones": [
    "us-west-2a",
    "us-west-2b"
  ]
}
```

- API-Einheiten finden Sie unter [EnableAvailabilityZonesForLoadBalancer AWS CLI-Befehlsreferenz](#).

modify-load-balancer-attributes

Das folgende Codebeispiel zeigt die Verwendung `modify-load-balancer-attributes`.

AWS CLI

Um die Attribute eines Load Balancers zu ändern

In diesem Beispiel wird das `CrossZoneLoadBalancing` Attribut des angegebenen Load Balancers geändert.

Befehl:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

Ausgabe:

```
{
  "LoadBalancerAttributes": {
    "CrossZoneLoadBalancing": {
      "Enabled": true
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

In diesem Beispiel wird das `ConnectionDraining` Attribut des angegebenen Load Balancers geändert.

Befehl:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout\":300}}"
```

Ausgabe:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": true,
      "Timeout": 300
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

- API-Einheiten finden Sie [ModifyLoadBalancerAttributes](#) in der AWS CLI Befehlsreferenz.

register-instances-with-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `register-instances-with-load-balancer`.

AWS CLI

Um Instances bei einem Load Balancer zu registrieren

In diesem Beispiel wird die angegebene Instance beim angegebenen Load Balancer registriert.

Befehl:

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
--instances i-d6f6fae3
```

Ausgabe:

```
{
  "Instances": [
    {
      "InstanceId": "i-d6f6fae3"
    },
    {
      "InstanceId": "i-207d9717"
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie [RegisterInstancesWithLoadBalancer](#) in der AWS CLI Befehlsreferenz.

remove-tags

Das folgende Codebeispiel zeigt die Verwendung `remove-tags`.

AWS CLI

Um Tags aus einem Load Balancer zu entfernen

In diesem Beispiel wird ein Tag aus dem angegebenen Load Balancer entfernt.

Befehl:

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- API-Einheiten finden Sie [RemoveTags](#) in der AWS CLI Befehlsreferenz.

set-load-balancer-listener-ssl-certificate

Das folgende Codebeispiel zeigt die Verwendung `set-load-balancer-listener-ssl-certificate`.

AWS CLI

Um das SSL-Zertifikat für einen HTTPS Load Balancer zu aktualisieren

Dieses Beispiel ersetzt das vorhandene SSL-Zertifikat für den angegebenen HTTPS Load Balancer.

Befehl:

```
aws elb set-load-balancer-listener-ssl-certificate --load-balancer-name my-load-balancer --load-balancer-port 443 --ssl-certificate-id arn:aws:iam::123456789012:server-certificate/new-server-cert
```

- API-Einheiten finden Sie [SetLoadBalancerListenerSslCertificate](#) in der AWS CLI Befehlsreferenz.

set-load-balancer-policies-for-backend-server

Das folgende Codebeispiel zeigt die Verwendung `set-load-balancer-policies-for-backend-server`.

AWS CLI

Um die Richtlinien zu ersetzen, die einem Port für eine Backend-Instance zugeordnet sind

Dieses Beispiel ersetzt die Richtlinien, die derzeit dem angegebenen Port zugeordnet sind.

Befehl:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

Um alle Richtlinien zu entfernen, die derzeit mit einem Port auf Ihrer Backend-Instance verknüpft sind

In diesem Beispiel werden alle Richtlinien entfernt, die dem angegebenen Port zugeordnet sind.

Befehl:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names []
```

Verwenden Sie den `describe-load-balancer-policies` Befehl, um zu bestätigen, dass die Richtlinien entfernt wurden.

- API-Einheiten finden Sie [SetLoadBalancerPoliciesForBackendServer](#) in der AWS CLI Befehlsreferenz.

set-load-balancer-policies-of-listener

Das folgende Codebeispiel zeigt die Verwendung `set-load-balancer-policies-of-listener`.

AWS CLI

Um die mit einem Listener verknüpften Richtlinien zu ersetzen

Dieses Beispiel ersetzt die Richtlinien, die derzeit dem angegebenen Listener zugeordnet sind.

Befehl:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
--load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

Um alle Richtlinien zu entfernen, die Ihrem Listener zugeordnet sind

In diesem Beispiel werden alle Richtlinien entfernt, die derzeit mit dem angegebenen Listener verknüpft sind.

Befehl:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
--load-balancer-port 443 --policy-names []
```

Verwenden Sie den Befehl, um zu bestätigen, dass die Richtlinien aus dem Load Balancer entfernt wurden. `describe-load-balancer-policies`

- API-Einheiten finden Sie [SetLoadBalancerPoliciesOfListener](#) in der AWS CLI Befehlsreferenz.

Elastic Load Balancing — Version 2-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Elastic Load Balancing — Version 2 Aktionen ausführen und gängige Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-listener-certificates

Das folgende Codebeispiel zeigt die Verwendung `add-listener-certificates`.

AWS CLI

Um einem sicheren Listener ein Zertifikat hinzuzufügen

In diesem Beispiel wird das angegebene Zertifikat dem angegebenen sicheren Listener hinzugefügt.

Befehl:

```
aws elbv2 add-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705
```

Ausgabe:

```
{
  "Certificates": [
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
      "IsDefault": false
    }
  ]
}
```

- API-Einheiten finden Sie unter [AddListenerCertificates AWS CLI-Befehlsreferenz](#).

add-tags

Das folgende Codebeispiel zeigt die Verwendung `add-tags`.

AWS CLI

Um einem Load Balancer Tags hinzuzufügen

Im folgenden `add-tags` Beispiel werden dem angegebenen Load Balancer die department Tags `project` und hinzugefügt.

```
aws elbv2 add-tags \
```

```
--resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
--tags "Key=project,Value=Lima" "Key=department,Value=digital-media"
```

- API-Einheiten finden Sie [AddTags](#) in der AWS CLI Befehlsreferenz.

create-listener

Das folgende Codebeispiel zeigt die Verwendung `create-listener`.

AWS CLI

Beispiel 1: Um einen HTTP Listener zu erstellen

Im folgenden `create-listener` Beispiel wird ein HTTP Listener für den angegebenen Application Load Balancer erstellt, der Anfragen an die angegebene Zielgruppe weiterleitet.

```
aws elbv2 create-listener \  
--load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
--protocol HTTP \  
--port 80 \  
--default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Weitere Informationen finden Sie unter [Tutorial: Einen Application Load Balancer mithilfe des erstellen AWS CLI](#) im Benutzerhandbuch für Application Load Balancers.

Beispiel 2: So erstellen Sie einen Listener HTTPS

Im folgenden `create-listener` Beispiel wird ein HTTPS Listener für den angegebenen Application Load Balancer erstellt, der Anfragen an die angegebene Zielgruppe weiterleitet. Sie müssen ein SSL-Zertifikat für einen Listener angeben. HTTPS-Sie können Zertifikate mit AWS Certificate Manager (ACM) erstellen und verwalten. Alternativ können Sie mit den TLS Tools SSL ein Zertifikat erstellen, das Zertifikat von einer Zertifizierungsstelle (CA) signieren lassen und das Zertifikat in AWS Identity and Access Management hochladen (IAM).

```
aws elbv2 create-listener \  
--load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
--protocol HTTPS \  
--port 443 \  
--default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

```

--protocol HTTPS \
--port 443 \
--certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
--ssl-policy ELBSecurityPolicy-2016-08 \
--default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067

```

Weitere Informationen finden [Sie unter Hinzufügen eines HTTPS Listeners](#) im Benutzerhandbuch für Application Load Balancers.

Beispiel 3: So erstellen Sie einen Listener TCP

Im folgenden `create-listener` Beispiel wird ein TCP Listener für den angegebenen Network Load Balancer erstellt, der Anfragen an die angegebene Zielgruppe weiterleitet.

```

aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \
  --protocol TCP \
  --port 80 \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78

```

Weitere Informationen finden Sie unter [Tutorial: Einen Network Load Balancer mithilfe des erstellen AWS CLI](#) im Benutzerhandbuch für Network Load Balancers.

Beispiel 4: So erstellen Sie einen Listener TLS

Im folgenden `create-listener` Beispiel wird ein TLS Listener für den angegebenen Network Load Balancer erstellt, der Anfragen an die angegebene Zielgruppe weiterleitet. Sie müssen ein SSL Zertifikat für einen Listener angeben. TLS

```

aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
  --protocol TLS \
  --port 443 \
  --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
  --ssl-policy ELBSecurityPolicy-2016-08 \

```

```
--default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Weitere Informationen finden Sie unter [TLSSListener für Ihren Network Load Balancer](#) im Benutzerhandbuch für Network Load Balancer.

Beispiel 5: So erstellen Sie einen Listener UDP

Im folgenden `create-listener` Beispiel wird ein UDP Listener für den angegebenen Network Load Balancer erstellt, der Anfragen an die angegebene Zielgruppe weiterleitet.

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \  
  --protocol UDP \  
  --port 53 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

Weitere Informationen finden Sie unter [Tutorial: Einen Network Load Balancer mithilfe des erstellen AWS CLI](#) im Benutzerhandbuch für Network Load Balancers.

Beispiel 6: So erstellen Sie einen Listener für das angegebene Gateway und die angegebene Weiterleitung

Im folgenden `create-listener` Beispiel wird ein Listener für den angegebenen Gateway Load Balancer erstellt, der Anfragen an die angegebene Zielgruppe weiterleitet.

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-  
east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/e0f9b3d5c7f7d3d6 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-  
east-1:850631746142:targetgroup/my-glb-targets/007ca469fae3bb1615
```

Ausgabe:

```
{  
  "Listeners": [  
    {  
      "ListenerArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:listener/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6/afc127db15f925de",
```

```

    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:loadbalancer/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6",
    "DefaultActions": [
      {
        "Type": "forward",
        "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615",
        "ForwardConfig": {
          "TargetGroups": [
            {
              "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615"
            }
          ]
        }
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit Gateway Load Balancers using the AWS CLI](#) im Benutzerhandbuch für Gateway Load Balancers.

- API-Einheiten finden Sie unter [CreateListener AWS CLI Befehlsreferenz](#).

create-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `create-load-balancer`.

AWS CLI

Beispiel 1: So erstellen Sie einen mit dem Internet verbundenen Load Balancer

Im folgenden `create-load-balancer` Beispiel wird ein mit dem Internet verbundener Application Load Balancer erstellt und die Availability Zones für die angegebenen Subnetze aktiviert.

```

aws elbv2 create-load-balancer \
  --name my-load-balancer \
  --subnets subnet-b7d581c0 subnet-8360a9e7

```

Ausgabe:

```
{
  "LoadBalancers": [
    {
      "Type": "application",
      "Scheme": "internet-facing",
      "IpAddressType": "ipv4",
      "VpcId": "vpc-3ac0fb5f",
      "AvailabilityZones": [
        {
          "ZoneName": "us-west-2a",
          "SubnetId": "subnet-8360a9e7"
        },
        {
          "ZoneName": "us-west-2b",
          "SubnetId": "subnet-b7d581c0"
        }
      ],
      "CreatedTime": "2017-08-25T21:26:12.920Z",
      "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
      "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
      "SecurityGroups": [
        "sg-5943793c"
      ],
      "LoadBalancerName": "my-load-balancer",
      "State": {
        "Code": "provisioning"
      },
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tutorial: Einen Application Load Balancer mithilfe des erstellen AWS CLI](#) im Benutzerhandbuch für Application Load Balancers.

Beispiel 2: So erstellen Sie einen internen Load Balancer

Im folgenden `create-load-balancer` Beispiel wird ein interner Application Load Balancer erstellt und die Availability Zones für die angegebenen Subnetze aktiviert.

```
aws elbv2 create-load-balancer \
  --name my-internal-load-balancer \
```



```
--scheme internal \  
--subnets subnet-b7d581c0 subnet-8360a9e7
```

Ausgabe:

```
{  
  "LoadBalancers": [  
    {  
      "Type": "application",  
      "Scheme": "internal",  
      "IpAddressType": "ipv4",  
      "VpcId": "vpc-3ac0fb5f",  
      "AvailabilityZones": [  
        {  
          "ZoneName": "us-west-2a",  
          "SubnetId": "subnet-8360a9e7"  
        },  
        {  
          "ZoneName": "us-west-2b",  
          "SubnetId": "subnet-b7d581c0"  
        }  
      ],  
      "CreatedTime": "2016-03-25T21:29:48.850Z",  
      "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",  
      "DNSName": "internal-my-internal-load-balancer-1529930873.us-  
west-2.elb.amazonaws.com",  
      "SecurityGroups": [  
        "sg-5943793c"  
      ],  
      "LoadBalancerName": "my-internal-load-balancer",  
      "State": {  
        "Code": "provisioning"  
      },  
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/app/my-internal-load-balancer/5b49b8d4303115c2"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Tutorial: Einen Application Load Balancer mithilfe des erstellen AWS CLI](#) im Benutzerhandbuch für Application Load Balancers.

Beispiel 3: So erstellen Sie einen Network Load Balancer

Im folgenden `create-load-balancer` Beispiel wird ein mit dem Internet verbundener Network Load Balancer erstellt und die Availability Zone für das angegebene Subnetz aktiviert. Es verwendet ein Subnetz-Mapping, um die angegebene Elastic IP-Adresse der Netzwerkschnittstelle zuzuordnen, die von den Load Balancer-Knoten für die Availability Zone verwendet wird.

```
aws elbv2 create-load-balancer \  
  --name my-network-load-balancer \  
  --type network \  
  --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a
```

Ausgabe:

```
{  
  "LoadBalancers": [  
    {  
      "Type": "network",  
      "Scheme": "internet-facing",  
      "IpAddressType": "ipv4",  
      "VpcId": "vpc-3ac0fb5f",  
      "AvailabilityZones": [  
        {  
          "LoadBalancerAddresses": [  
            {  
              "IpAddress": "35.161.207.171",  
              "AllocationId": "eipalloc-64d5890a"  
            }  
          ],  
          "ZoneName": "us-west-2b",  
          "SubnetId": "subnet-5264e837"  
        }  
      ],  
      "CreatedTime": "2017-10-15T22:41:25.657Z",  
      "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",  
      "DNSName": "my-network-load-balancer-5d1b75f4f1cee11e.elb.us-  
west-2.amazonaws.com",  
      "LoadBalancerName": "my-network-load-balancer",  
      "State": {  
        "Code": "provisioning"  
      },  
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e"
```

```

    }
  ]
}

```

Weitere Informationen finden Sie unter [Tutorial: Einen Network Load Balancer mithilfe des erstellen AWS CLI](#) im Benutzerhandbuch für Network Load Balancers.

Beispiel 4: So erstellen Sie einen Gateway Load Balancer

Im folgenden `create-load-balancer` Beispiel wird ein Gateway Load Balancer erstellt und die Availability Zones für die angegebenen Subnetze aktiviert.

```

aws elbv2 create-load-balancer \
  --name my-gateway-load-balancer \
  --type gateway \
  --subnets subnet-dc83f691 subnet-a62583f9

```

Ausgabe:

```

{
  "LoadBalancers": [
    {
      "Type": "gateway",
      "VpcId": "vpc-838475fe",
      "AvailabilityZones": [
        {
          "ZoneName": "us-east-1b",
          "SubnetId": "subnet-a62583f9"
        },
        {
          "ZoneName": "us-east-1a",
          "SubnetId": "subnet-dc83f691"
        }
      ],
      "CreatedTime": "2021-07-14T19:33:43.324000+00:00",
      "LoadBalancerName": "my-gateway-load-balancer",
      "State": {
        "Code": "provisioning"
      },
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/dfbb5a7d32cdee79"
    }
  ]
}

```

```
]
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Gateway Load Balancers using the AWS CLI](#) im Benutzerhandbuch für Gateway Load Balancers.

- API-Einheiten finden Sie unter [CreateLoadBalancer AWS CLI Befehlsreferenz](#).

create-rule

Das folgende Codebeispiel zeigt die Verwendung `create-rule`.

AWS CLI

Beispiel 1: Um eine Regel mit einer Pfadbedingung und einer Vorwärtsaktion zu erstellen

Im folgenden `create-rule` Beispiel wird eine Regel erstellt, die Anfragen an die angegebene Zielgruppe weiterleitet, wenn die das angegebene Muster URL enthält.

```
aws elbv2 create-rule \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
  --priority 5 \
  --conditions file://conditions-pattern.json \
  --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Inhalt von `conditions-pattern.json`:

```
[
  {
    "Field": "path-pattern",
    "PathPatternConfig": {
      "Values": ["/images/*"]
    }
  }
]
```

Beispiel 2: Um eine Regel mit einer Hostbedingung und einer festen Antwort zu erstellen

Im folgenden `create-rule` Beispiel wird eine Regel erstellt, die eine feste Antwort liefert, wenn der Hostname im Host-Header mit dem angegebenen Hostnamen übereinstimmt.

```
aws elbv2 create-rule \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \  
  --priority 10 \  
  --conditions file://conditions-host.json \  
  --actions file://actions-fixed-response.json
```

Inhalt von `conditions-host.json`

```
[  
  {  
    "Field": "host-header",  
    "HostHeaderConfig": {  
      "Values": ["*.example.com"]  
    }  
  }  
]
```

Inhalt von `actions-fixed-response.json`

```
[  
  {  
    "Type": "fixed-response",  
    "FixedResponseConfig": {  
      "MessageBody": "Hello world",  
      "StatusCode": "200",  
      "ContentType": "text/plain"  
    }  
  }  
]
```

Beispiel 3: Um eine Regel unter Verwendung einer Quell-IP-Adressbedingung, einer Authentifizierungsaktion und einer Weiterleitungsaktion zu erstellen

Im folgenden `create-rule` Beispiel wird eine Regel erstellt, die den Benutzer authentifiziert, wenn die Quell-IP-Adresse mit der angegebenen IP-Adresse übereinstimmt, und die Anfrage an die angegebene Zielgruppe weiterleitet, wenn die Authentifizierung erfolgreich ist.

```
aws elbv2 create-rule \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \  
  --priority 10 \  
  --conditions file://conditions-host.json \  
  --actions file://actions-fixed-response.json
```

```
--priority 20 \  
--conditions file://conditions-source-ip.json \  
--actions file://actions-authenticate.json
```

Inhalt von conditions-source-ip.json

```
[  
  {  
    "Field": "source-ip",  
    "SourceIpConfig": {  
      "Values": ["192.0.2.0/24", "198.51.100.10/32"]  
    }  
  }  
]
```

Inhalt von actions-authenticate.json

```
[  
  {  
    "Type": "authenticate-oidc",  
    "AuthenticateOidcConfig": {  
      "Issuer": "https://idp-issuer.com",  
      "AuthorizationEndpoint": "https://authorization-endpoint.com",  
      "TokenEndpoint": "https://token-endpoint.com",  
      "UserInfoEndpoint": "https://user-info-endpoint.com",  
      "ClientId": "abcdefghijklmnopqrstuvwxy123456789",  
      "ClientSecret": "123456789012345678901234567890",  
      "SessionCookieName": "my-cookie",  
      "SessionTimeout": 3600,  
      "Scope": "email",  
      "AuthenticationRequestExtraParams": {  
        "display": "page",  
        "prompt": "login"  
      }  
    },  
    "OnUnauthenticatedRequest": "deny"  
  },  
  "Order": 1  
},  
  {  
    "Type": "forward",  
    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:880185128111:targetgroup/cli-test/642a97ecb0e0f26b",  
    "Order": 2  
  }  
]
```

```
}  
]
```

- API-Einheiten finden Sie unter [CreateRule AWS CLI-Befehlsreferenz](#).

create-target-group

Das folgende Codebeispiel zeigt die Verwendung `create-target-group`.

AWS CLI

Beispiel 1: So erstellen Sie eine Zielgruppe für einen Application Load Balancer

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe für einen Application Load Balancer erstellt, in der Sie Ziele nach Instanz-ID registrieren (der Zieltyp ist `instance`). Diese Zielgruppe verwendet das HTTP Protokoll, Port 80 und die Standardeinstellungen für die Integritätsprüfung für eine HTTP Zielgruppe.

```
aws elbv2 create-target-group \  
  --name my-targets \  
  --protocol HTTP \  
  --port 80 \  
  --target-type instance \  
  --vpc-id vpc-3ac0fb5f
```

Ausgabe:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",  
      "TargetGroupName": "my-targets",  
      "Protocol": "HTTP",  
      "Port": 80,  
      "VpcId": "vpc-3ac0fb5f",  
      "HealthCheckProtocol": "HTTP",  
      "HealthCheckPort": "traffic-port",  
      "HealthCheckEnabled": true,  
      "HealthCheckIntervalSeconds": 30,  
      "HealthCheckTimeoutSeconds": 5,  
    }  
  ]  
}
```

```

        "HealthyThresholdCount": 5,
        "UnhealthyThresholdCount": 2,
        "HealthCheckPath": "/",
        "Matcher": {
            "HttpCode": "200"
        },
        "TargetType": "instance",
        "ProtocolVersion": "HTTP1",
        "IpAddressType": "ipv4"
    }
]
}

```

Weitere Informationen finden Sie unter [Eine Zielgruppe erstellen](#) im Benutzerhandbuch für Application Load Balancers.

Beispiel 2: So erstellen Sie eine Zielgruppe, um Traffic von einem Application Load Balancer zu einer Lambda-Funktion weiterzuleiten

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe für einen Application Load Balancer erstellt, wobei das Ziel eine Lambda-Funktion ist (der Zieltyp ist `Lambda`). Gesundheitschecks sind für diese Zielgruppe standardmäßig deaktiviert.

```

aws elbv2 create-target-group \
  --name my-lambda-target \
  --target-type Lambda

```

Ausgabe:

```

{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-lambda-target/a3003e085dbb8ddc",
      "TargetGroupName": "my-lambda-target",
      "HealthCheckEnabled": false,
      "HealthCheckIntervalSeconds": 35,
      "HealthCheckTimeoutSeconds": 30,
      "HealthyThresholdCount": 5,
      "UnhealthyThresholdCount": 2,
      "HealthCheckPath": "/",
      "Matcher": {

```



```

        "HttpCode": "200"
      },
      "TargetType": "lambda",
      "IpAddressType": "ipv4"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Lambda-Funktionen als Ziele](#) im Benutzerhandbuch für Application Load Balancers.

Beispiel 3: So erstellen Sie eine Zielgruppe für einen Network Load Balancer

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe für einen Network Load Balancer erstellt, in der Sie Ziele nach IP-Adresse registrieren (der Zieltyp ist `ip`). Diese Zielgruppe verwendet das TCP Protokoll, Port 80 und die Standardeinstellungen für die Integritätsprüfung für eine TCP Zielgruppe.

```

aws elbv2 create-target-group \
  --name my-ip-targets \
  --protocol TCP \
  --port 80 \
  --target-type ip \
  --vpc-id vpc-3ac0fb5f

```

Ausgabe:

```

{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-ip-targets/b6bba954d1361c78",
      "TargetGroupName": "my-ip-targets",
      "Protocol": "TCP",
      "Port": 80,
      "VpcId": "vpc-3ac0fb5f",
      "HealthCheckEnabled": true,
      "HealthCheckProtocol": "TCP",
      "HealthCheckPort": "traffic-port",
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 10,
      "HealthyThresholdCount": 5,
    }
  ]
}

```

```

        "UnhealthyThresholdCount": 2,
        "TargetType": "ip",
        "IpAddressType": "ipv4"
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen einer Zielgruppe](#) im Benutzerhandbuch für Network Load Balancers.

Beispiel 4: So erstellen Sie eine Zielgruppe für die Weiterleitung von Traffic von einem Network Load Balancer zu einem Application Load Balancer

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe für einen Network Load Balancer erstellt, in der Sie einen Application Load Balancer als Ziel registrieren (der Zieltyp ist `alb`).

```
aws elbv2 create-target-group --name my-alb-target --protocol --port 80 --target-type lab --vpc-id vpc-3ac0fb5f TCP
```

Ausgabe:

```

{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-alb-target/a3003e085dbb8ddc",
      "TargetGroupName": "my-alb-target",
      "Protocol": "TCP",
      "Port": 80,
      "VpcId": "vpc-838475fe",
      "HealthCheckProtocol": "HTTP",
      "HealthCheckPort": "traffic-port",
      "HealthCheckEnabled": true,
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 6,
      "HealthyThresholdCount": 5,
      "UnhealthyThresholdCount": 2,
      "HealthCheckPath": "/",
      "Matcher": {
        "HttpCode": "200-399"
      },
      "TargetType": "alb",
    }
  ]
}

```

```

        "IpAddressType": "ipv4"
      }
    ]
  }

```

Weitere Informationen finden [Sie unter Erstellen einer Zielgruppe mit einem Application Load Balancer als Ziel im Benutzerhandbuch für Network Load Balancers](#).

Beispiel 5: So erstellen Sie eine Zielgruppe für einen Gateway Load Balancer

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe für einen Gateway Load Balancer erstellt, wobei das Ziel eine Instance und das Zielgruppenprotokoll ist GENEVE.

```

aws elbv2 create-target-group \
  --name my-glb-targetgroup \
  --protocol GENEVE \
  --port 6081 \
  --target-type instance \
  --vpc-id vpc-838475fe

```

Ausgabe:

```

{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-glb-targetgroup/00c3d57eacd6f40b6f",
      "TargetGroupName": "my-glb-targetgroup",
      "Protocol": "GENEVE",
      "Port": 6081,
      "VpcId": "vpc-838475fe",
      "HealthCheckProtocol": "TCP",
      "HealthCheckPort": "80",
      "HealthCheckEnabled": true,
      "HealthCheckIntervalSeconds": 10,
      "HealthCheckTimeoutSeconds": 5,
      "HealthyThresholdCount": 5,
      "UnhealthyThresholdCount": 2,
      "TargetType": "instance"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Create a target group < https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html>](https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html) im Gateway Load Balancer User Guide.

- API-Einheiten [CreateTargetGroup AWS CLI](#) finden Sie in der Befehlsreferenz.

delete-listener

Das folgende Codebeispiel zeigt die Verwendung `delete-listener`.

AWS CLI

Um einen Listener zu löschen

Im folgenden `delete-listener` Beispiel wird der angegebene Listener gelöscht.

```
aws elbv2 delete-listener \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- API-Einheiten finden Sie unter [DeleteListener AWS CLI](#) Befehlsreferenz.

delete-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `delete-load-balancer`.

AWS CLI

Um einen Load Balancer zu löschen

Im folgenden `delete-load-balancer` Beispiel wird der angegebene Load Balancer gelöscht.

```
aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- API-Einheiten finden Sie [DeleteLoadBalancer](#) in der AWS CLI Befehlsreferenz.

delete-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-rule`.

AWS CLI

So löschen Sie eine Regel

Im folgenden `delete-rule` Beispiel wird die angegebene Regel gelöscht.

```
aws elbv2 delete-rule \  
  --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- API-Einheiten finden Sie unter [DeleteRule AWS CLI](#) Befehlsreferenz.

`delete-target-group`

Das folgende Codebeispiel zeigt die Verwendung `delete-target-group`.

AWS CLI

Um eine Zielgruppe zu löschen

Im folgenden `delete-target-group` Beispiel wird die angegebene Zielgruppe gelöscht.

```
aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Load Balancers](#) im Application Load Balancer Guide.

- API-Einheiten finden Sie unter [DeleteTargetGroup AWS CLI](#) Befehlsreferenz.

`deregister-targets`

Das folgende Codebeispiel zeigt die Verwendung `deregister-targets`.

AWS CLI

Beispiel 1: Um ein Ziel von einer Zielgruppe abzumelden

Im folgenden `deregister-targets` Beispiel wird die angegebene Instanz aus der angegebenen Zielgruppe entfernt.

```
aws elbv2 deregister-targets \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
  --targets Id=i-1234567890abcdef0
```

Beispiel 2: Um ein Ziel zu deregistrieren, das mithilfe von Port-Overrides registriert wurde

Im folgenden `deregister-targets` Beispiel wird eine Instanz aus einer Zielgruppe entfernt, die mithilfe von Port-Overrides registriert wurde.

```
aws elbv2 deregister-targets \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \
  --targets Id=i-1234567890abcdef0,Port=80 Id=i-1234567890abcdef0,Port=766
```

- API-Einheiten finden Sie unter [DeregisterTargets AWS CLI](#) Befehlsreferenz.

describe-account-limits

Das folgende Codebeispiel zeigt die Verwendung `describe-account-limits`.

AWS CLI

Um Ihre Elastic Load Balancing Balancing-Limits zu beschreiben

Das folgende `describe-account-limits` Beispiel zeigt die Elastic Load Balancing Balancing-Limits für Ihr AWS Konto in der aktuellen Region.

```
aws elbv2 describe-account-limits
```

Ausgabe:

```
{
  "Limits": [
    {
      "Name": "target-groups",
      "Max": "3000"
    },
    {
      "Name": "targets-per-application-load-balancer",
      "Max": "1000"
    }
  ]
}
```

```
  },
  {
    "Name": "listeners-per-application-load-balancer",
    "Max": "50"
  },
  {
    "Name": "rules-per-application-load-balancer",
    "Max": "100"
  },
  {
    "Name": "network-load-balancers",
    "Max": "50"
  },
  {
    "Name": "targets-per-network-load-balancer",
    "Max": "3000"
  },
  {
    "Name": "targets-per-availability-zone-per-network-load-balancer",
    "Max": "500"
  },
  {
    "Name": "listeners-per-network-load-balancer",
    "Max": "50"
  },
  {
    "Name": "condition-values-per-alb-rule",
    "Max": "5"
  },
  {
    "Name": "condition-wildcards-per-alb-rule",
    "Max": "5"
  },
  {
    "Name": "target-groups-per-application-load-balancer",
    "Max": "100"
  },
  {
    "Name": "target-groups-per-action-on-application-load-balancer",
    "Max": "5"
  },
  {
    "Name": "target-groups-per-action-on-network-load-balancer",
    "Max": "1"
```

```
    },
    {
      "Name": "certificates-per-application-load-balancer",
      "Max": "25"
    },
    {
      "Name": "certificates-per-network-load-balancer",
      "Max": "25"
    },
    {
      "Name": "targets-per-target-group",
      "Max": "1000"
    },
    {
      "Name": "target-id-registrations-per-application-load-balancer",
      "Max": "1000"
    },
    {
      "Name": "network-load-balancer-enis-per-vpc",
      "Max": "1200"
    },
    {
      "Name": "application-load-balancers",
      "Max": "50"
    },
    {
      "Name": "gateway-load-balancers",
      "Max": "100"
    },
    {
      "Name": "gateway-load-balancers-per-vpc",
      "Max": "100"
    },
    {
      "Name": "geneve-target-groups",
      "Max": "100"
    },
    {
      "Name": "targets-per-availability-zone-per-gateway-load-balancer",
      "Max": "300"
    }
  ]
}
```


Weitere Informationen finden Sie unter [Kontingente](#) in der AWS Allgemeinen Referenz.

- API-Einheiten finden Sie [DescribeAccountLimits](#) in der AWS CLI Befehlsreferenz.

describe-listener-certificates

Das folgende Codebeispiel zeigt die Verwendung `describe-listener-certificates`.

AWS CLI

Um die Zertifikate für einen sicheren Listener zu beschreiben

In diesem Beispiel werden die Zertifikate für den angegebenen sicheren Listener beschrieben.

Befehl:

```
aws elbv2 describe-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Ausgabe:

```
{
  "Certificates": [
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
      "IsDefault": false
    },
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557",
      "IsDefault": false
    },
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/fe59da96-6f58-4a22-8eed-6d0d50477e1d",
      "IsDefault": true
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeListenerCertificates](#) in der AWS CLI Befehlsreferenz.

describe-listeners

Das folgende Codebeispiel zeigt die Verwendung `describe-listeners`.

AWS CLI

Um einen Zuhörer zu beschreiben

Dieses Beispiel beschreibt den angegebenen Listener.

Befehl:

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:Listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Ausgabe:

```
{
  "Listeners": [
    {
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

Um die Listener für einen Load Balancer zu beschreiben

In diesem Beispiel werden die Listener für den angegebenen Load Balancer beschrieben.

Befehl:

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Ausgabe:

```
{
  "Listeners": [
    {
      "Port": 443,
      "Protocol": "HTTPS",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "SslPolicy": "ELBSecurityPolicy-2015-05",
      "Certificates": [
        {
          "CertificateArn": "arn:aws:iam::123456789012:server-certificate/my-server-cert"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65"
    },
    {
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

```

    }
  ]
}

```

- API-Einheiten finden Sie [DescribeListeners](#) in der AWS CLI Befehlsreferenz.

describe-load-balancer-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-load-balancer-attributes`.

AWS CLI

Zur Beschreibung von Load Balancer-Attributen

Im folgenden `describe-load-balancer-attributes` Beispiel werden die Attribute des angegebenen Load Balancers angezeigt.

```

aws elbv2 describe-load-balancer-attributes \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188

```

Die folgende Beispielausgabe zeigt die Attribute für einen Application Load Balancer.

```

{
  "Attributes": [
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "60",
      "Key": "idle_timeout.timeout_seconds"
    },
    {

```

```
    "Value": "false",
    "Key": "deletion_protection.enabled"
  },
  {
    "Value": "true",
    "Key": "routing.http2.enabled"
  }
]
```

Die folgende Beispielausgabe enthält die Attribute für einen Network Load Balancer.

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "false",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "false",
      "Key": "load_balancing.cross_zone.enabled"
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeLoadBalancerAttributes](#) in der AWS CLI Befehlsreferenz.

describe-load-balancers

Das folgende Codebeispiel zeigt die Verwendung von `describe-load-balancers`.

AWS CLI

Um einen Load Balancer zu beschreiben

Dieses Beispiel beschreibt den angegebenen Load Balancer.

Befehl:

```
aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Ausgabe:

```
{
  "LoadBalancers": [
    {
      "Type": "application",
      "Scheme": "internet-facing",
      "IpAddressType": "ipv4",
      "VpcId": "vpc-3ac0fb5f",
      "AvailabilityZones": [
        {
          "ZoneName": "us-west-2a",
          "SubnetId": "subnet-8360a9e7"
        },
        {
          "ZoneName": "us-west-2b",
          "SubnetId": "subnet-b7d581c0"
        }
      ],
      "CreatedTime": "2016-03-25T21:26:12.920Z",
      "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
      "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
      "SecurityGroups": [
        "sg-5943793c"
      ],
      "LoadBalancerName": "my-load-balancer",
      "State": {
        "Code": "active"
      },
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
    }
  ]
}
```

```
    }  
  ]  
}
```

Um alle Load Balancer zu beschreiben

In diesem Beispiel werden alle Ihre Load Balancer beschrieben.

Befehl:

```
aws elbv2 describe-load-balancers
```

- API-Einheiten finden Sie [DescribeLoadBalancers](#) in der AWS CLI Befehlsreferenz.

describe-rules

Das folgende Codebeispiel zeigt die Verwendung `describe-rules`.

AWS CLI

Beispiel 1: Um eine Regel zu beschreiben

Im folgenden `describe-rules` Beispiel werden Details für die angegebene Regel angezeigt.

```
aws elbv2 describe-rules \  
  --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/  
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

Beispiel 2: Um die Regeln für einen Listener zu beschreiben

Im folgenden `describe-rules` Beispiel werden Details zu den Regeln für den angegebenen Listener angezeigt. Die Ausgabe enthält die Standardregel und alle anderen Regeln, die Sie hinzugefügt haben.

```
aws elbv2 describe-rules \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- API-Einheiten finden Sie [DescribeRules](#) in der AWS CLI Befehlsreferenz.

describe-ssl-policies

Das folgende Codebeispiel zeigt die Verwendung `describe-ssl-policies`.

AWS CLI

Beispiel 1: Um die für die SSL Aushandlung verwendeten Richtlinien nach Load Balancer-Typ aufzulisten

Im folgenden `describe-ssl-policies` Beispiel werden die Namen der Richtlinien angezeigt, die Sie für die SSL Aushandlung mit einem Application Load Balancer verwenden können. Das Beispiel verwendet den `--query` Parameter, um nur die Namen der Richtlinien anzuzeigen.

```
aws elbv2 describe-ssl-policies \  
  --load-balancer-type application \  
  --query SslPolicies[*].Name
```

Ausgabe:

```
[  
  "ELBSecurityPolicy-2016-08",  
  "ELBSecurityPolicy-TLS13-1-2-2021-06",  
  "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",  
  "ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",  
  "ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",  
  "ELBSecurityPolicy-TLS13-1-1-2021-06",  
  "ELBSecurityPolicy-TLS13-1-0-2021-06",  
  "ELBSecurityPolicy-TLS13-1-3-2021-06",  
  "ELBSecurityPolicy-TLS-1-2-2017-01",  
  "ELBSecurityPolicy-TLS-1-1-2017-01",  
  "ELBSecurityPolicy-TLS-1-2-Ext-2018-06",  
  "ELBSecurityPolicy-FS-2018-06",  
  "ELBSecurityPolicy-2015-05",  
  "ELBSecurityPolicy-TLS-1-0-2015-04",  
  "ELBSecurityPolicy-FS-1-2-Res-2019-08",  
  "ELBSecurityPolicy-FS-1-1-2019-08",  
  "ELBSecurityPolicy-FS-1-2-2019-08",  
  "ELBSecurityPolicy-FS-1-2-Res-2020-10"  
]
```

Beispiel 2: Um die Richtlinien aufzulisten, die ein bestimmtes Protokoll unterstützen


```
"ECDHE-ECDSA-AES256-GCM-SHA384",  
"ECDHE-RSA-AES256-GCM-SHA384",  
"ECDHE-ECDSA-AES256-SHA384",  
"ECDHE-RSA-AES256-SHA384"  
]
```

Weitere Informationen finden Sie unter [Sicherheitsrichtlinien](#) im Benutzerhandbuch für Application Load Balancers.

- API-Einheiten finden Sie [DescribeSslPolicies](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Um die einem Load Balancer zugewiesenen Tags zu beschreiben

In diesem Beispiel werden die Tags beschrieben, die dem angegebenen Load Balancer zugewiesen sind.

Befehl:

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Ausgabe:

```
{  
  "TagDescriptions": [  
    {  
      "ResourceArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",  
      "Tags": [  
        {  
          "Value": "lima",  
          "Key": "project"  
        },  
        {  
          "Value": "digital-media",  
          "Key": "department"  
        }  
      ]  
    }  
  ]  
}
```

```

    ]
  }
}

```

- API-Einheiten finden Sie [DescribeTags](#) in der AWS CLI Befehlsreferenz.

describe-target-group-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-target-group-attributes`.

AWS CLI

Um Zielgruppenattribute zu beschreiben

Im folgenden `describe-target-group-attributes` Beispiel werden die Attribute der angegebenen Zielgruppe angezeigt.

```

aws elbv2 describe-target-group-attributes \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067

```

Die Ausgabe enthält die Attribute, wenn das Protokoll HTTP oder HTTPS und der Zieltyp `instance` oder `istip`.

```

{
  "Attributes": [
    {
      "Value": "false",
      "Key": "stickiness.enabled"
    },
    {
      "Value": "300",
      "Key": "deregistration_delay.timeout_seconds"
    },
    {
      "Value": "lb_cookie",
      "Key": "stickiness.type"
    },
    {
      "Value": "86400",

```

```

        "Key": "stickiness.lb_cookie.duration_seconds"
      },
      {
        "Value": "0",
        "Key": "slow_start.duration_seconds"
      }
    ]
  }

```

Die folgende Ausgabe enthält die Attribute, wenn das Protokoll HTTP oder HTTPS und der Zieltyp ist `lambda`.

```

{
  "Attributes": [
    {
      "Value": "false",
      "Key": "lambda.multi_value_headers.enabled"
    }
  ]
}

```

Die folgende Ausgabe enthält die Attribute, wenn das Protokoll TCP, TLSUDP, oder TCP_istUDP.

```

{
  "Attributes": [
    {
      "Value": "false",
      "Key": "proxy_protocol_v2.enabled"
    },
    {
      "Value": "300",
      "Key": "deregistration_delay.timeout_seconds"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeTargetGroupAttributes](#) in der AWS CLI Befehlsreferenz.

describe-target-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-target-groups`.

AWS CLI

Beispiel 1: Um eine Zielgruppe zu beschreiben

Im folgenden `describe-target-groups` Beispiel werden Details für die angegebene Zielgruppe angezeigt.

```
aws elbv2 describe-target-groups \  
  --target-group-arns arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Ausgabe:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",  
      "TargetGroupName": "my-targets",  
      "Protocol": "HTTP",  
      "Port": 80,  
      "VpcId": "vpc-3ac0fb5f",  
      "HealthCheckProtocol": "HTTP",  
      "HealthCheckPort": "traffic-port",  
      "HealthCheckEnabled": true,  
      "HealthCheckIntervalSeconds": 30,  
      "HealthCheckTimeoutSeconds": 5,  
      "HealthyThresholdCount": 5,  
      "UnhealthyThresholdCount": 2,  
      "HealthCheckPath": "/",  
      "Matcher": {  
        "HttpCode": "200"  
      },  
      "LoadBalancerArns": [  
        "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/my-load-balancer/50dc6c495c0c9188"  
      ],  
      "TargetType": "instance",  
      "ProtocolVersion": "HTTP1",  
      "IpAddressType": "ipv4"  
    }  
  ]  
}
```

Beispiel 2: Um alle Zielgruppen für einen Load Balancer zu beschreiben

Im folgenden `describe-target-groups` Beispiel werden Details für alle Zielgruppen für den angegebenen Load Balancer angezeigt. Das Beispiel verwendet den `--query` Parameter, um nur die Namen der Zielgruppen anzuzeigen.

```
aws elbv2 describe-target-groups \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --query TargetGroups[*].TargetGroupName
```

Ausgabe:

```
[  
  "my-instance-targets",  
  "my-ip-targets",  
  "my-lambda-target"  
]
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Application Load Balancers Guide.

- API-Einheiten finden Sie [DescribeTargetGroups](#) in der AWS CLI Befehlsreferenz.

`describe-target-health`

Das folgende Codebeispiel zeigt die Verwendung `describe-target-health`.

AWS CLI

Beispiel 1: Um den Zustand der Ziele für eine Zielgruppe zu beschreiben

Im folgenden `describe-target-health` Beispiel werden Gesundheitsdetails für die Ziele der angegebenen Zielgruppe angezeigt. Diese Ziele sind gesund.

```
aws elbv2 describe-target-health \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Ausgabe:

```
{  
  "TargetHealthDescriptions": [  
    {  
      "TargetHealth": "Healthy"  
    }  
  ]  
}
```

```

    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-ceddcd4d",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    },
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    }
  ]
}

```

Beispiel 2: Um den Zustand eines Ziels zu beschreiben

Im folgenden `describe-target-health` Beispiel werden Integritätsdetails für das angegebene Ziel angezeigt. Dieses Ziel ist fehlerfrei.

```

aws elbv2 describe-target-health \
  --targets Id=i-0f76fade,Port=80 \
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067

```

Ausgabe:

```

{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
    },
  ],
}

```

```

        "TargetHealth": {
            "State": "healthy"
        }
    ]
}

```

Die folgende Beispielausgabe bezieht sich auf ein Ziel, dessen Zielgruppe nicht in einer Aktion für einen Listener angegeben ist. Dieses Ziel kann keinen Datenverkehr vom Load Balancer empfangen.

```

{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "unused",
        "Reason": "Target.NotInUse",
        "Description": "Target group is not configured to receive traffic
from the load balancer"
      }
    }
  ]
}

```

Die folgende Beispielausgabe bezieht sich auf ein Ziel, dessen Zielgruppe gerade in einer Aktion für einen Listener angegeben wurde. Das Ziel wird noch registriert.

```

{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "initial",

```



```

        "Reason": "Elb.RegistrationInProgress",
        "Description": "Target registration is in progress"
    }
}

```

Die folgende Beispielausgabe bezieht sich auf ein fehlerhaftes Ziel.

```

{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "unhealthy",
        "Reason": "Target.Timeout",
        "Description": "Connection to target timed out"
      }
    }
  ]
}

```

Die folgende Beispielausgabe bezieht sich auf ein Ziel, bei dem es sich um eine Lambda-Funktion handelt, und die Integritätsprüfungen sind deaktiviert.

```

{
  "TargetHealthDescriptions": [
    {
      "Target": {
        "Id": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
        "AvailabilityZone": "all",
      },
      "TargetHealth": {
        "State": "unavailable",
        "Reason": "Target.HealthCheckDisabled",
        "Description": "Health checks are not enabled for this target"
      }
    }
  ]
}

```

```
}

```

- API-Einheiten finden Sie unter [DescribeTargetHealth AWS CLI-Befehlsreferenz](#).

modify-listener

Das folgende Codebeispiel zeigt die Verwendung `modify-listener`.

AWS CLI

Beispiel 1: Um die Standardaktion in eine Vorwärtsaktion zu ändern

Im folgenden `modify-listener` Beispiel wird die Standardaktion (in eine Vorwärtsaktion) für den angegebenen Listener geändert.

```
aws elbv2 modify-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```

Ausgabe:

```
{
  "Listeners": [
    {
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f",
          "Type": "forward"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "Port": 80,
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

Beispiel 2: Um die Standardaktion in eine Umleitungsaktion zu ändern

Im folgenden `modify-listener` Beispiel wird die Standardaktion in eine Umleitungsaktion für den angegebenen Listener geändert.

```
aws elbv2 modify-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
  --default-actions Type=redirect,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```

Ausgabe:

```
{
  "Listeners": [
    {
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f",
          "Type": "redirect"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "Port": 80,
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

Beispiel 3: Um das Serverzertifikat zu ändern

In diesem Beispiel wird das Serverzertifikat für den angegebenen HTTPS Listener geändert.

```
aws elbv2 modify-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65 \
  --certificates CertificateArn=arn:aws:iam::123456789012:server-certificate/my-new-server-cert
```

Ausgabe:

```
{
  "Listeners": [
    {
      "Protocol": "HTTPS",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "SslPolicy": "ELBSecurityPolicy-2015-05",
      "Certificates": [
        {
          "CertificateArn": "arn:aws:iam::123456789012:server-certificate/
my-new-server-cert"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "Port": 443,
      "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65"
    }
  ]
}
```

- API-Einheiten finden Sie unter [ModifyListener AWS CLI-Befehlsreferenz](#).

modify-load-balancer-attributes

Das folgende Codebeispiel zeigt die Verwendung `modify-load-balancer-attributes`.

AWS CLI

Um den Löschschutz zu aktivieren

In diesem Beispiel wird der Löschschutz für den angegebenen Load Balancer aktiviert.

Befehl:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --attributes Key=deletion_protection.enabled,Value=true
```

Ausgabe:

```
{
  "Attributes": [
    {
      "Value": "true",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "60",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    }
  ]
}
```

Um das Leerlauf-Timeout zu ändern

In diesem Beispiel wird der Wert für das Leerlauf-Timeout für den angegebenen Load Balancer geändert.

Befehl:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --attributes Key=idle_timeout.timeout_seconds,Value=30
```

Ausgabe:

```
{
  "Attributes": [
    {
      "Value": "30",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "true",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    }
  ]
}
```

Um Zugriffsprotokolle zu aktivieren

In diesem Beispiel werden Zugriffsprotokolle für den angegebenen Load Balancer aktiviert. Beachten Sie, dass der S3-Bucket in derselben Region wie der Load Balancer existieren muss und dass eine Richtlinie angehängt sein muss, die Zugriff auf den Elastic Load Balancing Service gewährt.

Befehl:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --attributes Key=access_logs.s3.enabled,Value=true Key=access_logs.s3.bucket,Value=my-loadbalancer-logs Key=access_logs.s3.prefix,Value=myapp
```

Ausgabe:

```
{
  "Attributes": [
    {
      "Value": "true",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "my-load-balancer-logs",
      "Key": "access_logs.s3.bucket"
    },
    {
      "Value": "myapp",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "60",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "false",
      "Key": "deletion_protection.enabled"
    }
  ]
}
```

- API-Einheiten finden Sie [ModifyLoadBalancerAttributes](#) in der AWS CLI Befehlsreferenz.

modify-rule

Das folgende Codebeispiel zeigt die Verwendung `modify-rule`.

AWS CLI

Um eine Regel zu ändern

Im folgenden `modify-rule` Beispiel werden die Aktionen und Bedingungen für die angegebene Regel aktualisiert.

```
aws elbv2 modify-rule \  
  --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \  
  \
```

```
--conditions Field=path-pattern,Values='/images/*'
--rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

Ausgabe:

```
{
  "Rules": [
    {
      "Priority": "10",
      "Conditions": [
        {
          "Field": "path-pattern",
          "Values": [
            "/images/*"
          ]
        }
      ],
      "RuleArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2/9683b2d02a6cabee",
      "IsDefault": false,
      "Actions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [ModifyRule](#) unter AWS CLI Befehlsreferenz.

modify-target-group-attributes

Das folgende Codebeispiel zeigt die Verwendung `modify-target-group-attributes`.

AWS CLI

Um das Timeout für die Verzögerung bei der Abmeldung zu ändern

In diesem Beispiel wird das Zeitlimit für die Verzögerung bei der Abmeldung auf den angegebenen Wert für die angegebene Zielgruppe festgelegt.

Befehl:

```
aws elbv2 modify-target-group-attributes --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 --attributes Key=deregistration_delay.timeout_seconds,Value=600
```

Ausgabe:

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "stickiness.enabled"
    },
    {
      "Value": "600",
      "Key": "deregistration_delay.timeout_seconds"
    },
    {
      "Value": "lb_cookie",
      "Key": "stickiness.type"
    },
    {
      "Value": "86400",
      "Key": "stickiness.lb_cookie.duration_seconds"
    }
  ]
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [ModifyTargetGroupAttributes](#).AWS CLI

modify-target-group

Das folgende Codebeispiel zeigt die Verwendung `modify-target-group`.

AWS CLI

Um die Konfiguration der Gesundheitsprüfung für eine Zielgruppe zu ändern

Im folgenden `modify-target-group` Beispiel wird die Konfiguration der Integritätsprüfungen geändert, die zur Bewertung des Zustands der Ziele für die angegebene Zielgruppe verwendet werden. Beachten Sie, dass Sie aufgrund der Art und Weise, wie Kommas CLI analysiert werden, den Bereich für die `--matcher` Option in einfache Anführungszeichen statt in doppelte Anführungszeichen setzen müssen.

```
aws elbv2 modify-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f \  
  --health-check-protocol HTTPS \  
  --health-check-port 443 \  
  --matcher HttpCode='200,299'
```

Ausgabe:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f",  
      "TargetGroupName": "my-https-targets",  
      "Protocol": "HTTPS",  
      "Port": 443,  
      "VpcId": "vpc-3ac0fb5f",  
      "HealthCheckProtocol": "HTTPS",  
      "HealthCheckPort": "443",  
      "HealthCheckEnabled": true,  
      "HealthCheckIntervalSeconds": 30,  
      "HealthCheckTimeoutSeconds": 5,  
      "HealthyThresholdCount": 5,  
      "UnhealthyThresholdCount": 2,  
      "Matcher": {  
        "HttpCode": "200,299"  
      },  
      "LoadBalancerArns": [  
        "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/my-load-balancer/50dc6c495c0c9188"  
      ],  
      "TargetType": "instance",  
      "ProtocolVersion": "HTTP1",  
      "IpAddressType": "ipv4"  
    }  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Application Load Balancers Guide.

- API-Einheiten finden Sie [ModifyTargetGroup](#) in der AWS CLI Befehlsreferenz.

register-targets

Das folgende Codebeispiel zeigt die Verwendung `register-targets`.

AWS CLI

Beispiel 1: Um Ziele anhand der Instanz-ID bei einer Zielgruppe zu registrieren

Im folgenden `register-targets` Beispiel werden die angegebenen Instanzen bei einer Zielgruppe registriert. Die Zielgruppe muss den Zieltyp `instance` haben.

```
aws elbv2 register-targets \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
  --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

Beispiel 2: Um Ziele mithilfe von Port-Overrides bei einer Zielgruppe zu registrieren

Im folgenden `register-targets` Beispiel wird die angegebene Instanz mithilfe mehrerer Ports bei einer Zielgruppe registriert. Auf diese Weise können Sie Container auf derselben Instanz wie Ziele in der Zielgruppe registrieren.

```
aws elbv2 register-targets \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \
  --targets Id=i-0598c7d356eba48d7,Port=80 Id=i-0598c7d356eba48d7,Port=766
```

Beispiel 3: Um Ziele anhand der IP-Adresse bei einer Zielgruppe zu registrieren

Im folgenden `register-targets` Beispiel werden die angegebenen IP-Adressen bei einer Zielgruppe registriert. Die Zielgruppe muss den Zieltyp `ip` haben.

```
aws elbv2 register-targets \
```

```
--target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \
--targets Id=10.0.1.15 Id=10.0.1.23
```

Beispiel 4: Um eine Lambda-Funktion als Ziel zu registrieren

Im folgenden `register-targets` Beispiel werden die angegebenen IP-Adressen bei einer Zielgruppe registriert. Die Zielgruppe muss den Zieltyp `lambda` haben. Sie müssen Elastic Load Balancing die Erlaubnis erteilen, die Lambda-Funktion aufzurufen.

```
aws elbv2 register-targets \
--target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \
--targets Id=arn:aws:lambda:us-west-2:123456789012:function:my-function
```

- API-Einheiten finden Sie [RegisterTargets](#) in der AWS CLI Befehlsreferenz.

remove-listener-certificates

Das folgende Codebeispiel zeigt die Verwendung `remove-listener-certificates`.

AWS CLI

Um ein Zertifikat von einem sicheren Listener zu entfernen

In diesem Beispiel wird das angegebene Zertifikat aus dem angegebenen sicheren Listener entfernt.

Befehl:

```
aws elbv2 remove-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705
```

- API-Einheiten finden Sie unter [RemoveListenerCertificates AWS CLI](#) Befehlsreferenz.

remove-tags

Das folgende Codebeispiel zeigt die Verwendung `remove-tags`.

AWS CLI

Um Tags aus einem Load Balancer zu entfernen

Im folgenden `remove-tags` Beispiel werden die `department` Tags `project` und aus dem angegebenen Load Balancer entfernt.

```
aws elbv2 remove-tags \  
  --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --tag-keys project department
```

- API-Einheiten finden Sie [RemoveTags](#) in der AWS CLI Befehlsreferenz.

set-ip-address-type

Das folgende Codebeispiel zeigt die Verwendung `set-ip-address-type`.

AWS CLI

Um den Adresstyp eines Load Balancers festzulegen

In diesem Beispiel wird der Adresstyp des angegebenen Load Balancers auf `dualstack` festgelegt. Den Load Balancer-Subnetzen müssen Blöcke zugeordnet sein. IPv6 CIDR

Befehl:

```
aws elbv2 set-ip-address-type --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --ip-address-type dualstack
```

Ausgabe:

```
{  
  "IpAddressType": "dualstack"  
}
```

- API-Einheiten finden Sie [SetIpAddressType](#) in der AWS CLI Befehlsreferenz.

set-rule-priorities

Das folgende Codebeispiel zeigt die Verwendung `set-rule-priorities`.

AWS CLI

Um die Regelpriorität festzulegen

In diesem Beispiel wird die Priorität der angegebenen Regel festgelegt.

Befehl:

```
aws elbv2 set-rule-priorities --rule-  
priorities RuleArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-  
rule/app/my-load-balancer/50dc6c495c0c9188/  
f2f7dc8efc522ab2/1291d13826f405c3,Priority=5
```

Ausgabe:

```
{  
  "Rules": [  
    {  
      "Priority": "5",  
      "Conditions": [  
        {  
          "Field": "path-pattern",  
          "Values": [  
            "/img/*"  
          ]  
        }  
      ],  
      "RuleArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-  
rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3",  
      "IsDefault": false,  
      "Actions": [  
        {  
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",  
          "Type": "forward"  
        }  
      ]  
    }  
  ]  
}
```

- API-Einheiten finden Sie [SetRulePriorities](#) in der AWS CLI Befehlsreferenz.

set-security-groups

Das folgende Codebeispiel zeigt die Verwendung `set-security-groups`.

AWS CLI

Um eine Sicherheitsgruppe einem Load Balancer zuzuordnen

In diesem Beispiel wird die angegebene Sicherheitsgruppe dem angegebenen Load Balancer zugeordnet.

Befehl:

```
aws elbv2 set-security-groups --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --security-groups sg-5943793c
```

Ausgabe:

```
{
  "SecurityGroupIds": [
    "sg-5943793c"
  ]
}
```

- API-Einheiten finden Sie [SetSecurityGroups](#) in der AWS CLI Befehlsreferenz.

set-subnets

Das folgende Codebeispiel zeigt die Verwendung `set-subnets`.

AWS CLI

Um Availability Zones für einen Load Balancer zu aktivieren

In diesem Beispiel wird die Availability Zone für das angegebene Subnetz für den angegebenen Load Balancer aktiviert.

Befehl:

```
aws elbv2 set-subnets --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --  
subnets subnet-8360a9e7 subnet-b7d581c0
```

Ausgabe:

```
{  
  "AvailabilityZones": [  
    {  
      "SubnetId": "subnet-8360a9e7",  
      "ZoneName": "us-west-2a"  
    },  
    {  
      "SubnetId": "subnet-b7d581c0",  
      "ZoneName": "us-west-2b"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [SetSubnets AWS CLI](#) Befehlsreferenz.

Elastic Transcoder Transcoder-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Elastic Transcoder Aktionen ausführen und gängige Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-job`.

AWS CLI

Um einen Job zu stornieren für ElasticTranscoder

Dadurch wird der angegebene Job für ElasticTranscoder storniert.

Befehl:

```
aws elastictranscoder cancel-job --id 333333333333-abcde3
```

- API-Einheiten finden Sie [CancelJob](#) in der AWS CLI Befehlsreferenz.

create-job

Das folgende Codebeispiel zeigt die Verwendung `create-job`.

AWS CLI

Um einen Job für zu erstellen ElasticTranscoder

Das folgende `create-job` Beispiel erstellt einen Job für ElasticTranscoder.

```
aws elastictranscoder create-job \  
  --pipeline-id 111111111111-abcde1 \  
  --inputs file://inputs.json \  
  --outputs file://outputs.json \  
  --output-key-prefix "recipes/" \  
  --user-metadata file://user-metadata.json
```

Inhalt von `inputs.json`:

```
[{  
  "Key": "ETS_example_file.mp4",  
  "FrameRate": "auto",
```

```
"Resolution": "auto",
"AspectRatio": "auto",
"Interlaced": "auto",
"Container": "mp4"
}]
```

Inhalt von `outputs.json`:

```
[
  {
    "Key": "webm/ETS_example_file-kindlefirehd.webm",
    "Rotate": "0",
    "PresetId": "1351620000001-100250"
  }
]
```

Inhalt von `user-metadata.json`:

```
{
  "Food type": "Italian",
  "Cook book": "recipe notebook"
}
```

Ausgabe:

```
{
  "Job": {
    "Status": "Submitted",
    "Inputs": [
      {
        "Container": "mp4",
        "FrameRate": "auto",
        "Key": "ETS_example_file.mp4",
        "AspectRatio": "auto",
        "Resolution": "auto",
        "Interlaced": "auto"
      }
    ],
    "Playlists": [],
    "Outputs": [
      {
        "Status": "Submitted",
```

```

        "Rotate": "0",
        "PresetId": "1351620000001-100250",
        "Watermarks": [],
        "Key": "webm/ETS_example_file-kindlefirehd.webm",
        "Id": "1"
    }
],
"PipelineId": "333333333333-abcde3",
"OutputKeyPrefix": "recipes/",
"UserMetadata": {
    "Cook book": "recipe notebook",
    "Food type": "Italian"
},
"Output": {
    "Status": "Submitted",
    "Rotate": "0",
    "PresetId": "1351620000001-100250",
    "Watermarks": [],
    "Key": "webm/ETS_example_file-kindlefirehd.webm",
    "Id": "1"
},
"Timing": {
    "SubmitTimeMillis": 1533838012298
},
"Input": {
    "Container": "mp4",
    "FrameRate": "auto",
    "Key": "ETS_example_file.mp4",
    "AspectRatio": "auto",
    "Resolution": "auto",
    "Interlaced": "auto"
},
"Id": "1533838012294-example",
"Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
}
}

```

- API-Einheiten finden Sie [CreateJob](#) in der AWS CLI Befehlsreferenz.

create-pipeline

Das folgende Codebeispiel zeigt die Verwendung `create-pipeline`.

AWS CLI

Um eine Pipeline für zu erstellen ElasticTranscoder

Das folgende create-pipeline Beispiel erstellt eine Pipeline für ElasticTranscoder.

```
aws elastictranscoder create-pipeline \  
  --name Default \  
  --input-bucket salesoffice.example.com-source \  
  --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \  
  --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-  
east-1:111222333444:ETS_Errors \  
  --content-config file://content-config.json \  
  --thumbnail-config file://thumbnail-config.json
```

Inhalt von content-config.json:

```
{  
  "Bucket": "salesoffice.example.com-public-promos",  
  "Permissions": [  
    {  
      "GranteeType": "Email",  
      "Grantee": "marketing-promos@example.com",  
      "Access": [  
        "FullControl"  
      ]  
    }  
  ],  
  "StorageClass": "Standard"  
}
```

Inhalt von thumbnail-config.json:

```
{  
  "Bucket": "salesoffice.example.com-public-promos-thumbnails",  
  "Permissions": [  
    {  
      "GranteeType": "Email",  
      "Grantee": "marketing-promos@example.com",  
      "Access": [  
        "FullControl"  
      ]  
    }  
  ]  
}
```

```

    }
  ],
  "StorageClass": "ReducedRedundancy"
}

```

Ausgabe:

```

{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "salesoffice.example.com-public-promos",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    },
    "Name": "Default",
    "ThumbnailConfig": {
      "Bucket": "salesoffice.example.com-public-promos-thumbnails",
      "StorageClass": "ReducedRedundancy",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    },
    "Notifications": {
      "Completed": "",
      "Warning": "",
      "Progressing": "",
      "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
    }
  },
}

```

```

    "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
    "InputBucket": "salesoffice.example.com-source",
    "Id": "1533765810590-example",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1533765810590-example"
  },
  "Warnings": [
    {
      "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
      "Code": "6006"
    }
  ]
}

```

- API-Einheiten finden Sie [CreatePipeline](#) unter AWS CLI Befehlsreferenz.

create-preset

Das folgende Codebeispiel zeigt die Verwendung `create-preset`.

AWS CLI

Um eine Voreinstellung für zu erstellen ElasticTranscoder

Das folgende `create-preset` Beispiel erstellt eine Voreinstellung für ElasticTranscoder.

```

aws elastictranscoder create-preset \
  --name DefaultPreset \
  --description "Use for published videos" \
  --container mp4 \
  --video file://video.json \
  --audio file://audio.json \
  --thumbnails file://thumbnails.json

```

Inhalt von `video.json`:

```

{
  "Codec": "H.264",

```

```

"CodecOptions":{
  "Profile":"main",
  "Level":"2.2",
  "MaxReferenceFrames":"3",
  "MaxBitRate":"",
  "BufferSize":"",
  "InterlacedMode":"Progressive",
  "ColorSpaceConversionMode":"None"
},
"KeyframesMaxDist":"240",
"FixedGOP":"false",
"BitRate":"1600",
"FrameRate":"auto",
"MaxFrameRate":"30",
"MaxWidth":"auto",
"MaxHeight":"auto",
"SizingPolicy":"Fit",
"PaddingPolicy":"Pad",
"DisplayAspectRatio":"auto",
"Watermarks":[
  {
    "Id":"company logo",
    "MaxWidth":"20%",
    "MaxHeight":"20%",
    "SizingPolicy":"ShrinkToFit",
    "HorizontalAlign":"Right",
    "HorizontalOffset":"10px",
    "VerticalAlign":"Bottom",
    "VerticalOffset":"10px",
    "Opacity":"55.5",
    "Target":"Content"
  }
]
}

```

Inhalt von `audio.json`:

```

{
  "Codec":"AAC",
  "CodecOptions":{
    "Profile":"AAC-LC"
  },
  "SampleRate":"44100",

```

```
"BitRate": "96",
"Channels": "2"
}
```

Inhalt von `thumbnails.json`:

```
{
  "Format": "png",
  "Interval": "120",
  "MaxWidth": "auto",
  "MaxHeight": "auto",
  "SizingPolicy": "Fit",
  "PaddingPolicy": "Pad"
}
```

Ausgabe:

```
{
  "Preset": {
    "Thumbnails": {
      "SizingPolicy": "Fit",
      "MaxWidth": "auto",
      "Format": "png",
      "PaddingPolicy": "Pad",
      "Interval": "120",
      "MaxHeight": "auto"
    },
    "Container": "mp4",
    "Description": "Use for published videos",
    "Video": {
      "SizingPolicy": "Fit",
      "MaxWidth": "auto",
      "PaddingPolicy": "Pad",
      "MaxFrameRate": "30",
      "FrameRate": "auto",
      "MaxHeight": "auto",
      "KeyframesMaxDist": "240",
      "FixedGOP": "false",
      "Codec": "H.264",
      "Watermarks": [
        {
          "SizingPolicy": "ShrinkToFit",
          "VerticalOffset": "10px",

```



```

        "VerticalAlign": "Bottom",
        "Target": "Content",
        "MaxWidth": "20%",
        "MaxHeight": "20%",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10px",
        "Opacity": "55.5",
        "Id": "company logo"
    }
],
"CodecOptions": {
    "Profile": "main",
    "MaxBitRate": "32",
    "InterlacedMode": "Progressive",
    "Level": "2.2",
    "ColorSpaceConversionMode": "None",
    "MaxReferenceFrames": "3",
    "BufferSize": "5"
},
"BitRate": "1600",
"DisplayAspectRatio": "auto"
},
"Audio": {
    "Channels": "2",
    "CodecOptions": {
        "Profile": "AAC-LC"
    },
    "SampleRate": "44100",
    "Codec": "AAC",
    "BitRate": "96"
},
"Type": "Custom",
"Id": "1533765290724-example"
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1533765290724-example",
"Name": "DefaultPreset"
},
"Warning": ""
}

```

- API-Einheiten finden Sie [CreatePreset](#) in der AWS CLI Befehlsreferenz.

delete-pipeline

Das folgende Codebeispiel zeigt die Verwendung `delete-pipeline`.

AWS CLI

Um die angegebene ElasticTranscoder Pipeline zu löschen

Dadurch wird die angegebene ElasticTranscoder Pipeline gelöscht.

Befehl:

```
aws elastictranscoder delete-pipeline --id 111111111111-abcde1
```

Ausgabe:

```
{  
  "Success": "true"  
}
```

- API-Einheiten finden Sie [DeletePipeline](#) in der AWS CLI Befehlsreferenz.

delete-preset

Das folgende Codebeispiel zeigt die Verwendung `delete-preset`.

AWS CLI

Um die angegebene ElasticTranscoder Voreinstellung zu löschen

Dadurch wird die angegebene ElasticTranscoder Voreinstellung gelöscht.

Befehl:

```
aws elastictranscoder delete-preset --id 555555555555-abcde5
```

- API-Einheiten finden Sie [DeletePreset](#) in der AWS CLI Befehlsreferenz.

list-jobs-by-pipeline

Das folgende Codebeispiel zeigt die Verwendung `list-jobs-by-pipeline`.

AWS CLI

Um eine Liste von ElasticTranscoder Jobs in der angegebenen Pipeline abzurufen

In diesem Beispiel wird eine Liste von ElasticTranscoder Jobs in der angegebenen Pipeline abgerufen.

Befehl:

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 111111111111-abcde1
```

Ausgabe:

```
{  
  "Jobs": []  
}
```

- APIEinzelheiten finden Sie unter [ListJobsByPipeline AWS CLI](#) Befehlsreferenz.

list-jobs-by-status

Das folgende Codebeispiel zeigt die Verwendung `list-jobs-by-status`.

AWS CLI

Um eine Liste von ElasticTranscoder Jobs mit dem Status Abgeschlossen abzurufen

In diesem Beispiel wird eine Liste von ElasticTranscoder Aufträgen mit dem Status Abgeschlossen abgerufen.

Befehl:

```
aws elastictranscoder list-jobs-by-status --status Complete
```

Ausgabe:

```
{  
  "Jobs": []  
}
```

- APIEinzelheiten finden Sie unter [ListJobsByStatus AWS CLI](#) Befehlsreferenz.

list-pipelines

Das folgende Codebeispiel zeigt die Verwendung `list-pipelines`.

AWS CLI

Um eine Liste von ElasticTranscoder Pipelines abzurufen

In diesem Beispiel wird eine Liste von ElasticTranscoder Pipelines abgerufen.

Befehl:

```
aws elastictranscoder list-pipelines
```

Ausgabe:

```
{
  "Pipelines": [
    {
      "Status": "Active",
      "ContentConfig": {
        "Bucket": "ets-example",
        "Permissions": []
      },
      "Name": "example-pipeline",
      "ThumbnailConfig": {
        "Bucket": "ets-example",
        "Permissions": []
      },
      "Notifications": {
        "Completed": "arn:aws:sns:us-west-2:123456789012:ets_example",
        "Warning": "",
        "Progressing": "",
        "Error": ""
      },
      "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
      "InputBucket": "ets-example",
      "OutputBucket": "ets-example",
      "Id": "3333333333333-abcde3",
      "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    {
      "Status": "Paused",
```

```
    "ContentConfig": {
      "Bucket": "ets-example",
      "Permissions": []
    },
    "Name": "example-php-test",
    "ThumbnailConfig": {
      "Bucket": "ets-example",
      "Permissions": []
    },
    "Notifications": {
      "Completed": "",
      "Warning": "",
      "Progressing": "",
      "Error": ""
    },
    "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
    "InputBucket": "ets-example",
    "OutputBucket": "ets-example",
    "Id": "3333333333333-abcde2",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde2"
  },
  {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-west-output",
      "Permissions": []
    },
    "Name": "pipeline-west",
    "ThumbnailConfig": {
      "Bucket": "ets-west-output",
      "Permissions": []
    },
    "Notifications": {
      "Completed": "arn:aws:sns:us-west-2:123456789012:ets-notifications",
      "Warning": "",
      "Progressing": "",
      "Error": ""
    },
    "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
    "InputBucket": "ets-west-input",
    "OutputBucket": "ets-west-output",
    "Id": "3333333333333-abcde1",
```

```
        "Arn": "arn:aws:elastictranscoder:us-  
west-2:123456789012:pipeline/333333333333-abcde1"  
      }  
    ]  
  }  
}
```

- API-Einheiten finden Sie unter [List Pipelines AWS CLI](#) Befehlsreferenz.

list-presets

Das folgende Codebeispiel zeigt die Verwendung `list-presets`.

AWS CLI

Um eine Liste von ElasticTranscoder Voreinstellungen abzurufen

In diesem Beispiel wird eine Liste von ElasticTranscoder Voreinstellungen abgerufen.

Befehl:

```
aws elastictranscoder list-presets --max-items 2
```

Ausgabe:

```
{  
  "Presets": [  
    {  
      "Container": "mp4",  
      "Name": "KindleFireHD-preset",  
      "Video": {  
        "Resolution": "1280x720",  
        "FrameRate": "30",  
        "KeyframesMaxDist": "90",  
        "FixedGOP": "false",  
        "Codec": "H.264",  
        "Watermarks": [],  
        "CodecOptions": {  
          "Profile": "main",  
          "MaxReferenceFrames": "3",  
          "ColorSpaceConversionMode": "None",  
          "InterlacedMode": "Progressive",  
          "Level": "4"  
        }  
      }  
    },  
  ],  
}
```

```
        "AspectRatio": "16:9",
        "BitRate": "2200"
    },
    "Audio": {
        "Channels": "2",
        "CodecOptions": {
            "Profile": "AAC-LC"
        },
        "SampleRate": "48000",
        "Codec": "AAC",
        "BitRate": "160"
    },
    "Type": "Custom",
    "Id": "333333333333-abcde2",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/333333333333-abcde2",
    "Thumbnails": {
        "AspectRatio": "16:9",
        "Interval": "60",
        "Resolution": "192x108",
        "Format": "png"
    }
},
{
    "Thumbnails": {
        "AspectRatio": "16:9",
        "Interval": "60",
        "Resolution": "192x108",
        "Format": "png"
    },
    "Container": "mp4",
    "Description": "Custom preset for transcoding jobs",
    "Video": {
        "Resolution": "1280x720",
        "FrameRate": "30",
        "KeyframesMaxDist": "90",
        "FixedGOP": "false",
        "Codec": "H.264",
        "Watermarks": [],
        "CodecOptions": {
            "Profile": "main",
            "MaxReferenceFrames": "3",
            "ColorSpaceConversionMode": "None",
            "InterlacedMode": "Progressive",
```

```

        "Level": "3.1"
      },
      "AspectRatio": "16:9",
      "BitRate": "2200"
    },
    "Audio": {
      "Channels": "2",
      "CodecOptions": {
        "Profile": "AAC-LC"
      },
      "SampleRate": "44100",
      "Codec": "AAC",
      "BitRate": "160"
    },
    "Type": "Custom",
    "Id": "333333333333-abcde3",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/333333333333-abcde3",
    "Name": "Roman's Preset"
  }
],
"NextToken": "eyJQYWdlVG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- API-Einheiten finden Sie unter [ListPresets AWS CLI](#) Befehlsreferenz.

read-job

Das folgende Codebeispiel zeigt die Verwendung `read-job`.

AWS CLI

Um einen ElasticTranscoder Job abzurufen

In diesem Beispiel wird der angegebene ElasticTranscoder Job abgerufen.

Befehl:

```
aws elastictranscoder read-job --id 1533838012294-example
```

Ausgabe:

```
{
```



```
"Job": {
  "Status": "Progressing",
  "Inputs": [
    {
      "Container": "mp4",
      "FrameRate": "auto",
      "Key": "ETS_example_file.mp4",
      "AspectRatio": "auto",
      "Resolution": "auto",
      "Interlaced": "auto"
    }
  ],
  "Playlists": [],
  "Outputs": [
    {
      "Status": "Progressing",
      "Rotate": "0",
      "PresetId": "1351620000001-100250",
      "Watermarks": [],
      "Key": "webm/ETS_example_file-kindlefirehd.webm",
      "Id": "1"
    }
  ],
  "PipelineId": "333333333333-abcde3",
  "OutputKeyPrefix": "recipes/",
  "UserMetadata": {
    "Cook book": "recipe notebook",
    "Food type": "Italian"
  },
  "Output": {
    "Status": "Progressing",
    "Rotate": "0",
    "PresetId": "1351620000001-100250",
    "Watermarks": [],
    "Key": "webm/ETS_example_file-kindlefirehd.webm",
    "Id": "1"
  },
  "Timing": {
    "SubmitTimeMillis": 1533838012298,
    "StartTimeMillis": 1533838013786
  },
  "Input": {
    "Container": "mp4",
    "FrameRate": "auto",
```

```
        "Key": "ETS_example_file.mp4",
        "AspectRatio": "auto",
        "Resolution": "auto",
        "Interlaced": "auto"
    },
    "Id": "1533838012294-example",
    "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}
```

- API-Einheiten finden Sie unter [ReadJob AWS CLI](#) Befehlsreferenz.

read-pipeline

Das folgende Codebeispiel zeigt die Verwendung `read-pipeline`.

AWS CLI

Um eine ElasticTranscoder Pipeline abzurufen

In diesem Beispiel wird die angegebene ElasticTranscoder Pipeline abgerufen.

Befehl:

```
aws elastictranscoder read-pipeline --id 333333333333-abcde3
```

Ausgabe:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
        }
      ]
    }
  }
}
```

```

        "GranteeType": "Email"
      }
    ]
  },
  "Name": "Default",
  "ThumbnailConfig": {
    "Bucket": "ets-example",
    "StorageClass": "ReducedRedundancy",
    "Permissions": [
      {
        "Access": [
          "FullControl"
        ],
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Notifications": {
    "Completed": "",
    "Warning": "",
    "Progressing": "",
    "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-example",
  "Id": "333333333333-abcde3",
  "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde3"
},
  "Warnings": [
    {
      "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
      "Code": "6006"
    }
  ]
}

```

- API-Einheiten finden Sie unter [ReadPipeline AWS CLI-Befehlsreferenz](#).

read-preset

Das folgende Codebeispiel zeigt die Verwendung `read-preset`.

AWS CLI

Um ein ElasticTranscoder Preset abzurufen

In diesem Beispiel wird die angegebene ElasticTranscoder Voreinstellung abgerufen.

Befehl:

```
aws elastictranscoder read-preset --id 1351620000001-500020
```

Ausgabe:

```
{
  "Preset": {
    "Thumbnails": {
      "SizingPolicy": "ShrinkToFit",
      "MaxWidth": "192",
      "Format": "png",
      "PaddingPolicy": "NoPad",
      "Interval": "300",
      "MaxHeight": "108"
    },
    "Container": "fmp4",
    "Description": "System preset: MPEG-Dash Video - 4.8M",
    "Video": {
      "SizingPolicy": "ShrinkToFit",
      "MaxWidth": "1280",
      "PaddingPolicy": "NoPad",
      "FrameRate": "30",
      "MaxHeight": "720",
      "KeyframesMaxDist": "60",
      "FixedGOP": "true",
      "Codec": "H.264",
      "Watermarks": [
        {
          "SizingPolicy": "ShrinkToFit",
          "VerticalOffset": "10%",
          "VerticalAlign": "Top",
          "Target": "Content",
          "MaxWidth": "10%",
```

```
        "MaxHeight": "10%",
        "HorizontalAlign": "Left",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "TopLeft"
    },
    {
        "SizingPolicy": "ShrinkToFit",
        "VerticalOffset": "10%",
        "VerticalAlign": "Top",
        "Target": "Content",
        "MaxWidth": "10%",
        "MaxHeight": "10%",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "TopRight"
    },
    {
        "SizingPolicy": "ShrinkToFit",
        "VerticalOffset": "10%",
        "VerticalAlign": "Bottom",
        "Target": "Content",
        "MaxWidth": "10%",
        "MaxHeight": "10%",
        "HorizontalAlign": "Left",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "BottomLeft"
    },
    {
        "SizingPolicy": "ShrinkToFit",
        "VerticalOffset": "10%",
        "VerticalAlign": "Bottom",
        "Target": "Content",
        "MaxWidth": "10%",
        "MaxHeight": "10%",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "BottomRight"
    }
],
"CodecOptions": {
```

```

        "Profile": "main",
        "MaxBitRate": "4800",
        "InterlacedMode": "Progressive",
        "Level": "3.1",
        "ColorSpaceConversionMode": "None",
        "MaxReferenceFrames": "3",
        "BufferSize": "9600"
    },
    "BitRate": "4800",
    "DisplayAspectRatio": "auto"
},
"Type": "System",
"Id": "1351620000001-500020",
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1351620000001-500020",
"Name": "System preset: MPEG-Dash Video - 4.8M"
}
}

```

- API-Einheiten finden Sie [ReadPreset](#) in der AWS CLI Befehlsreferenz.

update-pipeline-notifications

Das folgende Codebeispiel zeigt die Verwendung `update-pipeline-notifications`.

AWS CLI

Um die Benachrichtigungen einer ElasticTranscoder Pipeline zu aktualisieren

In diesem Beispiel werden die Benachrichtigungen der angegebenen ElasticTranscoder Pipeline aktualisiert.

Befehl:

```

aws elastictranscoder update-pipeline-notifications --id 111111111111-
abcde1 --notifications Progressing=arn:aws:sns:us-west-2:0123456789012:my-
topic,Completed=arn:aws:sns:us-west-2:0123456789012:my-topic,Warning=arn:aws:sns:us-
west-2:0123456789012:my-topic,Error=arn:aws:sns:us-east-1:111222333444:ETS_Errors

```

Ausgabe:

```
{
```

```

"Pipeline": {
  "Status": "Active",
  "ContentConfig": {
    "Bucket": "ets-example",
    "StorageClass": "Standard",
    "Permissions": [
      {
        "Access": [
          "FullControl"
        ],
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Name": "Default",
  "ThumbnailConfig": {
    "Bucket": "ets-example",
    "StorageClass": "ReducedRedundancy",
    "Permissions": [
      {
        "Access": [
          "FullControl"
        ],
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Notifications": {
    "Completed": "arn:aws:sns:us-west-2:0123456789012:my-topic",
    "Warning": "arn:aws:sns:us-west-2:0123456789012:my-topic",
    "Progressing": "arn:aws:sns:us-west-2:0123456789012:my-topic",
    "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-example",
  "Id": "111111111111-abcde1",
  "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/111111111111-abcde1"
}

```

- API-Einheiten finden Sie [UpdatePipelineNotifications](#) unter AWS CLI Befehlsreferenz.

update-pipeline-status

Das folgende Codebeispiel zeigt die Verwendung `update-pipeline-status`.

AWS CLI

Um den Status einer ElasticTranscoder Pipeline zu aktualisieren

In diesem Beispiel wird der Status der angegebenen ElasticTranscoder Pipeline aktualisiert.

Befehl:

```
aws elastictranscoder update-pipeline-status --id 111111111111-abcde1 --  
status Paused
```

Ausgabe:

```
{  
  "Pipeline": {  
    "Status": "Paused",  
    "ContentConfig": {  
      "Bucket": "ets-example",  
      "StorageClass": "Standard",  
      "Permissions": [  
        {  
          "Access": [  
            "FullControl"  
          ],  
          "Grantee": "marketing-promos@example.com",  
          "GranteeType": "Email"  
        }  
      ]  
    },  
    "Name": "Default",  
    "ThumbnailConfig": {  
      "Bucket": "ets-example",  
      "StorageClass": "ReducedRedundancy",  
      "Permissions": [  
        {  
          "Access": [  
            "FullControl"  
          ],  
          "Grantee": "marketing-promos@example.com",
```



```

        "GranteeType": "Email"
      }
    ]
  },
  "Notifications": {
    "Completed": "",
    "Warning": "",
    "Progressing": "",
    "Error": "arn:aws:sns:us-east-1:803981987763:ETS_Errors"
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-example",
  "Id": "111111111111-abcde1",
  "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/111111111111-abcde1"
}
}

```

- API-Einheiten finden Sie [UpdatePipelineStatus](#) unter AWS CLI Befehlsreferenz.

update-pipeline

Das folgende Codebeispiel zeigt die Verwendung `update-pipeline`.

AWS CLI

Um eine ElasticTranscoder Pipeline zu aktualisieren

Im folgenden `update-pipeline` Beispiel wird die angegebene ElasticTranscoder Pipeline aktualisiert.

```

aws elastictranscoder update-pipeline \
  --id 111111111111-abcde1 \
  --name DefaultExample \
  --input-bucket salesoffice.example.com-source \
  --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
  --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
  --content-config file://content-config.json \
  --thumbnail-config file://thumbnail-config.json

```

Inhalt von `content-config.json`:

```
{
  "Bucket": "salesoffice.example.com-public-promos",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",
      "Access": [
        "FullControl"
      ]
    }
  ],
  "StorageClass": "Standard"
}
```

Inhalt von `thumbnail-config.json`:

```
{
  "Bucket": "salesoffice.example.com-public-promos-thumbnails",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",
      "Access": [
        "FullControl"
      ]
    }
  ],
  "StorageClass": "ReducedRedundancy"
}
```

Ausgabe:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ]
        }
      ]
    }
  }
}
```

```
        ],
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Name": "DefaultExample",
  "ThumbnailConfig": {
    "Bucket": "ets-example",
    "StorageClass": "ReducedRedundancy",
    "Permissions": [
      {
        "Access": [
          "FullControl"
        ],
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Notifications": {
    "Completed": "",
    "Warning": "",
    "Progressing": "",
    "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-example",
  "Id": "333333333333-abcde3",
  "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde3"
},
"Warnings": [
  {
    "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
    "Code": "6006"
  }
]
}
```

- API-Einheiten finden Sie [UpdatePipeline](#) in der AWS CLI Befehlsreferenz.

ElastiCache Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren ElastiCache.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-resource

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-resource`.

AWS CLI

Um einer Ressource Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel werden einer Cluster- oder Snapshot-Ressource bis zu 10 Tags, Schlüssel-Wert-Paare, hinzugefügt.

```
aws elasticache add-tags-to-resource \  
  --resource-name "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-  
cluster" \  
  --tags '{"20150202":15, "ElastiCache":"Service"}
```

Ausgabe:

```
{  
  "TagList": [  
    {
```

```
        "Value": "20150202",
        "Key": "APIVersion"
    },
    {
        "Value": "ElastiCache",
        "Key": "Service"
    }
]
}
```

Weitere Informationen finden Sie unter [Kosten überwachen mit Cost Allocation Tags](#) im Elasticache-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AddTagsToResource](#) in der AWS CLI Befehlsreferenz.

authorize-cache-security-group-ingress

Das folgende Codebeispiel zeigt die Verwendung `authorize-cache-security-group-ingress`.

AWS CLI

Um die Cache-Sicherheitsgruppe für den Zugriff zu autorisieren

Das folgende `authorize-cache-security-group-ingress` Beispiel ermöglicht den Netzwerkzugriff auf eine Cache-Sicherheitsgruppe.

```
aws elasticache authorize-cache-security-group-ingress \
  --cache-security-group-name "my-sec-grp" \
  --ec2-security-group-name "my-ec2-sec-grp" \
  --ec2-security-group-owner-id "1234567890"
```

Der Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache im Elasticache-Benutzerhandbuch](#).

- APIEinzelheiten finden Sie [AuthorizeCacheSecurityGroupIngress](#) in der AWS CLI Befehlsreferenz.

batch-apply-update-action

Das folgende Codebeispiel zeigt die Verwendung `batch-apply-update-action`.

AWS CLI

Um ein Service-Update anzuwenden

Im folgenden `batch-apply-update-action` Beispiel wird ein Dienstupdate auf einen Redis-Cluster angewendet.

```
aws elasticache batch-apply-update-action \  
  --service-update-name elc-xxxxx406-xxx \  
  --replication-group-ids test-cluster
```

Ausgabe:

```
{  
  "ProcessedUpdateActions": [  
    {  
      "ReplicationGroupId": "pat-cluster",  
      "ServiceUpdateName": "elc-xxxxx406-xxx",  
      "UpdateActionStatus": "waiting-to-start"  
    }  
  ],  
  "UnprocessedUpdateActions": []  
}
```

Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache im Elasticache-Benutzerhandbuch](#).

- API-Einheiten finden Sie [BatchApplyUpdateAction](#) in der AWS CLI Befehlsreferenz.

batch-stop-update-action

Das folgende Codebeispiel zeigt die Verwendung `batch-stop-update-action`.

AWS CLI

Um ein Service-Update zu beenden

Im folgenden `batch-stop-update-action` Beispiel wird ein Dienstupdate auf einen Redis-Cluster angewendet.

```
aws elasticache batch-stop-update-action \  
  --service-update-name elc-xxxxx406-xxx \  
  --replication-group-ids test-cluster
```

```
--service-update-name elc-xxxxx406-xxx \  
--replication-group-ids test-cluster
```

Ausgabe:

```
{  
  "ProcessedUpdateActions": [  
    {  
      "ReplicationGroupId": "pat-cluster",  
      "ServiceUpdateName": "elc-xxxxx406-xxx",  
      "UpdateActionStatus": "stopping"  
    }  
  ],  
  "UnprocessedUpdateActions": []  
}
```

Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache im Elasticache-Benutzerhandbuch](#).

- API-Einheiten finden Sie [BatchStopUpdateAction](#) in der AWS CLI Befehlsreferenz.

copy-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-snapshot`.

AWS CLI

Um einen Snapshot zu kopieren

Im folgenden `copy-snapshot` Beispiel wird eine Kopie eines vorhandenen Snapshots erstellt.

```
aws elasticache copy-snapshot \  
  --source-snapshot-name "my-snapshot" \  
  --target-snapshot-name "my-snapshot-copy"
```

Ausgabe:

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis3.2",  
    "VpcId": "vpc-3820329f3",  
  }  
}
```

```

    "CacheClusterId": "my-redis4",
    "SnapshotRetentionLimit": 7,
    "NumCacheNodes": 1,
    "SnapshotName": "my-snapshot-copy",
    "CacheClusterCreateTime": "2016-12-21T22:24:04.955Z",
    "AutoMinorVersionUpgrade": true,
    "PreferredAvailabilityZone": "us-east-1c",
    "SnapshotStatus": "creating",
    "SnapshotSource": "manual",
    "SnapshotWindow": "07:00-08:00",
    "EngineVersion": "3.2.4",
    "NodeSnapshots": [
      {
        "CacheSize": "3 MB",
        "SnapshotCreateTime": "2016-12-28T07:00:52Z",
        "CacheNodeId": "0001",
        "CacheNodeCreateTime": "2016-12-21T22:24:04.955Z"
      }
    ],
    "CacheSubnetGroupName": "default",
    "Port": 6379,
    "PreferredMaintenanceWindow": "tue:09:30-tue:10:30",
    "CacheNodeType": "cache.m3.large"
  }
}

```

Weitere Informationen finden Sie unter [Exportieren eines Backup](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [CopySnapshot](#) in der AWS CLI Befehlsreferenz.

create-cache-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-cache-cluster`.

AWS CLI

Um einen Cache-Cluster zu erstellen

Im folgenden `create-cache-cluster` Beispiel wird mithilfe der Redis-Engine ein Cache-Cluster erstellt.

```
aws elasticache create-cache-cluster \
```



```
--cache-cluster-id "cluster-test" \  
--engine redis \  
--cache-node-type cache.m5.large \  
--num-cache-nodes 1
```

Ausgabe:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "cluster-test",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheNodeType": "cache.m5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "CacheClusterStatus": "creating",  
    "NumCacheNodes": 1,  
    "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",  
    "PendingModifiedValues": {},  
    "CacheSecurityGroups": [],  
    "CacheParameterGroup": {  
      "CacheParameterGroupName": "default.redis5.0",  
      "ParameterApplyStatus": "in-sync",  
      "CacheNodeIdsToReboot": []  
    },  
    "CacheSubnetGroupName": "default",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:30-07:30",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [Creating a Cluster](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCacheCluster](#) in der AWS CLI Befehlsreferenz.

create-cache-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `create-cache-parameter-group`.

AWS CLI

Um eine Cache-Parametergruppe zu erstellen

Das folgende `create-cache-parameter-group` Beispiel erstellt eine neue ElastiCache Amazon-Cache-Parametergruppe.

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-family "redis5.0" \  
  --cache-parameter-group-name "mygroup" \  
  --description "mygroup"
```

Ausgabe:

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "mygroup",  
    "CacheParameterGroupFamily": "redis5.0",  
    "Description": "my group"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Parametergruppe](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCacheParameterGroup](#) in der AWS CLI Befehlsreferenz.

create-cache-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `create-cache-subnet-group`.

AWS CLI

Um eine Cache-Subnetzgruppe zu erstellen

Im folgenden `create-cache-subnet-group` Beispiel wird eine neue Cache-Subnetzgruppe erstellt.

```
aws elasticache create-cache-subnet-group \  
  --cache-subnet-group-name "mygroup" \  
  --cache-subnet-group-description "my subnet group" \  
  --vpc-subnet-id "subnet-12345678" \  
  --vpc-id "vpc-12345678"
```

```
--subnet-ids "subnet-xxxxec4f"
```

Ausgabe:

```
{
  "CacheSubnetGroup": {
    "CacheSubnetGroupName": "mygroup",
    "CacheSubnetGroupDescription": "my subnet group",
    "VpcId": "vpc-a3e97cdb",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-xxxxec4f",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        }
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Creating a Cache Subnet Group](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateCacheSubnetGroup AWS CLI Befehlsreferenz](#).

create-global-replication-group

Das folgende Codebeispiel zeigt die Verwendung `create-global-replication-group`.

AWS CLI

Um eine globale Replikationsgruppe zu erstellen

Im folgenden `create-global-replication-group` Beispiel wird eine neue globale Replikationsgruppe erstellt.

```
aws elasticache create-global-replication-group \  
  --global-replication-group-id-suffix my-global-replication-group \  
  --primary-replication-group-id my-primary-cluster
```

Ausgabe:

```
{
  "GlobalReplicationGroup": {
    "GlobalReplicationGroupId": "sgaui-my-global-replication-group",
    "GlobalReplicationGroupDescription": " ",
    "Status": "creating",
    "CacheNodeType": "cache.r5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.6",
    "Members": [
      {
        "ReplicationGroupId": "my-primary-cluster",
        "ReplicationGroupRegion": "us-west-2",
        "Role": "PRIMARY",
        "AutomaticFailover": "enabled",
        "Status": "associating"
      }
    ],
    "ClusterEnabled": true,
    "GlobalNodeGroups": [
      {
        "GlobalNodeGroupId": "sgaui-my-global-replication-group-0001",
        "Slots": "0-16383"
      }
    ],
    "AuthTokenEnabled": false,
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateGlobalReplicationGroup](#).AWS CLI

create-replication-group

Das folgende Codebeispiel zeigt die Verwendung `create-replication-group`.

AWS CLI

Um eine Replikationsgruppe zu erstellen

Im folgenden `create-replication-group` Beispiel wird eine Redis-Replikationsgruppe (Clustermodus deaktiviert) oder eine Redis-Replikationsgruppe (Clustermodus aktiviert) erstellt. Dieser Vorgang ist nur für Redis gültig.

```
aws elasticache create-replication-group \  
  --replication-group-id "mygroup" \  
  --replication-group-description "my group" \  
  --engine "redis" \  
  --cache-node-type "cache.m5.large"
```

Ausgabe:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mygroup",  
    "Description": "my group",  
    "Status": "creating",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "mygroup-001"  
    ],  
    "AutomaticFailover": "disabled",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:00-07:00",  
    "ClusterEnabled": false,  
    "CacheNodeType": "cache.m5.large",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [Creating a Redis Replication Group](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateReplicationGroup AWS CLI](#) Befehlsreferenz.

create-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-snapshot`.

AWS CLI

Um einen Snapshot zu erstellen

Das folgende `create-snapshot` Beispiel erstellt einen Snapshot mit der Redis-Engine.

```
aws elasticache create-snapshot \  
  --snapshot-name mynsnapshot \  
  --cache-cluster-id cluster-test
```

Ausgabe:

```
{  
  "Snapshot": {  
    "SnapshotName": "mynsnapshot",  
    "CacheClusterId": "cluster-test",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "CacheNodeType": "cache.m5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "NumCacheNodes": 1,  
    "PreferredAvailabilityZone": "us-west-2b",  
    "CacheClusterCreateTime": "2020-03-19T03:12:01.483Z",  
    "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",  
    "Port": 6379,  
    "CacheParameterGroupName": "default.redis5.0",  
    "CacheSubnetGroupName": "default",  
    "VpcId": "vpc-a3e97cdb",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:30-07:30",  
    "NodeSnapshots": [  
      {  
        "CacheNodeId": "0001",  
        "CacheSize": "",  
        "CacheNodeCreateTime": "2020-03-19T03:12:01.483Z"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Backup and Restore ElastiCache für Redis](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateSnapshot AWS CLI](#) Befehlsreferenz.

create-user-group

Das folgende Codebeispiel zeigt die Verwendung `create-user-group`.

AWS CLI

Um eine Benutzergruppe zu erstellen

Das folgende `create-user-group` Beispiel erstellt eine neue Benutzergruppe.

```
aws elasticache create-user-group \  
  --user-group-id myusergroup \  
  --engine redis \  
  --user-ids default
```

Ausgabe:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "creating",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx52:usergroup:myusergroup"  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [CreateUserGroup](#).AWS CLI

create-user

Das folgende Codebeispiel zeigt die Verwendung `create-user`.

AWS CLI

Um einen Benutzer zu erstellen

Das folgende `create-user` Beispiel erstellt einen neuen Benutzer.

```
aws elasticache create-user \  
  --user-id user1 \  
  --user-name myUser \  
  --passwords mYnuUzrpAxXw2rdzx \  
  --engine redis \  
  --access-string "on ~app::* -@all +@read"
```

Ausgabe:

```
{  
  "UserId": "user2",  
  "UserName": "myUser",  
  "Status": "active",  
  "Engine": "redis",  
  "AccessString": "on ~app::* -@all +@read +@hash +@bitmap +@geo -setbit -bitfield  
-hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius -  
georadiusbymember",  
  "UserGroupIds": [],  
  "Authentication": {  
    "Type": "password",  
    "PasswordCount": 1  
  },  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [CreateUser](#).AWS CLI

decrease-node-groups-in-global-replication-group

Das folgende Codebeispiel zeigt die Verwendung `decrease-node-groups-in-global-replication-group`.

AWS CLI

Um die Anzahl der Knotengruppen in einer globalen Replikationsgruppe zu verringern

Im Folgenden wird die Anzahl der Knotengruppen mithilfe der Redis-Engine `decrease-node-groups-in-global-replication-group` verringert.

```
aws elasticache decrease-node-groups-in-global-replication-group \  
  --global-replication-group-id sgaui-test \  
  --node-group-count 1 \  
  --apply-immediately \  
  --global-node-groups-to-retain sgaui-test-0003
```

Ausgabe:

```
{  
  "GlobalReplicationGroup":  
  {  
    "GlobalReplicationGroupId": "sgaui-test",  
    "GlobalReplicationGroupDescription": "test",  
    "Status": "modifying",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "test-2",  
        "ReplicationGroupRegion": "us-east-1",  
        "Role": "SECONDARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      },  
      {  
        "ReplicationGroupId": "test-1",  
        "ReplicationGroupRegion": "us-west-2",  
        "Role": "PRIMARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      }  
    ],  
    "ClusterEnabled": true,  
    "GlobalNodeGroups": [  
      {
```

```
        "GlobalNodeGroupId": "sgaui-test-0001",
        "Slots": "0-449,1816-5461"
    },
    {
        "GlobalNodeGroupId": "sgaui-test-0002",
        "Slots": "6827-10922"
    },
    {
        "GlobalNodeGroupId": "sgaui-test-0003",
        "Slots": "10923-14052,15418-16383"
    },
    {
        "GlobalNodeGroupId": "sgaui-test-0004",
        "Slots": "450-1815,5462-6826,14053-15417"
    }
],
"AuthTokenEnabled": false,
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}
```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DecreaseNodeGroupsInGlobalReplicationGroup](#).AWS CLI

decrease-replica-count

Das folgende Codebeispiel zeigt die Verwendung `decrease-replica-count`.

AWS CLI

Um die Anzahl der Replikate zu verringern

Im folgenden `decrease-replica-count` Beispiel wird die Anzahl der Replikate in einer Redis-Replikationsgruppe (Clustermodus deaktiviert) oder die Anzahl der Replikatknoten in einer oder mehreren Knotengruppen (Shards) einer Redis-Replikationsgruppe (Clustermodus aktiviert) dynamisch verringert. Dieser Vorgang wird ohne Cluster-Ausfallzeit ausgeführt.

```
aws elasticache decrease-replica-count \
```

```

--replication-group-id my-cluster \
--apply-immediately \
--new-replica-count 2

```

Ausgabe:

```

{
  "ReplicationGroup": {
    "ReplicationGroupId": "my-cluster",
    "Description": " ",
    "Status": "modifying",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "myrepliac",
      "my-cluster-001",
      "my-cluster-002",
      "my-cluster-003"
    ],
    "NodeGroups": [
      {
        "NodeGroupId": "0001",
        "Status": "modifying",
        "PrimaryEndpoint": {
          "Address": "my-cluster.xxxxx.ng.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "ReaderEndpoint": {
          "Address": "my-cluster-
ro.xxxxx.ng.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "NodeGroupMembers": [
          {
            "CacheClusterId": "myrepliac",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
              "Address":
"myrepliac.xxxxx.0001.usw2.cache.amazonaws.com",
              "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
          },

```

```
        {
            "CacheClusterId": "my-cluster-001",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "my-
cluster-001.xxxxx.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "primary"
        },
        {
            "CacheClusterId": "my-cluster-002",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "my-
cluster-002.xxxxx.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "my-cluster-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "my-
cluster-003.xxxxx.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        }
    ]
}
],
"AutomaticFailover": "disabled",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.xlarge",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
```

```
}
```

Weitere Informationen finden Sie unter [Ändern der Anzahl der Replikate](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DecreaseReplicaCount AWS CLI](#) Befehlsreferenz.

delete-cache-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-cache-cluster`.

AWS CLI

Um einen Cache-Cluster zu löschen

Im folgenden `delete-cache-cluster` Beispiel wird der angegebene, zuvor bereitgestellte Cluster gelöscht. Der Befehl löscht alle zugehörigen Cache-Knoten, Knotenendpunkte und den Cluster selbst. Wenn Sie von diesem Vorgang eine erfolgreiche Antwort erhalten, beginnt Amazon ElastiCache sofort mit dem Löschen des Clusters. Sie können diesen Vorgang nicht abbrechen oder rückgängig machen.

Dieser Vorgang ist für Folgendes nicht gültig:

Redis (Clustermodus aktiviert) Cluster
Ein Cluster, der das zuletzt gelesene Replikat einer Replikationsgruppe ist
Eine Knotengruppe (Shard), für die der Multi-AZ-Modus aktiviert ist
Ein Cluster aus einer Redis-Replikationsgruppe (Clustermodus aktiviert)
Ein Cluster, der sich nicht im verfügbaren Status befindet

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id "my-cluster-002"
```

Ausgabe:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "my-cluster-002",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheNodeType": "cache.r5.xlarge",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",
```

```

    "CacheClusterStatus": "deleting",
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-west-2a",
    "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
    "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
    "PendingModifiedValues": {},
    "NotificationConfiguration": {
        "TopicArn": "arn:aws:sns:us-west-x:xxxxxxx4152:My_Topic",
        "TopicStatus": "active"
    },
    "CacheSecurityGroups": [],
    "CacheParameterGroup": {
        "CacheParameterGroupName": "mygroup",
        "ParameterApplyStatus": "in-sync",
        "CacheNodeIdsToReboot": []
    },
    "CacheSubnetGroupName": "kxkxk",
    "AutoMinorVersionUpgrade": true,
    "SecurityGroups": [
        {
            "SecurityGroupId": "sg-xxxxxxxxxx9836",
            "Status": "active"
        },
        {
            "SecurityGroupId": "sg-xxxxxxxxxxxx7b",
            "Status": "active"
        }
    ],
    "ReplicationGroupId": "my-cluster",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "07:30-08:30",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
}
}

```

[Weitere Informationen finden Sie unter Löschen eines Clusters im Elasticache-Benutzerhandbuch.](#)

- API-Einheiten finden Sie [DeleteCacheCluster](#) in der AWS CLI Befehlsreferenz.

delete-cache-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `delete-cache-parameter-group`.

AWS CLI

Um eine Cache-Parametergruppe zu löschen

Im folgenden `delete-cache-parameter-group` Beispiel wird die angegebene Cache-Parametergruppe gelöscht. Sie können eine Cache-Parametergruppe nicht löschen, wenn sie mit Cache-Clustern verknüpft ist.

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myparamgroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Parametergruppe](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteCacheParameterGroup](#) in der AWS CLI Befehlsreferenz.

`delete-cache-subnet-group`

Das folgende Codebeispiel zeigt die Verwendung `delete-cache-subnet-group`.

AWS CLI

Um eine Cache-Subnetzgruppe zu löschen

Im folgenden `delete-cache-subnet-group` Beispiel wird die angegebene Cache-Subnetzgruppe gelöscht. Sie können eine Cache-Subnetzgruppe nicht löschen, wenn sie mit Clustern verknüpft ist.

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name "mygroup"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Subnetzgruppe](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteCacheSubnetGroup AWS CLI](#) Befehlsreferenz.

delete-global-replication-group

Das folgende Codebeispiel zeigt die Verwendung `delete-global-replication-group`.

AWS CLI

Um eine globale Replikationsgruppe zu löschen

Im folgenden `delete-global-replication-group` Beispiel wird eine neue globale Replikationsgruppe gelöscht.

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my-global-replication-group \  
  --retain-primary-replication-group
```

Ausgabe:

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "sgaui-my-grg",  
    "GlobalReplicationGroupDescription": "my-grg",  
    "Status": "deleting",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "my-cluster-grg",  
        "ReplicationGroupRegion": "us-west-2",  
        "Role": "PRIMARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      }  
    ],  
    "ClusterEnabled": false,  
    "AuthTokenEnabled": false,  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteGlobalReplicationGroup](#). AWS CLI

delete-replication-group

Das folgende Codebeispiel zeigt die Verwendung `delete-replication-group`.

AWS CLI

Um eine Replikationsgruppe zu löschen

Im folgenden `delete-replication-group` Beispiel wird eine bestehende Replikationsgruppe gelöscht. Standardmäßig löscht dieser Vorgang die gesamte Replikationsgruppe, einschließlich der Primärreplikate und aller Lesereplikate. Wenn die Replikationsgruppe nur über eine Primärgruppe verfügt, können Sie optional nur die Read Replicas löschen und dabei die Primärreplikate beibehalten, indem Sie `=true` setzen. `RetainPrimaryCluster`

Wenn Sie von diesem Vorgang eine erfolgreiche Antwort erhalten, beginnt Amazon ElastiCache sofort mit dem Löschen der ausgewählten Ressourcen. Sie können diesen Vorgang nicht stornieren oder rückgängig machen. Gilt nur für Redis.

```
aws elasticache delete-replication-group \  
  --replication-group-id "mygroup"
```

Ausgabe:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mygroup",  
    "Description": "my group",  
    "Status": "deleting",  
    "PendingModifiedValues": {},  
    "AutomaticFailover": "disabled",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:00-07:00",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

- API-Einheiten finden Sie [DeleteReplicationGroup](#) in der AWS CLI Befehlsreferenz.

delete-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-snapshot`.

AWS CLI

So löschen Sie einen Snapshot

Im folgenden `delete-snapshot` Beispiel wurde ein Snapshot mithilfe der Redis-Engine gelöscht.

```
aws elasticache delete-snapshot \  
  --snapshot-name mysnapshot
```

Ausgabe:

```
{  
  "Snapshot": {  
    "SnapshotName": "my-cluster-snapshot",  
    "ReplicationGroupId": "mycluster",  
    "ReplicationGroupDescription": "mycluster",  
    "SnapshotStatus": "deleting",  
    "SnapshotSource": "manual",  
    "CacheNodeType": "cache.r5.xlarge",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "PreferredMaintenanceWindow": "thu:12:00-thu:13:00",  
    "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxxxx152:My_Topic",  
    "Port": 6379,  
    "CacheParameterGroupName": "default.redis5.0.cluster.on",  
    "CacheSubnetGroupName": "default",  
    "VpcId": "vpc-a3e97cdb",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 1,  
    "SnapshotWindow": "13:00-14:00",  
    "NumNodeGroups": 4,  
    "AutomaticFailover": "enabled",  
    "NodeSnapshots": [  
      {  
        "CacheClusterId": "mycluster-0002-003",  
        "NodeGroupId": "0002",  
        "CacheNodeId": "0001",  
        "CacheSize": "6 MB",
```

```

    "CacheNodeCreateTime": "2020-06-18T00:05:44.719000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  },
  {
    "CacheClusterId": "mycluster-0003-003",
    "NodeGroupId": "0003",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2019-12-05T19:13:15.912000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  },
  {
    "CacheClusterId": "mycluster-0004-002",
    "NodeGroupId": "0004",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2019-12-09T19:44:34.324000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  },
  {
    "CacheClusterId": "mycluster-0005-003",
    "NodeGroupId": "0005",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2020-06-18T00:05:44.775000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  }
]
}
}

```

Weitere Informationen finden Sie unter [Backup and Restore ElastiCache für Redis](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteSnapshot AWS CLI](#) Befehlsreferenz.

delete-user-group

Das folgende Codebeispiel zeigt die Verwendung `delete-user-group`.

AWS CLI

Um eine Benutzergruppe zu löschen

Im folgenden `delete-user-group` Beispiel wird eine Benutzergruppe gelöscht.

```
aws elasticache delete-user-group \  
  --user-group-id myusergroup
```

Ausgabe:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "deleting",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [DeleteUserGroup](#).AWS CLI

delete-user

Das folgende Codebeispiel zeigt die Verwendung `delete-user`.

AWS CLI

Benutzer löschen

Das folgende `delete-user` Beispiel löscht einen Benutzer.

```
aws elasticache delete-user \  
  --user-id user2
```

Ausgabe:

```
{  
  "UserId": "user1",  
  "UserName": "myUser",  
  "Status": "deleting",
```

```
"Engine": "redis",
"AccessString": "on ~* +@all",
"UserGroupIds": [
  "myusergroup"
],
"Authentication": {
  "Type": "password",
  "PasswordCount": 1
},
"ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [DeleteUser](#).AWS CLI

describe-cache-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-cache-clusters`.

AWS CLI

Um einen Cache-Cluster zu beschreiben

Das folgende `describe-cache-clusters` Beispiel beschreibt einen Cache-Cluster.

```
aws elasticache describe-cache-clusters
```

Ausgabe:

```
{
  "CacheClusters": [
    {
      "CacheClusterId": "my-cluster-003",
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.5",
      "CacheClusterStatus": "available",
      "NumCacheNodes": 1,
      "PreferredAvailabilityZone": "us-west-2a",
```

```
"CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
"PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
"PendingModifiedValues": {},
"NotificationConfiguration": {
  "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxxx152:My_Topic",
  "TopicStatus": "active"
},
"CacheSecurityGroups": [],
"CacheParameterGroup": {
  "CacheParameterGroupName": "default.redis5.0",
  "ParameterApplyStatus": "in-sync",
  "CacheNodeIdsToReboot": []
},
"CacheSubnetGroupName": "kxkxk",
"AutoMinorVersionUpgrade": true,
"SecurityGroups": [
  {
    "SecurityGroupId": "sg-xxxxxd7b",
    "Status": "active"
  }
],
"ReplicationGroupId": "my-cluster",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "06:30-07:30",
"AuthTokenEnabled": false,
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxxx152:cluster:my-cache-
cluster",
"ReplicationGroupLogDeliveryEnabled": false,
"LogDeliveryConfigurations": [
  {
    "LogType": "slow-log",
    "DestinationType": "cloudwatch-logs",
    "DestinationDetails": {
      "CloudWatchLogsDetails": {
        "LogGroup": "test-log"
      }
    },
    "LogFormat": "text",
    "Status": "active"
  }
]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Managing Clusters](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCacheClusters](#) in der AWS CLI Befehlsreferenz.

describe-cache-engine-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-cache-engine-versions`.

AWS CLI

Um eine Cache-Engine-Version zu beschreiben

Das folgende `describe-cache-engine-versions` Beispiel gibt eine Liste der verfügbaren Cache-Engines und ihrer Versionen zurück.

```
aws elasticache describe-cache-engine-versions \
  --engine "Redis"
```

Ausgabe:

```
{
  "CacheEngineVersions": [
    {
      "Engine": "redis",
      "EngineVersion": "2.6.13",
      "CacheParameterGroupFamily": "redis2.6",
      "CacheEngineDescription": "Redis",
      "CacheEngineVersionDescription": "redis version 2.6.13"
    },
    {
      "Engine": "redis",
      "EngineVersion": "2.8.19",
      "CacheParameterGroupFamily": "redis2.8",
      "CacheEngineDescription": "Redis",
      "CacheEngineVersionDescription": "redis version 2.8.19"
    },
    {
      "Engine": "redis",
      "EngineVersion": "2.8.21",
      "CacheParameterGroupFamily": "redis2.8",
```

```
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 2.8.21"
  },
  {
    "Engine": "redis",
    "EngineVersion": "2.8.22",
    "CacheParameterGroupFamily": "redis2.8",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 2.8.22"
  },
  {
    "Engine": "redis",
    "EngineVersion": "2.8.23",
    "CacheParameterGroupFamily": "redis2.8",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 2.8.23"
  },
  {
    "Engine": "redis",
    "EngineVersion": "2.8.24",
    "CacheParameterGroupFamily": "redis2.8",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 2.8.24"
  },
  {
    "Engine": "redis",
    "EngineVersion": "2.8.6",
    "CacheParameterGroupFamily": "redis2.8",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 2.8.6"
  },
  {
    "Engine": "redis",
    "EngineVersion": "3.2.10",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.10"
  },
  {
    "Engine": "redis",
    "EngineVersion": "3.2.4",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.4"
  }
```



```
  },
  {
    "Engine": "redis",
    "EngineVersion": "3.2.6",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.6"
  },
  {
    "Engine": "redis",
    "EngineVersion": "4.0.10",
    "CacheParameterGroupFamily": "redis4.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 4.0.10"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.0",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.0"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.3",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.3"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.4",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.4"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.5",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.5"
  }
]
```

```
}
```

- API-Einheiten finden Sie [DescribeCacheEngineVersions](#) unter AWS CLI Befehlsreferenz.

describe-cache-parameter-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-cache-parameter-groups`.

AWS CLI

Um eine Cache-Parametergruppe zu beschreiben

Das folgende `describe-cache-parameter-groups` Beispiel gibt eine Liste von Beschreibungen von Cache-Parametergruppen zurück.

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name "mygroup"
```

Ausgabe:

```
{  
  "CacheParameterGroups": [  
    {  
      "CacheParameterGroupName": "mygroup",  
      "CacheParameterGroupFamily": "redis5.0",  
      "Description": " "  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Konfiguration von Engine-Parametern mithilfe von Parametergruppen](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeCacheParameterGroups AWS CLI](#) Befehlsreferenz.

describe-cache-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-cache-parameters`.

AWS CLI

Um Cache-Parameter zu beschreiben

Das folgende Beispiel "describe-cache-parameters" gibt die detaillierte Parameterliste für die angegebene Cache-Parametergruppe zurück.

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name "myparamgroup"
```

Ausgabe:

```
{  
  "Parameters": [  
    {  
      "ParameterName": "activedefrag",  
      "ParameterValue": "yes",  
      "Description": "Enabled active memory defragmentation",  
      "Source": "user",  
      "DataType": "string",  
      "AllowedValues": "yes,no",  
      "IsModifiable": true,  
      "MinimumEngineVersion": "5.0.0",  
      "ChangeType": "immediate"  
    },  
    {  
      "ParameterName": "active-defrag-cycle-max",  
      "ParameterValue": "75",  
      "Description": "Maximal effort for defrag in CPU percentage",  
      "Source": "user",  
      "DataType": "integer",  
      "AllowedValues": "1-75",  
      "IsModifiable": true,  
      "MinimumEngineVersion": "5.0.0",  
      "ChangeType": "immediate"  
    },  
    {  
      "ParameterName": "active-defrag-cycle-min",  
      "ParameterValue": "5",  
      "Description": "Minimal effort for defrag in CPU percentage",  
      "Source": "user",  
      "DataType": "integer",  
      "AllowedValues": "1-75",  
      "IsModifiable": true,  
      "MinimumEngineVersion": "5.0.0",  
      "ChangeType": "immediate"  
    }  
  ],  
}
```

```
{
  "ParameterName": "active-defrag-ignore-bytes",
  "ParameterValue": "104857600",
  "Description": "Minimum amount of fragmentation waste to start active
defrag",
  "Source": "user",
  "DataType": "integer",
  "AllowedValues": "1048576-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "active-defrag-max-scan-fields",
  "ParameterValue": "1000",
  "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
  "Source": "user",
  "DataType": "integer",
  "AllowedValues": "1-1000000",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "active-defrag-threshold-lower",
  "ParameterValue": "10",
  "Description": "Minimum percentage of fragmentation to start active
defrag",
  "Source": "user",
  "DataType": "integer",
  "AllowedValues": "1-100",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "active-defrag-threshold-upper",
  "ParameterValue": "100",
  "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
  "Source": "user",
  "DataType": "integer",
  "AllowedValues": "1-100",
```

```
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "activeresharding",
    "ParameterValue": "yes",
    "Description": "Apply rehashing or not.",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "appendfsync",
    "ParameterValue": "everysec",
    "Description": "fsync policy for AOF persistence",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "always,everysec,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "appendonly",
    "ParameterValue": "no",
    "Description": "Enable Redis persistence.",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-hard-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer hard limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
```

```
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in seconds.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
    "ParameterValue": "33554432",
    "Description": "Pubsub client output buffer hard limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
    "ParameterValue": "8388608",
    "Description": "Pubsub client output buffer soft limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
```

```
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
    "ParameterValue": "60",
    "Description": "Pubsub client output buffer soft limit in seconds.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
    "ParameterValue": "60",
    "Description": "Replica client output buffer soft limit in seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-query-buffer-limit",
    "ParameterValue": "1073741824",
    "Description": "Max size of a single client query buffer",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1048576-1073741824",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "close-on-replica-write",
    "ParameterValue": "yes",
    "Description": "If enabled, clients who attempt to write to a read-only
replica will be disconnected. Applicable to 2.8.23 and higher.",
    "Source": "user",
    "DataType": "string",
```

```
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "cluster-enabled",
    "ParameterValue": "no",
    "Description": "Enable cluster mode",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "cluster-require-full-coverage",
    "ParameterValue": "no",
    "Description": "Whether cluster becomes unavailable if one or more slots
are not covered",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "databases",
    "ParameterValue": "16",
    "Description": "Set the number of databases.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-1200000",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "hash-max-ziplist-entries",
    "ParameterValue": "512",
    "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
```



```
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "hash-max-ziplist-value",
    "ParameterValue": "64",
    "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "hll-sparse-max-bytes",
    "ParameterValue": "3000",
    "Description": "HyperLogLog sparse representation bytes limit",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-16000",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-eviction",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on evictions",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-expire",
    "ParameterValue": "no",
```

```
    "Description": "Perform an asynchronous delete on expired keys",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-server-del",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on key updates",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lfu-decay-time",
    "ParameterValue": "1",
    "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lfu-log-factor",
    "ParameterValue": "10",
    "Description": "The log factor for incrementing key counter for LFU
eviction policy",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
```

```
    "ParameterName": "list-compress-depth",
    "ParameterValue": "0",
    "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "list-max-ziplist-size",
    "ParameterValue": "-2",
    "Description": "The number of entries allowed per internal list node can
be specified as a fixed maximum size or a maximum number of elements",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "-5,-4,-3,-2,-1,1-",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lua-replicate-commands",
    "ParameterValue": "yes",
    "Description": "Always enable Lua effect replication or not",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lua-time-limit",
    "ParameterValue": "5000",
    "Description": "Max execution time of a Lua script in milliseconds. 0
for unlimited execution without warnings.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "5000",
    "IsModifiable": false,
```

```
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxclients",
    "ParameterValue": "65000",
    "Description": "The maximum number of Redis clients.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-65000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "maxmemory-policy",
    "ParameterValue": "volatile-lru",
    "Description": "Max memory policy.",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxmemory-samples",
    "ParameterValue": "3",
    "Description": "Max memory samples.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "min-replicas-max-lag",
    "ParameterValue": "10",
    "Description": "The maximum amount of replica lag in seconds beyond which the master would stop taking writes. A value of 0 means the master always takes writes.",
    "Source": "user",
```

```

        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "min-replicas-to-write",
        "ParameterValue": "0",
        "Description": "The minimum number of replicas that must be present with
lag no greater than min-replicas-max-lag for master to take writes. Setting this to
0 means the master always takes writes.",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "notify-keyspace-events",
        "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
        "Source": "user",
        "DataType": "string",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "proto-max-bulk-len",
        "ParameterValue": "536870912",
        "Description": "Max size of a single element request",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "1048576-536870912",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "rename-commands",
        "ParameterValue": "",

```

```

    "Description": "Redis commands that can be dynamically renamed by the
customer",
    "Source": "user",
    "DataType": "string",
    "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.3",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "repl-backlog-size",
    "ParameterValue": "1048576",
    "Description": "The replication backlog size in bytes for PSYNC. This is
the size of the buffer which accumulates slave data when slave is disconnected for
some time, so that when slave reconnects again, only transfer the portion of data
which the slave missed. Minimum value is 16K.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "16384-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "repl-backlog-ttl",
    "ParameterValue": "3600",
    "Description": "The amount of time in seconds after the master no longer
have any slaves connected for the master to free the replication backlog. A value
of 0 means to never release the backlog.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-allow-chaining",
    "ParameterValue": "no",
    "Description": "Configures if chaining of replicas is allowed",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",

```

```
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-ignore-maxmemory",
    "ParameterValue": "yes",
    "Description": "Determines if replica ignores maxmemory setting by not
evicting items independent from the master",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-lazy-flush",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous flushDB during replica sync",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "reserved-memory-percent",
    "ParameterValue": "25",
    "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "set-max-intset-entries",
    "ParameterValue": "512",
```

```
    "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "slowlog-log-slower-than",
    "ParameterValue": "10000",
    "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "slowlog-max-len",
    "ParameterValue": "128",
    "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "stream-node-max-bytes",
    "ParameterValue": "4096",
    "Description": "The maximum size of a single node in a stream in bytes",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
```



```
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "stream-node-max-entries",
    "ParameterValue": "100",
    "Description": "The maximum number of items a single node in a stream
can contain",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "tcp-keepalive",
    "ParameterValue": "300",
    "Description": "If non-zero, send ACKs every given number of seconds.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "timeout",
    "ParameterValue": "0",
    "Description": "Close connection if client is idle for a given number of
seconds, or never if 0.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0,20-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "zset-max-ziplist-entries",
    "ParameterValue": "128",
    "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed.",
    "Source": "user",
    "DataType": "integer",
```

```

        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "zset-max-ziplist-value",
        "ParameterValue": "64",
        "Description": "The threshold of biggest sorted set entries in order for
the dataset to be compressed.",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    }
]
}

```

Weitere Informationen finden Sie unter [Parameter Management](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCacheParameters](#) in der AWS CLI Befehlsreferenz.

describe-cache-subnet-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-cache-subnet-groups`.

AWS CLI

Um Cache-Subnetzgruppen zu beschreiben

Das folgende `describe-cache-subnet-groups` Beispiel gibt eine Liste von Subnetzgruppen zurück.

```
aws elasticache describe-cache-subnet-groups
```

Ausgabe:

```

{
  "CacheSubnetGroups": [
    {

```

```
"CacheSubnetGroupName": "default",
"CacheSubnetGroupDescription": "Default CacheSubnetGroup",
"VpcId": "vpc-a3e97cdb",
"Subnets": [
  {
    "SubnetIdentifier": "subnet-8d4bacf5",
    "SubnetAvailabilityZone": {
      "Name": "us-west-2b"
    }
  },
  {
    "SubnetIdentifier": "subnet-dde21380",
    "SubnetAvailabilityZone": {
      "Name": "us-west-2c"
    }
  },
  {
    "SubnetIdentifier": "subnet-6485ec4f",
    "SubnetAvailabilityZone": {
      "Name": "us-west-2d"
    }
  },
  {
    "SubnetIdentifier": "subnet-b4ebebff",
    "SubnetAvailabilityZone": {
      "Name": "us-west-2a"
    }
  }
],
{
  "CacheSubnetGroupName": "kxkxk",
  "CacheSubnetGroupDescription": "mygroup",
  "VpcId": "vpc-a3e97cdb",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-b4ebebff",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      }
    }
  ]
},
{
```

```

    "CacheSubnetGroupName": "test",
    "CacheSubnetGroupDescription": "test",
    "VpcId": "vpc-a3e97cdb",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-b4ebebff",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#) im Elasticache-Benutzerhandbuch oder [Subnetze und Subnetzgruppen](#) im Memcached-Benutzerhandbuch.

- Einzelheiten API finden Sie in der Befehlsreferenz. [DescribeCacheSubnetGroups](#) AWS CLI

describe-engine-default-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-engine-default-parameters`.

AWS CLI

Um die Standardparameter der Engine zu beschreiben

Im folgenden `describe-engine-default-parameters` Beispiel werden die Standard-Engine und die Systemparameterinformationen für die angegebene Cache-Engine zurückgegeben.

```

aws elasticache describe-engine-default-parameters \
  --cache-parameter-group-family redis5.0

```

Ausgabe:

```

{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "activedefrag",
        "ParameterValue": "no",

```

```
    "Description": "Enabled active memory defragmentation",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-cycle-max",
    "ParameterValue": "75",
    "Description": "Maximal effort for defrag in CPU percentage",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-cycle-min",
    "ParameterValue": "5",
    "Description": "Minimal effort for defrag in CPU percentage",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-ignore-bytes",
    "ParameterValue": "104857600",
    "Description": "Minimum amount of fragmentation waste to start
active defrag",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1048576-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-max-scan-fields",
```

```

        "ParameterValue": "1000",
        "Description": "Maximum number of set/hash/zset/list fields that
will be processed from the main dictionary scan",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1-1000000",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "active-defrag-threshold-lower",
        "ParameterValue": "10",
        "Description": "Minimum percentage of fragmentation to start active
defrag",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1-100",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "active-defrag-threshold-upper",
        "ParameterValue": "100",
        "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1-100",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "activeresharding",
        "ParameterValue": "yes",
        "Description": "Apply rehashing or not.",
        "Source": "system",
        "DataType": "string",
        "AllowedValues": "yes,no",
        "IsModifiable": false,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "requires-reboot"
    }

```

```
    },
    {
      "ParameterName": "appendfsync",
      "ParameterValue": "everysec",
      "Description": "fsync policy for AOF persistence",
      "Source": "system",
      "DataType": "string",
      "AllowedValues": "always, everysec, no",
      "IsModifiable": false,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "appendonly",
      "ParameterValue": "no",
      "Description": "Enable Redis persistence.",
      "Source": "system",
      "DataType": "string",
      "AllowedValues": "yes, no",
      "IsModifiable": false,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "client-output-buffer-limit-normal-hard-limit",
      "ParameterValue": "0",
      "Description": "Normal client output buffer hard limit in bytes.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "client-output-buffer-limit-normal-soft-limit",
      "ParameterValue": "0",
      "Description": "Normal client output buffer soft limit in bytes.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    }
  ],
  {
    "ParameterName": "client-output-buffer-limit-normal-hard-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer hard limit in bytes.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in bytes.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  }
],
{
  "ParameterName": "client-output-buffer-limit-normal-hard-limit",
  "ParameterValue": "0",
  "Description": "Normal client output buffer hard limit in bytes.",
  "Source": "system",
  "DataType": "integer",
  "AllowedValues": "0-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "client-output-buffer-limit-normal-soft-limit",
  "ParameterValue": "0",
  "Description": "Normal client output buffer soft limit in bytes.",
  "Source": "system",
  "DataType": "integer",
  "AllowedValues": "0-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
}
```

```
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
    "ParameterValue": "33554432",
    "Description": "Pubsub client output buffer hard limit in bytes.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
    "ParameterValue": "8388608",
    "Description": "Pubsub client output buffer soft limit in bytes.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
    "ParameterValue": "60",
    "Description": "Pubsub client output buffer soft limit in seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  }
}
```



```
    },
    {
      "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
      "ParameterValue": "60",
      "Description": "Replica client output buffer soft limit in
seconds.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": false,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "client-query-buffer-limit",
      "ParameterValue": "1073741824",
      "Description": "Max size of a single client query buffer",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "1048576-1073741824",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "close-on-replica-write",
      "ParameterValue": "yes",
      "Description": "If enabled, clients who attempt to write to a read-
only replica will be disconnected. Applicable to 2.8.23 and higher.",
      "Source": "system",
      "DataType": "string",
      "AllowedValues": "yes,no",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "cluster-enabled",
      "ParameterValue": "no",
      "Description": "Enable cluster mode",
      "Source": "system",
      "DataType": "string",
      "AllowedValues": "yes,no",
      "IsModifiable": false,
```

```
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "cluster-require-full-coverage",
    "ParameterValue": "no",
    "Description": "Whether cluster becomes unavailable if one or more
slots are not covered",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "databases",
    "ParameterValue": "16",
    "Description": "Set the number of databases.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-1200000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "hash-max-ziplist-entries",
    "ParameterValue": "512",
    "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "hash-max-ziplist-value",
    "ParameterValue": "64",
    "Description": "The threshold of biggest hash entries in order for
the dataset to be compressed.",
    "Source": "system",
```

```
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "hll-sparse-max-bytes",
    "ParameterValue": "3000",
    "Description": "HyperLogLog sparse representation bytes limit",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-16000",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-eviction",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on evictions",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-expire",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on expired keys",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-server-del",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on key updates",
    "Source": "system",
```

```
        "DataType": "string",
        "AllowedValues": "yes,no",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "lfu-decay-time",
        "ParameterValue": "1",
        "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "lfu-log-factor",
        "ParameterValue": "10",
        "Description": "The log factor for incrementing key counter for LFU
eviction policy",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "list-compress-depth",
        "ParameterValue": "0",
        "Description": "Number of quicklist ziplist nodes from each side
of the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
```

```
    "ParameterName": "list-max-ziplist-size",
    "ParameterValue": "-2",
    "Description": "The number of entries allowed per internal list node
can be specified as a fixed maximum size or a maximum number of elements",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "-5,-4,-3,-2,-1,1-",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lua-replicate-commands",
    "ParameterValue": "yes",
    "Description": "Always enable Lua effect replication or not",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lua-time-limit",
    "ParameterValue": "5000",
    "Description": "Max execution time of a Lua script in milliseconds.
0 for unlimited execution without warnings.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "5000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxclients",
    "ParameterValue": "65000",
    "Description": "The maximum number of Redis clients.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-65000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
```

```
    },
    {
      "ParameterName": "maxmemory-policy",
      "ParameterValue": "volatile-lru",
      "Description": "Max memory policy.",
      "Source": "system",
      "DataType": "string",
      "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "maxmemory-samples",
      "ParameterValue": "3",
      "Description": "Max memory samples.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "1-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "min-replicas-max-lag",
      "ParameterValue": "10",
      "Description": "The maximum amount of replica lag in seconds beyond
which the master would stop taking writes. A value of 0 means the master always
takes writes.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "min-replicas-to-write",
      "ParameterValue": "0",
      "Description": "The minimum number of replicas that must be present
with lag no greater than min-replicas-max-lag for master to take writes. Setting
this to 0 means the master always takes writes.",
      "Source": "system",
```

```

        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "notify-keyspace-events",
        "Description": "The keyspace events for Redis to notify Pub/Sub
clients about. By default all notifications are disabled",
        "Source": "system",
        "DataType": "string",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "proto-max-bulk-len",
        "ParameterValue": "536870912",
        "Description": "Max size of a single element request",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1048576-536870912",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "rename-commands",
        "ParameterValue": "",
        "Description": "Redis commands that can be dynamically renamed by
the customer",
        "Source": "system",
        "DataType": "string",
        "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.3",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "repl-backlog-size",
        "ParameterValue": "1048576",

```

```
    "Description": "The replication backlog size in bytes for PSYNC.  
This is the size of the buffer which accumulates slave data when slave is  
disconnected for some time, so that when slave reconnects again, only transfer the  
portion of data which the slave missed. Minimum value is 16K.",  
    "Source": "system",  
    "DataType": "integer",  
    "AllowedValues": "16384-",  
    "IsModifiable": true,  
    "MinimumEngineVersion": "5.0.0",  
    "ChangeType": "immediate"  
  },  
  {  
    "ParameterName": "repl-backlog-ttl",  
    "ParameterValue": "3600",  
    "Description": "The amount of time in seconds after the master no  
longer have any slaves connected for the master to free the replication backlog. A  
value of 0 means to never release the backlog.",  
    "Source": "system",  
    "DataType": "integer",  
    "AllowedValues": "0-",  
    "IsModifiable": true,  
    "MinimumEngineVersion": "5.0.0",  
    "ChangeType": "immediate"  
  },  
  {  
    "ParameterName": "replica-allow-chaining",  
    "ParameterValue": "no",  
    "Description": "Configures if chaining of replicas is allowed",  
    "Source": "system",  
    "DataType": "string",  
    "AllowedValues": "yes,no",  
    "IsModifiable": false,  
    "MinimumEngineVersion": "5.0.0",  
    "ChangeType": "immediate"  
  },  
  {  
    "ParameterName": "replica-ignore-maxmemory",  
    "ParameterValue": "yes",  
    "Description": "Determines if replica ignores maxmemory setting by  
not evicting items independent from the master",  
    "Source": "system",  
    "DataType": "string",  
    "AllowedValues": "yes,no",  
    "IsModifiable": false,
```



```
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-lazy-flush",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous flushDB during replica
sync",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "reserved-memory-percent",
    "ParameterValue": "25",
    "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "set-max-intset-entries",
    "ParameterValue": "512",
    "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "slowlog-log-slower-than",
    "ParameterValue": "10000",
```

```
        "Description": "The execution time, in microseconds, to exceed in
order for the command to get logged. Note that a negative number disables the slow
log, while a value of zero forces the logging of every command.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "slowlog-max-len",
        "ParameterValue": "128",
        "Description": "The length of the slow log. There is no limit to
this length. Just be aware that it will consume memory. You can reclaim memory used
by the slow log with SLOWLOG RESET.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "stream-node-max-bytes",
        "ParameterValue": "4096",
        "Description": "The maximum size of a single node in a stream in
bytes",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "stream-node-max-entries",
        "ParameterValue": "100",
        "Description": "The maximum number of items a single node in a
stream can contain",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
```

```
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "tcp-keepalive",
    "ParameterValue": "300",
    "Description": "If non-zero, send ACKs every given number of
seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "timeout",
    "ParameterValue": "0",
    "Description": "Close connection if client is idle for a given
number of seconds, or never if 0.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0,20-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "zset-max-ziplist-entries",
    "ParameterValue": "128",
    "Description": "The maximum number of sorted set entries in order
for the dataset to be compressed.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "zset-max-ziplist-value",
    "ParameterValue": "64",
    "Description": "The threshold of biggest sorted set entries in order
for the dataset to be compressed.",
```

```

        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    }
]
}
}

```

- API-Einheiten finden Sie [DescribeEngineDefaultParameters](#) unter AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um Ereignisse einer Replikationsgruppe zu beschreiben

Das folgende `describe-events` Beispiel gibt eine Liste von Ereignissen für eine Replikationsgruppe zurück.

```

aws elasticache describe-events \
  --source-identifier test-cluster \
  --source-type replication-group

```

Ausgabe:

```

{
  "Events": [
    {
      "SourceIdentifier": "test-cluster",
      "SourceType": "replication-group",
      "Message": "Automatic failover has been turned on for replication group
test-cluster",
      "Date": "2020-03-18T23:51:34.457Z"
    },
    {
      "SourceIdentifier": "test-cluster",
      "SourceType": "replication-group",

```

```
    "Message": "Replication group test-cluster created",
    "Date": "2020-03-18T23:50:31.378Z"
  }
]
}
```

Weitere Informationen finden Sie unter [Monitoring Events](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

describe-global-replication-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-global-replication-groups`.

AWS CLI

Um globale Replikationsgruppen zu beschreiben

Im folgenden `describe-global-replication-groups` Beispiel werden Details eines globalen Datenspeichers zurückgegeben.

```
aws elasticache describe-global-replication-groups \
  --global-replication-group-id my-grg
```

Ausgabe:

```
{
  "GlobalReplicationGroups": [
    {
      "GlobalReplicationGroupId": "my-grg",
      "GlobalReplicationGroupDescription": "my-grg",
      "Status": "creating",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.6",
      "ClusterEnabled": false,
      "AuthTokenEnabled": false,
      "TransitEncryptionEnabled": false,
      "AtRestEncryptionEnabled": false
    }
  ]
}
```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DescribeGlobalReplicationGroups](#).AWS CLI

describe-replication-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-replication-groups`.

AWS CLI

Um eine Liste mit Details zur Replikationsgruppe zurückzugeben

Im folgenden `describe-replication-groups` Beispiel werden die Replikationsgruppen zurückgegeben.

```
aws elasticache describe-replication-groups
```

Ausgabe:

```
{
  "ReplicationGroups": [
    {
      "ReplicationGroupId": "my-cluster",
      "Description": "mycluster",
      "Status": "available",
      "PendingModifiedValues": {},
      "MemberClusters": [
        "pat-cluster-001",
        "pat-cluster-002",
        "pat-cluster-003",
        "pat-cluster-004"
      ],
      "NodeGroups": [
        {
          "NodeGroupId": "0001",
          "Status": "available",
          "PrimaryEndpoint": {
            "Address": "my-
cluster.xxxxih.ng.0001.usw2.cache.amazonaws.com",
            "Port": 6379
          },
          "ReaderEndpoint": {
```

```
        "Address": "my-cluster-
ro.xxxxih.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "NodeGroupMembers": [
        {
            "CacheClusterId": "my-cluster-001",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "pat-
cluster-001.xxxxih.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "primary"
        },
        {
            "CacheClusterId": "my-cluster-002",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "pat-
cluster-002.xxxxih.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "my-cluster-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "pat-
cluster-003.xxxxih.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "my-cluster-004",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "pat-
cluster-004.xxxxih.0001.usw2.cache.amazonaws.com",
```

```

        "Port": 6379
      },
      "PreferredAvailabilityZone": "us-west-2a",
      "CurrentRole": "replica"
    }
  ]
}
],
"AutomaticFailover": "disabled",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.xlarge",
"AuthTokenEnabled": false,
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxxx152:replicationgroup:my-cluster",
"LogDeliveryConfigurations": [
  {
    "LogType": "slow-log",
    "DestinationType": "cloudwatch-logs",
    "DestinationDetails": {
      "CloudWatchLogsDetails": {
        "LogGroup": "test-log"
      }
    },
    "LogFormat": "json",
    "Status": "active"
  }
]
}
]
}
}

```

Weitere Informationen finden Sie unter [Managing Clusters](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReplicationGroups](#) in der AWS CLI Befehlsreferenz.

describe-reserved-cache-nodes-offerings

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-cache-nodes-offerings`.

AWS CLI

Um zu beschreiben reserved-cache-nodes-offerings

Das folgende describe-reserved-cache-nodes-offerings Beispiel gibt Details einer reserved-cache-node Option zurück.

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Ausgabe:

```
{
  "ReservedCacheNodesOfferings": [
    {
      "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eachbcd8e5",
      "CacheNodeType": "cache.t3.small",
      "Duration": 31536000,
      "FixedPrice": 97.0,
      "UsagePrice": 0.0,
      "ProductDescription": "memcached",
      "OfferingType": "Partial Upfront",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.011,
          "RecurringChargeFrequency": "Hourly"
        }
      ]
    },
    {
      "ReservedCacheNodesOfferingId": "0443a27b-4da5-4b90-b92d-929fbd7abed2",
      "CacheNodeType": "cache.m3.2xlarge",
      "Duration": 31536000,
      "FixedPrice": 1772.0,
      "UsagePrice": 0.0,
      "ProductDescription": "redis",
      "OfferingType": "Heavy Utilization",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.25,
          "RecurringChargeFrequency": "Hourly"
        }
      ]
    }
  ],
}
```

```

    ...
  ]
}

```

Weitere Informationen finden Sie unter Informationen [über Reserved Node Offerings](#) im Elasticache Redis User Guide oder [Getting Info about Reserved Node Offerings](#) im Elasticache Memcached User Guide.

- API-Einheiten finden Sie unter Befehlsreferenz. [DescribeReservedCacheNodesOfferings](#) AWS CLI

describe-reserved-cache-nodes

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-cache-nodes`.

AWS CLI

Um reservierte Cache-Knoten zu beschreiben

Das folgende `describe-reserved-cache-nodes` Beispiel gibt Informationen über reservierte Cache-Knoten für dieses Konto oder über den angegebenen reservierten Cache-Knoten zurück.

```
aws elasticache describe-reserved-cache-nodes
```

Ausgabe:

```

{
  "ReservedCacheNodes": [
    {
      "ReservedCacheNodeId": "mynode",
      "ReservedCacheNodesOfferingId": "xxxxxxxxx-xxxxx-xxxxx-xxxx-xxxxxxxxx71",
      "CacheNodeType": "cache.t3.small",
      "StartTime": "2019-12-06T02:50:44.003Z",
      "Duration": 31536000,
      "FixedPrice": 0.0,
      "UsagePrice": 0.0,
      "CacheNodeCount": 1,
      "ProductDescription": "redis",
      "OfferingType": "No Upfront",
      "State": "payment-pending",
      "RecurringCharges": [

```

```

        {
            "RecurringChargeAmount": 0.023,
            "RecurringChargeFrequency": "Hourly"
        }
    ],
    "ReservationARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxxx52:reserved-instance:mynode"
}
]
}

```

Weitere Informationen finden Sie unter [Managing Costs with Reserved Nodes](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReservedCacheNodes](#) in der AWS CLI Befehlsreferenz.

describe-service-updates

Das folgende Codebeispiel zeigt die Verwendung `describe-service-updates`.

AWS CLI

Um Service-Updates zu beschreiben

Im folgenden `describe-service-updates` Beispiel werden Details zu Dienstupdates zurückgegeben.

```
aws elasticache describe-service-updates
```

Ausgabe:

```

{
  "ServiceUpdates": [
    {
      "ServiceUpdateName": "elc-xxxxxxxx7-001",
      "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
      "ServiceUpdateEndDate": "2020-02-09T15:59:59Z",
      "ServiceUpdateSeverity": "important",
      "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
      "ServiceUpdateStatus": "available",
      "ServiceUpdateDescription": "Upgrades to improve the security,
reliability, and operational performance of your ElastiCache nodes",
      "ServiceUpdateType": "security-update",
    }
  ]
}

```

```

    "Engine": "redis, memcached",
    "EngineVersion": "redis 2.6.13 and onwards, memcached 1.4.5 and
onwards",
    "AutoUpdateAfterRecommendedApplyByDate": false,
    "EstimatedUpdateTime": "30 minutes per node"
  },
  {
    "ServiceUpdateName": "elc-xxxxxxxx4-001",
    "ServiceUpdateReleaseDate": "2019-06-11T15:00:00Z",
    "ServiceUpdateEndDate": "2019-10-01T09:24:00Z",
    "ServiceUpdateSeverity": "important",
    "ServiceUpdateRecommendedApplyByDate": "2019-07-11T14:59:59Z",
    "ServiceUpdateStatus": "expired",
    "ServiceUpdateDescription": "Upgrades to improve the security,
reliability, and operational performance of your ElastiCache nodes",
    "ServiceUpdateType": "security-update",
    "Engine": "redis",
    "EngineVersion": "redis 3.2.6, redis 4.0 and onwards",
    "AutoUpdateAfterRecommendedApplyByDate": false,
    "EstimatedUpdateTime": "30 minutes per node"
  }
]
}

```

- API-Einheiten finden Sie [DescribeServiceUpdates](#) unter AWS CLI Befehlsreferenz.

describe-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshots`.

AWS CLI

Um Schnappschüsse zu beschreiben

Das folgende „`describe-snapshots`“-Beispiel gibt Informationen über Ihre Cluster- oder Replikationsgruppen-Snapshots zurück.

```
aws elasticache describe-snapshots
```

Ausgabe:

```
{
```

```
"Snapshots": [
  {
    "SnapshotName": "automatic.my-cluster2-002-2019-12-05-06-38",
    "CacheClusterId": "my-cluster2-002",
    "SnapshotStatus": "available",
    "SnapshotSource": "automated",
    "CacheNodeType": "cache.r5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.5",
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-west-2a",
    "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
    "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
    "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxx52:My_Topic",
    "Port": 6379,
    "CacheParameterGroupName": "default.redis5.0",
    "CacheSubnetGroupName": "kxkxk",
    "VpcId": "vpc-a3e97cdb",
    "AutoMinorVersionUpgrade": true,
    "SnapshotRetentionLimit": 1,
    "SnapshotWindow": "06:30-07:30",
    "NodeSnapshots": [
      {
        "CacheNodeId": "0001",
        "CacheSize": "5 MB",
        "CacheNodeCreateTime": "2019-11-26T01:22:52.396Z",
        "SnapshotCreateTime": "2019-12-05T06:38:23Z"
      }
    ]
  },
  {
    "SnapshotName": "myreplica-backup",
    "CacheClusterId": "myreplica",
    "SnapshotStatus": "available",
    "SnapshotSource": "manual",
    "CacheNodeType": "cache.r5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.5",
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-west-2a",
    "CacheClusterCreateTime": "2019-11-26T00:14:52.439Z",
    "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
    "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxx152:My_Topic",
    "Port": 6379,
```

```
"CacheParameterGroupName": "default.redis5.0",
"CacheSubnetGroupName": "kxkxk",
"VpcId": "vpc-a3e97cdb",
"AutoMinorVersionUpgrade": true,
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "09:00-10:00",
"NodeSnapshots": [
  {
    "CacheNodeId": "0001",
    "CacheSize": "5 MB",
    "CacheNodeCreateTime": "2019-11-26T00:14:52.439Z",
    "SnapshotCreateTime": "2019-11-26T00:25:01Z"
  }
]
},
{
  "SnapshotName": "my-cluster",
  "CacheClusterId": "my-cluster-003",
  "SnapshotStatus": "available",
  "SnapshotSource": "manual",
  "CacheNodeType": "cache.r5.large",
  "Engine": "redis",
  "EngineVersion": "5.0.5",
  "NumCacheNodes": 1,
  "PreferredAvailabilityZone": "us-west-2a",
  "CacheClusterCreateTime": "2019-11-25T23:56:17.186Z",
  "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
  "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
  "Port": 6379,
  "CacheParameterGroupName": "default.redis5.0",
  "CacheSubnetGroupName": "kxkxk",
  "VpcId": "vpc-a3e97cdb",
  "AutoMinorVersionUpgrade": true,
  "SnapshotRetentionLimit": 0,
  "SnapshotWindow": "09:00-10:00",
  "NodeSnapshots": [
    {
      "CacheNodeId": "0001",
      "CacheSize": "5 MB",
      "CacheNodeCreateTime": "2019-11-25T23:56:17.186Z",
      "SnapshotCreateTime": "2019-11-26T03:08:33Z"
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Backup and Restore ElastiCache für Redis](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeSnapshots AWS CLI Befehlsreferenz](#).

describe-update-actions

Das folgende Codebeispiel zeigt die Verwendung `describe-update-actions`.

AWS CLI

Um Aktualisierungsaktionen zu beschreiben

Das folgende `describe-update-actions` Beispiel gibt Details zu Aktualisierungsaktionen zurück.

```
aws elasticache describe-update-actions
```

Ausgabe:

```
{
  "UpdateActions": [
    {
      "ReplicationGroupId": "mycluster",
      "ServiceUpdateName": "elc-20191007-001",
      "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
      "ServiceUpdateSeverity": "important",
      "ServiceUpdateStatus": "available",
      "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
      "ServiceUpdateType": "security-update",
      "UpdateActionAvailableDate": "2019-12-05T19:15:19.995Z",
      "UpdateActionStatus": "complete",
      "NodesUpdated": "9/9",
      "UpdateActionStatusModifiedDate": "2019-12-05T19:15:20.461Z",
      "SlaMet": "n/a",
      "Engine": "redis"
    },
    {
      "CacheClusterId": "my-memcached-cluster",
      "ServiceUpdateName": "elc-20191007-001",
```

```
    "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
    "ServiceUpdateSeverity": "important",
    "ServiceUpdateStatus": "available",
    "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
    "ServiceUpdateType": "security-update",
    "UpdateActionAvailableDate": "2019-12-04T18:26:05.349Z",
    "UpdateActionStatus": "complete",
    "NodesUpdated": "1/1",
    "UpdateActionStatusModifiedDate": "2019-12-04T18:26:05.352Z",
    "SlaMet": "n/a",
    "Engine": "redis"
  },
  {
    "ReplicationGroupId": "my-cluster",
    "ServiceUpdateName": "elc-20191007-001",
    "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
    "ServiceUpdateSeverity": "important",
    "ServiceUpdateStatus": "available",
    "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
    "ServiceUpdateType": "security-update",
    "UpdateActionAvailableDate": "2019-11-26T03:36:26.320Z",
    "UpdateActionStatus": "complete",
    "NodesUpdated": "4/4",
    "UpdateActionStatusModifiedDate": "2019-12-04T22:11:12.664Z",
    "SlaMet": "n/a",
    "Engine": "redis"
  },
  {
    "ReplicationGroupId": "my-cluster2",
    "ServiceUpdateName": "elc-20191007-001",
    "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
    "ServiceUpdateSeverity": "important",
    "ServiceUpdateStatus": "available",
    "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
    "ServiceUpdateType": "security-update",
    "UpdateActionAvailableDate": "2019-11-26T01:26:01.617Z",
    "UpdateActionStatus": "complete",
    "NodesUpdated": "3/3",
    "UpdateActionStatusModifiedDate": "2019-11-26T01:26:01.753Z",
    "SlaMet": "n/a",
    "Engine": "redis"
  }
]
```



```
}
```

Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache im Elasticache-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DescribeUpdateActions](#) in der AWS CLI Befehlsreferenz.

describe-user-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-user-groups`.

AWS CLI

Um Benutzergruppen zu beschreiben

Das folgende `describe-user-groups` Beispiel gibt eine Liste von Benutzergruppen zurück.

```
aws elasticache describe-user-groups
```

Ausgabe:

```
{
  "UserGroups": [
    {
      "UserGroupId": "myusergroup",
      "Status": "active",
      "Engine": "redis",
      "UserIds": [
        "default"
      ],
      "ReplicationGroups": [],
      "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [DescribeUserGroups](#).AWS CLI

describe-users

Das folgende Codebeispiel zeigt die Verwendung `describe-users`.

AWS CLI

Um Benutzer zu beschreiben

Das folgende `describe-users` Beispiel gibt eine Liste von Benutzern zurück.

```
aws elasticache describe-users
```

Ausgabe:

```
{
  "Users": [
    {
      "UserId": "default",
      "UserName": "default",
      "Status": "active",
      "Engine": "redis",
      "AccessString": "on ~* +@all",
      "UserGroupIds": [
        "myusergroup"
      ],
      "Authentication": {
        "Type": "no-password"
      },
      "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:default"
    },
    {
      "UserId": "user1",
      "UserName": "myUser",
      "Status": "active",
      "Engine": "redis",
      "AccessString": "on ~* +@all",
      "UserGroupIds": [],
      "Authentication": {
        "Type": "password",
        "PasswordCount": 1
      },
      "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
    }
  ],
}
```

```

    {
      "UserId": "user2",
      "UserName": "myUser",
      "Status": "active",
      "Engine": "redis",
      "AccessString": "on ~app:* -@all +@read +@hash +@bitmap +@geo -setbit -
bitfield -hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius
-georadiusbymember",
      "UserGroupIds": [],
      "Authentication": {
        "Type": "password",
        "PasswordCount": 1
      },
      "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [DescribeUsers](#).AWS CLI

disassociate-global-replication-group

Das folgende Codebeispiel zeigt die Verwendung `disassociate-global-replication-group`.

AWS CLI

Um einen sekundären Cluster von einer globalen Replikationsgruppe zu trennen

Im folgenden `disassociate-global-replication-group` Beispiel wird ein sekundärer Cluster aus einem globalen Datenspeicher entfernt

```

aws elasticache disassociate-global-replication-group \
  --global-replication-group-id my-grg \
  --replication-group-id my-cluster-grg-secondary \
  --replication-group-region us-east-1

```

Ausgabe:

```

{
  "GlobalReplicationGroup": {

```

```

    "GlobalReplicationGroupId": "my-grg",
    "GlobalReplicationGroupDescription": "my-grg",
    "Status": "modifying",
    "CacheNodeType": "cache.r5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.6",
    "Members": [
      {
        "ReplicationGroupId": "my-cluster-grg-secondary",
        "ReplicationGroupRegion": "us-east-1",
        "Role": "SECONDARY",
        "AutomaticFailover": "enabled",
        "Status": "associated"
      },
      {
        "ReplicationGroupId": "my-cluster-grg",
        "ReplicationGroupRegion": "us-west-2",
        "Role": "PRIMARY",
        "AutomaticFailover": "enabled",
        "Status": "associated"
      }
    ],
    "ClusterEnabled": false,
    "AuthTokenEnabled": false,
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}

```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DisassociateGlobalReplicationGroup](#).AWS CLI

increase-node-groups-in-global-replication-group

Das folgende Codebeispiel zeigt die Verwendung `increase-node-groups-in-global-replication-group`.

AWS CLI

Um die Anzahl der Knotengruppen in einer globalen Replikationsgruppe zu erhöhen

Im Folgenden wird die Anzahl der Knotengruppen mithilfe der Redis-Engine `increase-node-groups-in-global-replication-group` erhöht.

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --global-replication-group-id sgai-pat-test-4 \  
  --node-group-count 6 \  
  --apply-immediately
```

Ausgabe:

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "sgai-test-4",  
    "GlobalReplicationGroupDescription": "test-4",  
    "Status": "modifying",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "my-cluster-b",  
        "ReplicationGroupRegion": "us-east-1",  
        "Role": "SECONDARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      },  
      {  
        "ReplicationGroupId": "my-cluster-a",  
        "ReplicationGroupRegion": "us-west-2",  
        "Role": "PRIMARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      }  
    ],  
    "ClusterEnabled": true,  
    "GlobalNodeGroups": [  
      {  
        "GlobalNodeGroupId": "sgai-test-4-0001",  
        "Slots": "0-234,2420-5461"  
      },  
      {  
        "GlobalNodeGroupId": "sgai-test-4-0002",  
        "Slots": "5462-5904,6997-9830"  
      }  
    ]  
  }  
}
```

```
    },
    {
      "GlobalNodeId": "sgaii-test-4-0003",
      "Slots": "10923-11190,13375-16383"
    },
    {
      "GlobalNodeId": "sgaii-test-4-0004",
      "Slots": "235-2419,5905-6996"
    },
    {
      "GlobalNodeId": "sgaii-test-4-0005",
      "Slots": "9831-10922,11191-13374"
    }
  ],
  "AuthTokenEnabled": false,
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [IncreaseNodeGroupsInGlobalReplicationGroup](#).AWS CLI

increase-replica-count

Das folgende Codebeispiel zeigt die Verwendung `increase-replica-count`.

AWS CLI

Um die Anzahl der Replikate zu erhöhen

Das folgende `increase-replica-count` Beispiel macht eines von zwei Dingen. Es kann die Anzahl der Replikate in einer Redis-Replikationsgruppe (Clustermodus deaktiviert) dynamisch erhöhen. Oder es kann die Anzahl der Replikatknoten in einer oder mehreren Knotengruppen (Shards) einer Redis-Replikationsgruppe (Clustermodus aktiviert) dynamisch erhöhen. Dieser Vorgang wird ohne Cluster-Ausfallzeiten ausgeführt.

```
aws elasticache increase-replica-count \
  --replication-group-id "my-cluster" \
```

```
--apply-immediately \  
--new-replica-count 3
```

Ausgabe:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "my-cluster",  
    "Description": " ",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "my-cluster-001",  
      "my-cluster-002",  
      "my-cluster-003",  
      "my-cluster-004"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "my-  
cluster.xxxxxih.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "my-cluster-  
ro.xxxxxxih.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "NodeGroupMembers": [  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0001.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "primary"  
          },  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0002",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0002.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "secondary"  
          },  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0003",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0003.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "secondary"  
          },  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0004",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0004.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "secondary"  
          }  
        ],  
        "PreferredAvailabilityZone": "us-west-2a",  
        "CurrentRole": "primary"  
      },  
      {  
        "NodeGroupId": "0002",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "my-  
cluster.xxxxxih.ng.0002.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "my-cluster-  
ro.xxxxxxih.ng.0002.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "NodeGroupMembers": [  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0005",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0005.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "primary"  
          },  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0006",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0006.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "secondary"  
          }  
        ],  
        "PreferredAvailabilityZone": "us-west-2a",  
        "CurrentRole": "primary"  
      },  
      {  
        "NodeGroupId": "0003",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "my-  
cluster.xxxxxih.ng.0003.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "my-cluster-  
ro.xxxxxxih.ng.0003.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "NodeGroupMembers": [  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0007",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0007.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "primary"  
          },  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0008",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0008.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "secondary"  
          }  
        ],  
        "PreferredAvailabilityZone": "us-west-2a",  
        "CurrentRole": "primary"  
      },  
      {  
        "NodeGroupId": "0004",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "my-  
cluster.xxxxxih.ng.0004.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "my-cluster-  
ro.xxxxxxih.ng.0004.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "NodeGroupMembers": [  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0009",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0009.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "primary"  
          },  
          {  
            "CacheClusterId": "my-cluster-001",  
            "CacheNodeId": "0010",  
            "ReadEndpoint": {  
              "Address": "my-  
cluster-001.xxxxxih.0010.usw2.cache.amazonaws.com",  
              "Port": 6379  
            },  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CurrentRole": "secondary"  
          }  
        ],  
        "PreferredAvailabilityZone": "us-west-2a",  
        "CurrentRole": "primary"  
      }  
    ],  
    "PreferredAvailabilityZone": "us-west-2a",  
    "CurrentRole": "primary"  
  }  
}
```

```

        {
            "CacheClusterId": "my-cluster-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "my-
cluster-003.xxxxxih.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        }
    ]
}
],
"AutomaticFailover": "disabled",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.xlarge",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}

```

Weitere Informationen finden Sie unter [Erhöhung der Anzahl von Replikaten in einem Shard](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [IncreaseReplicaCount](#) in AWS CLI der Befehlsreferenz.

list-allowed-node-type-modifications

Das folgende Codebeispiel zeigt die Verwendung `list-allowed-node-type-modifications`.

AWS CLI

Um die zulässigen Knotenänderungen aufzulisten

Das folgende `list-allowed-node-type-modifications` Beispiel listet alle verfügbaren Knotentypen auf, auf die Sie den aktuellen Knotentyp Ihres Redis-Clusters oder Ihrer Replikationsgruppe skalieren können.

```
aws elasticache list-allowed-node-type-modifications \
```



```
--replication-group-id "my-replication-group"
```

Ausgabe:

```
{
  "ScaleUpModifications": [
    "cache.m5.12xlarge",
    "cache.m5.24xlarge",
    "cache.m5.4xlarge",
    "cache.r5.12xlarge",
    "cache.r5.24xlarge",
    "cache.r5.2xlarge",
    "cache.r5.4xlarge"
  ],
  "ScaleDownModifications": [
    "cache.m3.large",
    "cache.m3.medium",
    "cache.m3.xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.m5.2xlarge",
    "cache.m5.large",
    "cache.m5.xlarge",
    "cache.r3.large",
    "cache.r4.large",
    "cache.r4.xlarge",
    "cache.r5.large",
    "cache.t2.medium",
    "cache.t2.micro",
    "cache.t2.small",
    "cache.t3.medium",
    "cache.t3.micro",
    "cache.t3.small"
  ]
}
```

Weitere Informationen finden Sie unter [Skalierung ElastiCache für Redis-Cluster](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAllowedNodeTypeModifications](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet Tags für eine Ressource auf.

```
aws elasticache list-tags-for-resource \
  --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

Ausgabe:

```
{
  "TagList": [
    {
      "Key": "Project",
      "Value": "querySpeedUp"
    },
    {
      "Key": "Environment",
      "Value": "PROD"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Auflisten von Tags mithilfe von AWS CLI im Elasticache-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

modify-cache-cluster

Das folgende Codebeispiel zeigt die Verwendung `modify-cache-cluster`.

AWS CLI

Um Cache-Cluster zu ändern

Im folgenden `modify-cache-cluster` Beispiel werden die Einstellungen für den angegebenen Cluster geändert.

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id "my-cluster" \  
  --num-cache-nodes 1
```

Ausgabe:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "my-cluster",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheNodeType": "cache.m5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "CacheClusterStatus": "available",  
    "NumCacheNodes": 1,  
    "PreferredAvailabilityZone": "us-west-2c",  
    "CacheClusterCreateTime": "2019-12-04T18:24:56.652Z",  
    "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",  
    "PendingModifiedValues": {},  
    "CacheSecurityGroups": [],  
    "CacheParameterGroup": {  
      "CacheParameterGroupName": "default.redis5.0",  
      "ParameterApplyStatus": "in-sync",  
      "CacheNodeIdsToReboot": []  
    },  
    "CacheSubnetGroupName": "default",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "07:00-08:00",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

Weitere Informationen finden Sie unter [Ändern eines ElastiCache Clusters](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyCacheCluster](#) in der AWS CLI Befehlsreferenz.

modify-cache-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `modify-cache-parameter-group`.

AWS CLI

Um eine Cache-Parametergruppe zu ändern

Im folgenden `modify-cache-parameter-group` Beispiel werden die Parameter der angegebenen Cache-Parametergruppe geändert.

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

Ausgabe:

```
{  
  "CacheParameterGroupName": "mygroup"  
}
```

Weitere Informationen finden Sie unter [Ändern einer Parametergruppe](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyCacheParameterGroup](#) in der AWS CLI Befehlsreferenz.

modify-cache-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `modify-cache-subnet-group`.

AWS CLI

Um eine Cache-Subnetzgruppe zu ändern

Im folgenden `modify-cache-subnet-group` Beispiel wird die angegebene Cache-Subnetzgruppe geändert.

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name kxkxk \  
  --cache-subnet-group-description "mygroup"
```

Ausgabe:

```
{
  "CacheSubnetGroup": {
    "CacheSubnetGroupName": "kxkxk",
    "CacheSubnetGroupDescription": "mygroup",
    "VpcId": "vpc-xxxxcdb",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-xxxxbff",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Ändern einer Subnetzgruppe](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ModifyCacheSubnetGroup AWS CLI Befehlsreferenz](#).

modify-global-replication-group

Das folgende Codebeispiel zeigt die Verwendung `modify-global-replication-group`.

AWS CLI

Um eine globale Replikationsgruppe zu ändern

Im Folgenden werden die `modify-global-replication-group` Eigenschaften einer globalen Replikationsgruppe geändert, wobei in diesem Fall das automatische Failover mithilfe der Redis-Engine deaktiviert wird.

```
aws elasticache modify-global-replication-group \
  --global-replication-group-id sgaui-pat-group \
  --apply-immediately \
  --no-automatic-failover-enabled
```

Output

```
{
  "GlobalReplicationGroup": {
```

```
"GlobalReplicationGroupId": "sgau-test-group",
"GlobalReplicationGroupDescription": " ",
"Status": "modifying",
"CacheNodeType": "cache.r5.large",
"Engine": "redis",
"EngineVersion": "5.0.6",
"ClusterEnabled": false,
"AuthTokenEnabled": false,
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}
```

Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe von Global Datastore](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [ModifyGlobalReplicationGroup](#).AWS CLI

modify-replication-group-shard-configuration

Das folgende Codebeispiel zeigt die Verwendung `modify-replication-group-shard-configuration`.

AWS CLI

Um die Shard-Konfiguration einer Replikationsgruppe zu ändern

Im Folgenden wird die Anzahl der Knotengruppen mithilfe der Redis-Engine `modify-replication-group-shard-configuration` verringert.

```
aws elasticache modify-replication-group-shard-configuration \
  --replication-group-id mycluster \
  --node-group-count 3 \
  --apply-immediately \
  --node-groups-to-remove 0002
```

Output

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "mycluster",
    "Description": "mycluster",
    "GlobalReplicationGroupInfo": {},
```

```
"Status": "modifying",
"PendingModifiedValues": {},
"MemberClusters": [
  "mycluster-0002-001",
  "mycluster-0002-002",
  "mycluster-0002-003",
  "mycluster-0003-001",
  "mycluster-0003-002",
  "mycluster-0003-003",
  "mycluster-0003-004",
  "mycluster-0004-001",
  "mycluster-0004-002",
  "mycluster-0004-003",
  "mycluster-0005-001",
  "mycluster-0005-002",
  "mycluster-0005-003"
],
"NodeGroups": [
  {
    "NodeGroupId": "0002",
    "Status": "modifying",
    "Slots": "894-1767,3134-4443,5149-5461,6827-7332,12570-13662",
    "NodeGroupMembers": [
      {
        "CacheClusterId": "mycluster-0002-001",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2c"
      },
      {
        "CacheClusterId": "mycluster-0002-002",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2a"
      },
      {
        "CacheClusterId": "mycluster-0002-003",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2b"
      }
    ]
  },
  {
    "NodeGroupId": "0003",
    "Status": "modifying",
```

```
    "Slots":
      "0-324,5462-5692,6784-6826,7698-8191,10923-11075,12441-12569,13663-16383",
      "NodeGroupMembers": [
        {
          "CacheClusterId": "mycluster-0003-001",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2c"
        },
        {
          "CacheClusterId": "mycluster-0003-002",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2b"
        },
        {
          "CacheClusterId": "mycluster-0003-003",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2a"
        },
        {
          "CacheClusterId": "mycluster-0003-004",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2c"
        }
      ]
    },
    {
      "NodeGroupId": "0004",
      "Status": "modifying",
      "Slots": "325-336,4706-5148,7333-7697,9012-10922,11076-12440",
      "NodeGroupMembers": [
        {
          "CacheClusterId": "mycluster-0004-001",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2b"
        },
        {
          "CacheClusterId": "mycluster-0004-002",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2a"
        },
        {
          "CacheClusterId": "mycluster-0004-003",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2c"
        }
      ]
    }
  ]
}
```



```

    }
  ]
},
{
  "NodeGroupId": "0005",
  "Status": "modifying",
  "Slots": "337-893,1768-3133,4444-4705,5693-6783,8192-9011",
  "NodeGroupMembers": [
    {
      "CacheClusterId": "mycluster-0005-001",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2a"
    },
    {
      "CacheClusterId": "mycluster-0005-002",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2c"
    },
    {
      "CacheClusterId": "mycluster-0005-003",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2b"
    }
  ]
}
],
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"ConfigurationEndpoint": {
  "Address": "mycluster.g2xbih.clustercfg.usw2.cache.amazonaws.com",
  "Port": 6379
},
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "13:00-14:00",
"ClusterEnabled": true,
"CacheNodeType": "cache.r5.xlarge",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}

```

Weitere Informationen finden Sie unter [Skalierung ElastiCache für Redis-Cluster](#) im ElastiCache-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyReplicationGroupShardConfiguration](#) in der AWS CLI Befehlsreferenz.

modify-replication-group

Das folgende Codebeispiel zeigt die Verwendung `modify-replication-group`.

AWS CLI

Um eine Replikationsgruppe zu ändern

Im Folgenden wird Multi-AZ mithilfe der `modify-replication-group` Redis-Engine deaktiviert.

```
aws elasticache modify-replication-group \  
  --replication-group-id test-cluster \  
  --no-multi-az-enabled \  
  --apply-immediately
```

Output

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "test-cluster",  
    "Description": "test-cluster",  
    "GlobalReplicationGroupInfo": {  
      "GlobalReplicationGroupId": "sgaui-pat-group",  
      "GlobalReplicationGroupMemberRole": "PRIMARY"  
    },  
    "Status": "available",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "test-cluster-001",  
      "test-cluster-002",  
      "test-cluster-003"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "available",  
        "PrimaryEndpoint": {  
          "Address": "test-  
cluster.g2xbih.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379
```

```
    },
    "ReaderEndpoint": {
      "Address": "test-cluster-
ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "NodeGroupMembers": [
      {
        "CacheClusterId": "test-cluster-001",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "test-
cluster-001.g2xbih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "primary"
      },
      {
        "CacheClusterId": "test-cluster-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "test-
cluster-002.g2xbih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
      },
      {
        "CacheClusterId": "test-cluster-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "test-
cluster-003.g2xbih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "replica"
      }
    ]
  },
  "SnapshottingClusterId": "test-cluster-002",
```

```
    "AutomaticFailover": "enabled",
    "MultiAZ": "disabled",
    "SnapshotRetentionLimit": 1,
    "SnapshotWindow": "08:00-09:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r5.large",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyReplicationGroup](#) in der AWS CLI Befehlsreferenz.

modify-user-group

Das folgende Codebeispiel zeigt die Verwendung `modify-user-group`.

AWS CLI

Um eine Benutzergruppe zu ändern

Im folgenden `modify-user-group` Beispiel wird der Benutzergruppe ein Benutzer hinzugefügt.

```
aws elasticache modify-user-group \
  --user-group-id myusergroup \
  --user-ids-to-add user1
```

Ausgabe:

```
{
  "UserGroupId": "myusergroup",
  "Status": "modifying",
  "Engine": "redis",
  "UserIds": [
    "default"
  ],
  "PendingChanges": {
    "UserIdsToAdd": [
      "user1"
    ]
  }
}
```

```

    },
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
  }

```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [ModifyUserGroup](#).AWS CLI

modify-user

Das folgende Codebeispiel zeigt die Verwendung `modify-user`.

AWS CLI

Um einen Benutzer zu ändern

Im folgenden `modify-user` Beispiel wird die Zugriffszeichenfolge eines Benutzers geändert.

```

aws elasticache modify-user \
  --user-id user2 \
  --append-access-string "on ~* +@all"

```

Ausgabe:

```

{
  "UserId": "user2",
  "UserName": "myUser",
  "Status": "modifying",
  "Engine": "redis",
  "AccessString": "on ~* +@all",
  "UserGroupIds": [],
  "Authentication": {
    "Type": "password",
    "PasswordCount": 1
  },
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
}

```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [ModifyUser](#).AWS CLI

purchase-reserved-cache-nodes-offering

Das folgende Codebeispiel zeigt die Verwendung `purchase-reserved-cache-nodes-offering`.

AWS CLI

Um eine zu kaufen `reserved-cache-node-offering`

Im folgenden `purchase-reserved-cache-nodes-offering` Beispiel können Sie ein Angebot mit reservierten Cache-Knoten erwerben.

```
aws elasticache purchase-reserved-cache-nodes-offering \  
--reserved-cache-nodes-offering-id xxxxxxxx-4da5-4b90-b92d-929fbd7abed2
```

Output

```
{  
  "ReservedCacheNode": {  
    "ReservedCacheNodeId": "ri-2020-06-30-17-59-40-474",  
    "ReservedCacheNodesOfferingId": "xxxxxxxx-4da5-4b90-b92d-929fbd7abed2",  
    "CacheNodeType": "cache.m3.2xlarge",  
    "StartTime": "2020-06-30T17:59:40.474000+00:00",  
    "Duration": 31536000,  
    "FixedPrice": 1772.0,  
    "UsagePrice": 0.0,  
    "CacheNodeCount": 1,  
    "ProductDescription": "redis",  
    "OfferingType": "Heavy Utilization",  
    "State": "payment-pending",  
    "RecurringCharges": [  
      {  
        "RecurringChargeAmount": 0.25,  
        "RecurringChargeFrequency": "Hourly"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Informationen zu Reserved Node-Angeboten](#) im Elasticache Redis-Benutzerhandbuch oder [Informationen zu Reserved Node-Angeboten](#) im Elasticache Memcached User Guide.

- API-Einheiten finden Sie unter Befehlsreferenz. [PurchaseReservedCacheNodesOffering](#) AWS CLI

reboot-cache-cluster

Das folgende Codebeispiel zeigt die Verwendung `reboot-cache-cluster`.

AWS CLI

Um einen Cache-Cluster neu zu starten

Im folgenden `reboot-cache-cluster` Beispiel werden einige oder alle Cache-Knoten in einem bereitgestellten Cluster neu gestartet. Dieser Vorgang wendet alle geänderten Cache-Parametergruppen auf den Cluster an. Der Neustartvorgang findet so schnell wie möglich statt und führt zu einem kurzzeitigen Ausfall des Clusters. Während des Neustarts wird der Clusterstatus auf `REBOOTING` gesetzt.

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id "my-cluster-001" \  
  --cache-node-ids-to-reboot "0001"
```

Ausgabe:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "my-cluster-001",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheNodeType": "cache.r5.xlarge",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "CacheClusterStatus": "rebooting cache cluster nodes",  
    "NumCacheNodes": 1,  
    "PreferredAvailabilityZone": "us-west-2a",  
    "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",  
    "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",  
    "PendingModifiedValues": {},  
  }  
}
```

```
"NotificationConfiguration": {
  "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
  "TopicStatus": "active"
},
"CacheSecurityGroups": [],
"CacheParameterGroup": {
  "CacheParameterGroupName": "mygroup",
  "ParameterApplyStatus": "in-sync",
  "CacheNodeIdsToReboot": []
},
"CacheSubnetGroupName": "kxkxk",
"AutoMinorVersionUpgrade": true,
"SecurityGroups": [
  {
    "SecurityGroupId": "sg-xxxxxxxxxxxx836",
    "Status": "active"
  },
  {
    "SecurityGroupId": "sg-xxxxxxx7b",
    "Status": "active"
  }
],
"ReplicationGroupId": "my-cluster",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}
```

Weitere Informationen finden Sie unter [Einen Cluster neu starten < https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters_rebooting.html](https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters_rebooting.html) im Elasticache-Benutzerhandbuch.

- Einzelheiten API finden Sie in der Befehlsreferenz. [RebootCacheCluster](#) AWS CLI

reset-cache-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `reset-cache-parameter-group`.

AWS CLI

Um eine Cache-Parametergruppe zurückzusetzen

Im folgenden `reset-cache-parameter-group` Beispiel werden die Parameter einer Cache-Parametergruppe auf den Engine- oder Systemstandardwert geändert. Sie können bestimmte Parameter zurücksetzen, indem Sie eine Liste mit Parameternamen einreichen. Um die gesamte Cache-Parametergruppe zurückzusetzen, geben Sie die `--cache-parameter-group-name` Parameter `--reset-all-parameters` und an.

```
aws elasticache reset-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --reset-all-parameters
```

Ausgabe:

```
{  
  "CacheParameterGroupName": "mygroup"  
}
```

- APIEinzelheiten finden Sie [ResetCacheParameterGroup](#) in der AWS CLI Befehlsreferenz.

start-migration

Das folgende Codebeispiel zeigt die Verwendung `start-migration`.

AWS CLI

Um eine Migration zu starten

Im Folgenden werden Ihre Daten mithilfe der Redis-Engine von selbst gehostetem Redis auf Amazon EC2 zu Amazon `start-migration` ElastiCache migriert.

```
aws elasticache start-migration \  
  --replication-group-id test \  
  --customer-node-endpoint-  
  list "Address='test.g2xbih.ng.0001.usw2.cache.amazonaws.com',Port=6379"
```

Output

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "test",  
    "Description": "test",  
    "GlobalReplicationGroupInfo": {},  
  },  
}
```

```
"Status": "modifying",
"PendingModifiedValues": {},
"MemberClusters": [
  "test-001",
  "test-002",
  "test-003"
],
"NodeGroups": [
  {
    "NodeGroupId": "0001",
    "Status": "available",
    "PrimaryEndpoint": {
      "Address": "test.g2xbih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "ReaderEndpoint": {
      "Address": "test-ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "NodeGroupMembers": [
      {
        "CacheClusterId": "test-001",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address":
"test-001.g2xbih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "primary"
      },
      {
        "CacheClusterId": "test-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address":
"test-002.g2xbih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2c",
        "CurrentRole": "replica"
      },
      {
        "CacheClusterId": "test-003",
```

```

        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address":
"test-003.g2xbih.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    }
]
}
],
"SnapshottingClusterId": "test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}

```

Weitere Informationen finden Sie unter [Online-Migration zu ElastiCache](#) im Elasticache-Benutzerhandbuch.

- API-Einheiten finden Sie [StartMigration](#) in der AWS CLI Befehlsreferenz.

test-failover

Das folgende Codebeispiel zeigt die Verwendung `test-failover`.

AWS CLI

Um das Failover einer Knotengruppe zu testen

Im folgenden `test-failover` Beispiel wird das automatische Failover für die angegebene Knotengruppe (in der Konsole als `Shard` bezeichnet) in einer Replikationsgruppe (in der Konsole als `Cluster` bezeichnet) getestet.

```

aws elasticache test-failover /
  --replication-group-id "mycluster" /

```

```
--node-group-id "0001"
```

Ausgabe:

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "mycluster",
    "Description": "My Cluster",
    "Status": "available",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "mycluster-0001-001",
      "mycluster-0001-002",
      "mycluster-0001-003",
      "mycluster-0002-001",
      "mycluster-0002-002",
      "mycluster-0002-003",
      "mycluster-0003-001",
      "mycluster-0003-002",
      "mycluster-0003-003"
    ],
    "NodeGroups": [
      {
        "NodeGroupId": "0001",
        "Status": "available",
        "Slots": "0-5461",
        "NodeGroupMembers": [
          {
            "CacheClusterId": "mycluster-0001-001",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2b"
          },
          {
            "CacheClusterId": "mycluster-0001-002",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2a"
          },
          {
            "CacheClusterId": "mycluster-0001-003",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2c"
          }
        ]
      }
    ]
  }
}
```

```
    },
    {
      "NodeGroupId": "0002",
      "Status": "available",
      "Slots": "5462-10922",
      "NodeGroupMembers": [
        {
          "CacheClusterId": "mycluster-0002-001",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2a"
        },
        {
          "CacheClusterId": "mycluster-0002-002",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2b"
        },
        {
          "CacheClusterId": "mycluster-0002-003",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2c"
        }
      ]
    },
    {
      "NodeGroupId": "0003",
      "Status": "available",
      "Slots": "10923-16383",
      "NodeGroupMembers": [
        {
          "CacheClusterId": "mycluster-0003-001",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2c"
        },
        {
          "CacheClusterId": "mycluster-0003-002",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2b"
        },
        {
          "CacheClusterId": "mycluster-0003-003",
          "CacheNodeId": "0001",
          "PreferredAvailabilityZone": "us-west-2a"
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "AutomaticFailover": "enabled",
  "ConfigurationEndpoint": {
    "Address": "mycluster.xxxxih.clustercfg.usw2.cache.amazonaws.com",
    "Port": 6379
  },
  "SnapshotRetentionLimit": 1,
  "SnapshotWindow": "13:00-14:00",
  "ClusterEnabled": true,
  "CacheNodeType": "cache.r5.large",
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
}
```

- API-Einheiten finden Sie [TestFailover](#) in der AWS CLI Befehlsreferenz.

MediaStore Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaStore.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-container

Das folgende Codebeispiel zeigt die Verwendung `create-container`.

AWS CLI

Um einen Container zu erstellen

Im folgenden `create-container` Beispiel wird ein neuer, leerer Container erstellt.

```
aws mediastore create-container --container-name ExampleContainer
```

Ausgabe:

```
{
  "Container": {
    "AccessLoggingEnabled": false,
    "CreationTime": 1563557265,
    "Name": "ExampleContainer",
    "Status": "CREATING",
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen eines Containers](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [CreateContainer](#) in der AWS CLI Befehlsreferenz.

delete-container-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-container-policy`.

AWS CLI

Um eine Container-Richtlinie zu löschen

Im folgenden `delete-container-policy` Beispiel wird die Richtlinie gelöscht, die dem angegebenen Container zugewiesen ist. Wenn die Richtlinie gelöscht wird, weist AWS Elemental dem Container MediaStore automatisch die Standardrichtlinie zu.

```
aws mediastore delete-container-policy \
  --container-name LiveEvents
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteContainerPolicy](#) in der AWS MediaStore APIElemental-Referenz.

- APIEinzelheiten finden Sie [DeleteContainerPolicy](#) in der AWS CLI Befehlsreferenz.

delete-container

Das folgende Codebeispiel zeigt die Verwendung `delete-container`.

AWS CLI

Um einen Container zu löschen

Im folgenden `delete-container` Beispiel wird der angegebene Container gelöscht. Sie können einen Container nur löschen, wenn er keine Objekte enthält.

```
aws mediastore delete-container \  
  --container-name=ExampleLiveDemo
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Containers](#) im AWS Elemental MediaStore User Guide.

- APIEinzelheiten finden Sie [DeleteContainer](#) in der AWS CLI Befehlsreferenz.

delete-cors-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-cors-policy`.

AWS CLI

Um eine CORS Richtlinie zu löschen

Im folgenden `delete-cors-policy` Beispiel wird die ursprungsübergreifende Resource Sharing (CORS) -Richtlinie gelöscht, die dem angegebenen Container zugewiesen ist.

```
aws mediastore delete-cors-policy \  
  --container-name ExampleContainer
```


Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer CORS Richtlinie](#) im AWS Elemental User Guide MediaStore .

- APIEinzelheiten finden Sie [DeleteCorsPolicy](#) in der AWS CLI Befehlsreferenz.

delete-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-lifecycle-policy`.

AWS CLI

Um eine Objektlebenszyklus-Richtlinie zu löschen

Im folgenden `delete-lifecycle-policy` Beispiel wird die Objektlebenszyklus-Richtlinie gelöscht, die an den angegebenen Container angehängt ist. Es kann bis zu 20 Minuten dauern, bis diese Änderung wirksam wird.

```
aws mediastore delete-lifecycle-policy \  
  --container-name LiveEvents
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Object Lifecycle-Richtlinie](#) im AWS Elemental MediaStore User Guide.

- APIEinzelheiten finden Sie [DeleteLifecyclePolicy](#) in der AWS CLI Befehlsreferenz.

describe-container

Das folgende Codebeispiel zeigt die Verwendung `describe-container`.

AWS CLI

Um die Details eines Containers anzuzeigen

Im folgenden `describe-container` Beispiel werden die Details des angegebenen Containers angezeigt.

```
aws mediastore describe-container \  
  --container-name ExampleContainer
```

Ausgabe:

```
{
  "Container": {
    "CreationTime": 1563558086,
    "AccessLoggingEnabled": false,
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
    "Status": "ACTIVE",
    "Name": "ExampleContainer",
    "Endpoint": "https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com"
  }
}
```

Weitere Informationen finden Sie unter [Anzeigen der Details für einen Container](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [DescribeContainer](#) in der AWS CLI Befehlsreferenz.

describe-object

Das folgende Codebeispiel zeigt die Verwendung `describe-object`.

AWS CLI

Um eine Liste von Objekten und Ordnern in einem bestimmten Container anzuzeigen

Im folgenden `describe-object` Beispiel werden Elemente (Objekte und Ordner) angezeigt, die in einem bestimmten Container gespeichert sind.

```
aws mediastore-data describe-object \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \
  --path /folder_name/file1234.jpg
```

Ausgabe:

```
{
  "ContentType": "image/jpeg",
  "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
  "ContentLength": "2307346",
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f5555555555555555da6d3"
```

```
}
```

Weitere Informationen finden Sie unter [Anzeigen der Details eines Objekts](#) im AWS MediaStore Elemental-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeObject](#) in der AWS CLI Befehlsreferenz.

get-container-policy

Das folgende Codebeispiel zeigt die Verwendung `get-container-policy`.

AWS CLI

Um eine Container-Richtlinie anzuzeigen

Das folgende `get-container-policy` Beispiel zeigt die ressourcenbasierte Richtlinie des angegebenen Containers.

```
aws mediastore get-container-policy \  
  --container-name ExampleLiveDemo
```

Ausgabe:

```
{  
  "Policy": {  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Sid": "PublicReadOverHttps",  
        "Effect": "Allow",  
        "Principal": {  
          "AWS": "arn:aws:iam::111122223333:root"  
        },  
        "Action": [  
          "mediastore:GetObject",  
          "mediastore:DescribeObject"  
        ],  
        "Resource": "arn:aws:mediastore:us-west-2:111122223333:container/  
ExampleLiveDemo/",  
        "Condition": {  
          "Bool": {  
            "aws:SecureTransport": "true"  
          }  
        }  
      }  
    ]  
  }  
}
```

```

    }
  }
]
}
}

```

Weitere Informationen finden Sie unter [Container-Richtlinie anzeigen](#) im AWS Elemental User Guide MediaStore .

- API-Einheiten finden Sie [GetContainerPolicy](#) in der AWS CLI Befehlsreferenz.

get-cors-policy

Das folgende Codebeispiel zeigt die Verwendung `get-cors-policy`.

AWS CLI

Um eine CORS Richtlinie anzuzeigen

Im folgenden `get-cors-policy` Beispiel wird die ursprungsübergreifende Richtlinie für die gemeinsame Nutzung von Ressourcen (CORS) angezeigt, die dem angegebenen Container zugewiesen ist.

```

aws mediastore get-cors-policy \
  --container-name ExampleContainer \
  --region us-west-2

```

Ausgabe:

```

{
  "CorsPolicy": [
    {
      "AllowedMethods": [
        "GET",
        "HEAD"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedOrigins": [
        ""
      ],
      "AllowedHeaders": [

```

```
        ""
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [CORSRichtlinie anzeigen](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [GetCorsPolicy](#) in der AWS CLI Befehlsreferenz.

get-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `get-lifecycle-policy`.

AWS CLI

Um eine Objektlebenszyklus-Richtlinie anzuzeigen

Das folgende `get-lifecycle-policy` Beispiel zeigt die Objektlebenszyklus-Richtlinie, die an den angegebenen Container angehängt ist.

```
aws mediastore get-lifecycle-policy \  
  --container-name LiveEvents
```

Ausgabe:

```
{  
  "LifecyclePolicy": {  
    "rules": [  
      {  
        "definition": {  
          "path": [  
            {  
              "prefix": "Football/"  
            },  
            {  
              "prefix": "Baseball/"  
            }  
          ],  
          "days_since_create": [  
            {
```



```
{
  "Containers": [
    {
      "CreationTime": 1505317931,
      "Endpoint": "https://aaabbbcccddee.data.mediastore.us-
west-2.amazonaws.com",
      "Status": "ACTIVE",
      "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
      "AccessLoggingEnabled": false,
      "Name": "ExampleLiveDemo"
    },
    {
      "CreationTime": 1506528818,
      "Endpoint": "https://fffggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
      "Status": "ACTIVE",
      "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
      "AccessLoggingEnabled": false,
      "Name": "ExampleContainer"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS Elemental MediaStore User Guide unter [Eine Liste von Containern anzeigen](#).

- API-Einheiten finden Sie [ListContainers](#) in der AWS CLI Befehlsreferenz.

list-items

Das folgende Codebeispiel zeigt die Verwendung `list-items`.

AWS CLI

Beispiel 1: Um eine Liste von Objekten und Ordnern in einem bestimmten Container anzuzeigen

Im folgenden `list-items` Beispiel werden Elemente (Objekte und Ordner) angezeigt, die im angegebenen Container gespeichert sind.

```
aws mediastore-data list-items \
```



```
--endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com
```

Ausgabe:

```
{
  "Items": [
    {
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379,
      "Name": "filename.jpg",
      "Type": "OBJECT",
      "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
      "ContentLength": 3784
    },
    {
      "Type": "FOLDER",
      "Name": "ExampleLiveDemo"
    }
  ]
}
```

Beispiel 2: Um eine Liste von Objekten und Ordnern in einem bestimmten Ordner anzuzeigen

Im folgenden `list-items` Beispiel werden Elemente (Objekte und Ordner) angezeigt, die in einem bestimmten Ordner gespeichert sind.

```
aws mediastore-data list-items \  
--endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com
```

Ausgabe:

```
{
  "Items": [
    {
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379,
      "Name": "filename.jpg",
      "Type": "OBJECT",
      "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
      "ContentLength": 3784
    }
  ]
}
```

```
    },
    {
      "Type": "FOLDER",
      "Name": "ExampleLiveDemo"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS Elemental MediaStore User Guide unter [Eine Objektliste anzeigen](#).

- API-Einheiten finden Sie [ListItems](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für einen Container aufzulisten

Im folgenden `list-tags-for-resource` Beispiel werden die Tag-Schlüssel und -Werte angezeigt, die dem angegebenen Container zugewiesen sind.

```
aws mediastore list-tags-for-resource \
  --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

Ausgabe:

```
{
  "Tags": [
    {
      "Value": "Test",
      "Key": "Environment"
    },
    {
      "Value": "West",
      "Key": "Region"
    }
  ]
}
```

Weitere Informationen finden Sie [ListTagsForResource](#) in der AWS Elemental MediaStore API Reference.

- API-Einheiten finden Sie unter [ListTagsForResource AWS CLI](#) Befehlsreferenz.

put-container-policy

Das folgende Codebeispiel zeigt die Verwendung `put-container-policy`.

AWS CLI

Um eine Container-Richtlinie zu bearbeiten

Im folgenden `put-container-policy` Beispiel wird dem angegebenen Container eine andere Richtlinie zugewiesen. In diesem Beispiel ist die aktualisierte Richtlinie in einer Datei mit dem Namen `LiveEventsContainerPolicy.json` definiert.

```
aws mediastore put-container-policy \  
  --container-name LiveEvents \  
  --policy file:///LiveEventsContainerPolicy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Bearbeiten einer Container-Richtlinie](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [PutContainerPolicy](#) in der AWS CLI Befehlsreferenz.

put-cors-policy

Das folgende Codebeispiel zeigt die Verwendung `put-cors-policy`.

AWS CLI

Beispiel 1: Um eine CORS Richtlinie hinzuzufügen

Das folgende `put-cors-policy` Beispiel fügt dem angegebenen Container eine ursprungsübergreifende Richtlinie für die gemeinsame Nutzung von Ressourcen (CORS) hinzu. Der Inhalt der CORS Richtlinie befindet sich in der Datei mit dem Namen `corsPolicy.json`.

```
aws mediastore put-cors-policy \  
  --policy file:///corsPolicy.json
```

```
--container-name ExampleContainer \  
--cors-policy file://corsPolicy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen einer CORS Richtlinie zu einem Container](#) im AWS Elemental MediaStore User Guide.

Beispiel 2: Um eine CORS Richtlinie zu bearbeiten

Im folgenden `put-cors-policy` Beispiel wird die ursprungsübergreifende Resource Sharing (CORS) -Richtlinie aktualisiert, die dem angegebenen Container zugewiesen ist. Der Inhalt der aktualisierten CORS Richtlinie befindet sich in der Datei mit dem Namen `corsPolicy2.json`.

Weitere Informationen finden Sie unter [Bearbeiten einer CORS Richtlinie](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [PutCorsPolicy](#) in der AWS CLI Befehlsreferenz.

put-lifecycle-policy

Das folgende Codebeispiel zeigt die Verwendung `put-lifecycle-policy`.

AWS CLI

Um eine Objektlebenszyklus-Richtlinie zu erstellen

Im folgenden `put-lifecycle-policy` Beispiel wird eine Objektlebenszyklus-Richtlinie an den angegebenen Container angehängt. Auf diese Weise können Sie angeben, wie lange der Service Objekte in Ihrem Container speichern soll. MediaStore löscht Objekte im Container, sobald sie ihr Ablaufdatum erreicht haben, wie in der Richtlinie angegeben, die sich in der genannten `LiveEventsLifecyclePolicy.json` Datei befindet.

```
aws mediastore put-lifecycle-policy \  
--container-name ExampleContainer \  
--lifecycle-policy file://ExampleLifecyclePolicy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen einer Object Lifecycle-Richtlinie zu einem Container](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie unter [PutLifecyclePolicy AWS CLI Befehlsreferenz](#).

put-object

Das folgende Codebeispiel zeigt die Verwendung `put-object`.

AWS CLI

Um ein Objekt hochzuladen

Im folgenden `put-object` Beispiel wird ein Objekt in den angegebenen Container hochgeladen. Sie können einen Ordnerpfad angeben, in dem das Objekt innerhalb des Containers gespeichert wird. Wenn der Ordner bereits existiert, speichert AWS Elemental das Objekt im Ordner. Wenn der Ordner nicht existiert, erstellt der Dienst ihn und speichert das Objekt dann in dem Ordner.

```
aws mediastore-data put-object \  
  --endpoint https://aaabbbcccddee.data.mediastore.us-west-2.amazonaws.com \  
  --body README.md \  
  --path /folder_name/README.md \  
  --cache-control "max-age=6, public" \  
  --content-type binary/octet-stream
```

Ausgabe:

```
{  
  "ContentSHA256":  
    "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",  
  "StorageClass": "TEMPORAL",  
  "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"  
}
```

Weitere Informationen finden Sie unter [Hochladen eines Objekts](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [PutObject](#) in der AWS CLI Befehlsreferenz.

start-access-logging

Das folgende Codebeispiel zeigt die Verwendung `start-access-logging`.

AWS CLI

Um die Zugriffsprotokollierung für einen Container zu aktivieren

Im folgenden `start-access-logging` Beispiel wird die Zugriffsprotokollierung für den angegebenen Container aktiviert.

```
aws mediastore start-access-logging \  
  --container-name LiveEvents
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren der Zugriffsprotokollierung für einen Container](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [StartAccessLogging](#) in der AWS CLI Befehlsreferenz.

stop-access-logging

Das folgende Codebeispiel zeigt die Verwendung `stop-access-logging`.

AWS CLI

Um die Zugriffsprotokollierung für einen Container zu deaktivieren

Im folgenden `stop-access-logging` Beispiel wird die Zugriffsprotokollierung für den angegebenen Container deaktiviert.

```
aws mediastore stop-access-logging \  
  --container-name LiveEvents
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deaktivieren der Zugriffsprotokollierung für einen Container](#) im AWS Elemental User Guide MediaStore .

- API-Einheiten finden Sie [StopAccessLogging](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags zu einem Container hinzuzufügen

Im folgenden `tag-resource` Beispiel werden dem angegebenen Container Tag-Schlüssel und -Werte hinzugefügt.

```
aws mediastore tag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value": "Test"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [TagResource](#) in der AWS Elemental MediaStore API Reference.

- API-Einheiten finden Sie unter [TagResource AWS CLI Befehlsreferenz](#).

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einem Container zu entfernen

Im folgenden `untag-resource` Beispiel werden der angegebene Tag-Schlüssel und der zugehörige Wert aus einem Container entfernt.

```
aws mediastore untag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tag-keys Region
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UntagResource](#) in der AWS Elemental MediaStore API Reference. .

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

EMR Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren EMR.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-instance-fleet

Das folgende Codebeispiel zeigt die Verwendung `add-instance-fleet`.

AWS CLI

Um einem Cluster eine Task-Instance-Flotte hinzuzufügen

In diesem Beispiel wird dem angegebenen Cluster eine neue Task-Instance-Flotte hinzugefügt.

Befehl:

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification='Timeo
```

Ausgabe:

```
{
  "ClusterId": "j-12ABCDEFGHI34JK",
  "InstanceFleetId": "if-23ABCDEFGHI45JJ"
}
```

- API Einheiten finden Sie [AddInstanceFleet](#) in der AWS CLI Befehlsreferenz.

add-steps

Das folgende Codebeispiel zeigt die Verwendung `add-steps`.

AWS CLI

1. Um benutzerdefinierte JAR Schritte zu einem Cluster hinzuzufügen

Befehl:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps
  Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://mybucket/
mytest.jar,Args=arg1,arg2,arg3
  Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://mybucket/
mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3
```

Erforderliche Parameter:

Jar

Optionale Parameter:

Type, Name, ActionOnFailure, Args

Ausgabe:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

2. Um Streaming-Schritte zu einem Cluster hinzuzufügen

Befehl:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=STREAMING,Name='Streaming
Program',ActionOnFailure=CONTINUE,Args=[-files,s3://elasticmapreduce/samples/
wordcount/wordSplitter.py,-mapper,wordSplitter.py,-reducer,aggregate,-input,s3://
elasticmapreduce/samples/wordcount/input,-output,s3://mybucket/wordcount/output]
```

Erforderliche Parameter:

```
Type, Args
```

Optionale Parameter:

```
Name, ActionOnFailure
```

JSON-Äquivalent (Inhalt von step.json):

```
[
  {
    "Name": "JSON Streaming Step",
    "Args": ["-files", "s3://elasticmapreduce/samples/wordcount/wordSplitter.py", "-mapper", "wordSplitter.py", "-reducer", "aggregate", "-input", "s3://elasticmapreduce/samples/wordcount/input", "-output", "s3://mybucket/wordcount/output"],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
  }
]
```

NOTE: JSON Argumente müssen Optionen und Werte als eigene Elemente in der Liste enthalten.

Befehl (mit step.json):

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file:///./step.json
```

Ausgabe:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

3. Um einen Streaming-Schritt mit mehreren Dateien zu einem Cluster hinzuzufügen (JSON nur)

JSON (mehrere Dateien.json):

```
[
  {
    "Name": "JSON Streaming Step",
    "Type": "STREAMING",
    "ActionOnFailure": "CONTINUE",
    "Args": [
      "-files",
      "s3://mybucket/mapper.py,s3://mybucket/reducer.py",
      "-mapper",
      "mapper.py",
      "-reducer",
      "reducer.py",
      "-input",
      "s3://mybucket/input",
      "-output",
      "s3://mybucket/output"]
  }
]
```

Befehl:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file://./multiplefiles.json
```

Erforderliche Parameter:

Type, Args

Optionale Parameter:

Name, ActionOnFailure

Ausgabe:

```
{
  "StepIds":[
    "s-XXXXXXXX",
  ]
}
```

4. Um Hive-Schritte zu einem Cluster hinzuzufügen

Befehl:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=HIVE,Name='Hive
program',ActionOnFailure=CONTINUE,Args=[-f,s3://mybucket/myhivescript.q,-
d,INPUT=s3://mybucket/myhiveinput,-d,OUTPUT=s3://mybucket/myhiveoutput,arg1,arg2]
Type=HIVE,Name='Hive steps',ActionOnFailure=TERMINATE_CLUSTER,Args=[-
f,s3://elasticmapreduce/samples/hive-ads/libs/model-build.q,-d,INPUT=s3://
elasticmapreduce/samples/hive-ads/tables,-d,OUTPUT=s3://mybucket/hive-ads/
output/2014-04-18/11-07-32,-d,LIBS=s3://elasticmapreduce/samples/hive-ads/libs]
```

Erforderliche Parameter:

Type, Args

Optionale Parameter:

Name, ActionOnFailure

Ausgabe:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

5. Um Pig-Schritte zu einem Cluster hinzuzufügen**Befehl:**

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=PIG,Name='Pig
program',ActionOnFailure=CONTINUE,Args=[-f,s3://mybucket/mypigscript.pig,-
p,INPUT=s3://mybucket/mypiginput,-p,OUTPUT=s3://mybucket/mypigoutput,arg1,arg2]
Type=PIG,Name='Pig program',Args=[-f,s3://elasticmapreduce/samples/pig-apache/do-
reports2.pig,-p,INPUT=s3://elasticmapreduce/samples/pig-apache/input,-p,OUTPUT=s3://
mybucket/pig-apache/output,arg1,arg2]
```

Erforderliche Parameter:

Type, Args

Optionale Parameter:

Name, ActionOnFailure

Ausgabe:

```
{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

6. Um Impala-Schritte zu einem Cluster hinzuzufügen

Befehl:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=IMPALA,Name='Impala
program',ActionOnFailure=CONTINUE,Args=--impala-script,s3://myimpala/input,--
console-output-path,s3://myimpala/output
```

Erforderliche Parameter:

Type, Args

Optionale Parameter:

Name, ActionOnFailure

Ausgabe:

```
{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

- API-Einheiten finden Sie [AddSteps](#) in der AWS CLI Befehlsreferenz.

add-tags

Das folgende Codebeispiel zeigt die Verwendung `add-tags`.

AWS CLI

1. Um Tags zu einem Cluster hinzuzufügen

Befehl:

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
address="123 East NW Seattle"
```

Ausgabe:

```
None
```

2. Um die Tags eines Clusters aufzulisten

--Befehl:

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

Ausgabe:

```
[
  {
    "Value": "male",
    "Key": "sex"
  },
  {
    "Value": "123 East NW Seattle",
    "Key": "address"
  },
  {
    "Value": "John Doe",
    "Key": "name"
  },
  {
```

```
    "Value": "29",  
    "Key": "age"  
  }  
]
```

- API-Einheiten finden Sie [AddTags](#) in der AWS CLI Befehlsreferenz.

create-cluster-examples

Das folgende Codebeispiel zeigt die Verwendung `create-cluster-examples`.

AWS CLI

In den meisten der folgenden Beispiele wird davon ausgegangen, dass Sie Ihre EMR Amazon-Servicerolle und Ihr EC2 Amazon-Instance-Profil angegeben haben. Wenn Sie dies nicht getan haben, müssen Sie jede erforderliche IAM Rolle angeben oder den `--use-default-roles` Parameter bei der Erstellung Ihres Clusters verwenden. Weitere Informationen zur Angabe von IAM Rollen finden [Sie unter Configure IAM Roles for Amazon EMR Permissions to AWS Services](#) im Amazon EMR Management Guide.

Beispiel 1: Um einen Cluster zu erstellen

Das folgende `create-cluster` Beispiel erstellt einen einfachen EMR Cluster.

```
aws emr create-cluster \  
  --release-label emr-5.14.0 \  
  --instance-type m4.large \  
  --instance-count 2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: So erstellen Sie einen EMR Amazon-Cluster mit Standard ServiceRole und InstanceProfile Rollen

Im folgenden `create-cluster` Beispiel wird ein EMR Amazon-Cluster erstellt, der die `--instance-groups` Konfiguration verwendet.

```
aws emr create-cluster \  
  --release-label emr-5.14.0 \  
  --service-role EMR_DefaultRole \  
  --instance-groups
```

```
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \  
--instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Beispiel 3: So erstellen Sie einen EMR Amazon-Cluster, der eine Instance-Flotte verwendet

Im folgenden `create-cluster` Beispiel wird ein EMR Amazon-Cluster erstellt, der die `--instance-fleets` Konfiguration verwendet und zwei Instance-Typen für jede Flotte und zwei EC2 Subnetze angibt.

```
aws emr create-cluster \  
--release-label emr-5.14.0 \  
--service-role EMR_DefaultRole \  
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-  
ab12345c','subnet-de67890f'] \  
--instance-fleets  
InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m4.la  
InstanceFleetType=CORE,TargetSpotCapacity=11,InstanceTypeConfigs=['{InstanceType=m4.large,B
```

Beispiel 4: Um einen Cluster mit Standardrollen zu erstellen

Im folgenden `create-cluster` Beispiel wird der `--use-default-roles` Parameter verwendet, um die Standarddienstrolle und das Instanzprofil anzugeben.

```
aws emr create-cluster \  
--release-label emr-5.9.0 \  
--use-default-roles \  
--instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE  
\  
--auto-terminate
```

Beispiel 5: Um einen Cluster zu erstellen und die zu installierenden Anwendungen anzugeben

Im folgenden `create-cluster` Beispiel wird der `--applications` Parameter verwendet, um die Anwendungen anzugeben, die Amazon EMR installiert. In diesem Beispiel werden Hadoop, Hive und Pig installiert.

```
aws emr create-cluster \  
--applications Name=Hadoop Name=Hive Name=Pig \  
--release-label emr-5.9.0 \  
--auto-terminate
```



```

--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
--auto-terminate

```

Beispiel 6: Um einen Cluster zu erstellen, der Spark enthält

Im folgenden Beispiel wird Spark installiert.

```

aws emr create-cluster \
  --release-label emr-5.9.0 \
  --applications Name=Spark \
  --ec2-attributes KeyName=myKey \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate

```

Beispiel 7: Um einen benutzerdefinierten Wert anzugebenAMI, der für Cluster-Instanzen verwendet werden soll

Das folgende create-cluster Beispiel erstellt eine Cluster-Instance, die auf Amazon Linux AMI mit ID basiertami-a518e6df.

```

aws emr create-cluster \
  --name "Cluster with My Custom AMI" \
  --custom-ami-id ami-a518e6df \
  --ebs-root-volume-size 20 \
  --release-label emr-5.9.0 \
  --use-default-roles \
  --instance-count 2 \
  --instance-type m4.large

```

Beispiel 8: Um Anwendungskonfigurationen anzupassen

In den folgenden Beispielen --configurations wird der Parameter verwendet, um eine JSON Konfigurationsdatei anzugeben, die Anwendungsanpassungen für Hadoop enthält. Weitere Informationen finden Sie unter [Configuring Applications](#) im EMRAmazon-Versionshandbuch.

Inhalt von configurations.json:

```
[
```

```

    {
      "Classification": "mapred-site",
      "Properties": {
        "mapred.tasktracker.map.tasks.maximum": 2
      }
    },
    {
      "Classification": "hadoop-env",
      "Properties": {},
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "HADOOP_DATANODE_HEAPSIZE": 2048,
            "HADOOP_NAMENODE_OPTS": "-XX:GCTimeRatio=19"
          }
        }
      ]
    }
  ]
}
]

```

Das folgende Beispiel verweist auf `configurations.json` eine lokale Datei.

```

aws emr create-cluster \
  --configurations file://configurations.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate

```

Das folgende Beispiel verweist auf `configurations.json` eine Datei in Amazon S3.

```

aws emr create-cluster \
  --configurations https://s3.amazonaws.com/myBucket/configurations.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate

```

Beispiel 9: Um einen Cluster mit Master-, Core- und Task-Instance-Gruppen zu erstellen

Im folgenden `create-cluster` Beispiel werden Typ und Anzahl der EC2 Instanzen angegeben, die für Master-, Core- und Task-Instanzgruppen verwendet `--instance-groups` werden sollen.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --instance-
groups Name=Master,InstanceGroupType=MASTER,InstanceType=m4.large,InstanceCount=1 Name=Core,
```

Beispiel 10: Um anzugeben, dass ein Cluster nach Abschluss aller Schritte beendet werden soll

Im folgenden `create-cluster` Beispiel wird `--auto-terminate` angegeben, dass der Cluster nach Abschluss aller Schritte automatisch heruntergefahren werden soll.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large
InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.large \
  --auto-terminate
```

Beispiel 11: Um Cluster-Konfigurationsdetails wie das EC2 Amazon-Schlüsselpaar, die Netzwerkconfiguration und Sicherheitsgruppen anzugeben

Im folgenden `create-cluster` Beispiel wird ein Cluster mit dem Namen des EC2 Amazon-Schlüsselpaars `myKey` und einem benutzerdefinierten Instance-Profil mit dem Namen `erstelltmyProfile`. Schlüsselpaare werden verwendet, um SSH Verbindungen zu Clusterknoten, meistens zum Master-Knoten, zu autorisieren. Weitere Informationen finden Sie unter [Verwenden eines EC2 Amazon-Schlüsselpaars für SSH Anmeldeinformationen](#) im Amazon EMR Management Guide.

```
aws emr create-cluster \
  --ec2-attributes KeyName=myKey,InstanceProfile=myProfile \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate
```

Das folgende Beispiel erstellt einen Cluster in einem VPC Amazon-Subnetz.

```
aws emr create-cluster \
```

```

--ec2-attributes SubnetId=subnet-xxxxx \
--release-label emr-5.9.0 \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
--auto-terminate

```

Im folgenden Beispiel wird ein Cluster in der us-east-1b Availability Zone erstellt.

```

aws emr create-cluster \
--ec2-attributes AvailabilityZone=us-east-1b \
--release-label emr-5.9.0 \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

Im folgenden Beispiel wird ein Cluster erstellt und nur die von Amazon EMR verwalteten Sicherheitsgruppen angegeben.

```

aws emr create-cluster \
--release-label emr-5.9.0 \
--service-role myServiceRole \
--ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1 \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

Das folgende Beispiel erstellt einen Cluster und spezifiziert nur zusätzliche EC2 Amazon-Sicherheitsgruppen.

```

aws emr create-cluster \
--release-label emr-5.9.0 \
--service-role myServiceRole \
--ec2-attributes InstanceProfile=myRole,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

Das folgende Beispiel erstellt einen Cluster und spezifiziert die EMR -verwalteten Sicherheitsgruppen sowie zusätzliche Sicherheitsgruppen.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --service-role myServiceRole \
  --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Das folgende Beispiel erstellt einen Cluster in einem VPC privaten Subnetz und verwendet eine bestimmte EC2 Amazon-Sicherheitsgruppe, um den EMR Amazon-Servicezugriff zu aktivieren, der für Cluster in privaten Subnetzen erforderlich ist.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --service-role myServiceRole \
  --ec2-attributes InstanceProfile=myRole,ServiceAccessSecurityGroup=sg-service-
access,EmrManagedMasterSecurityGroup=sg-master,EmrManagedSlaveSecurityGroup=sg-slave
\
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Im folgenden Beispiel werden die Konfigurationsparameter für Sicherheitsgruppen mithilfe einer lokal gespeicherten JSON Datei mit `ec2_attributes.json` dem Namen angegeben. NOTE: JSON Argumente müssen Optionen und Werte als eigene Elemente in der Liste enthalten.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --service-role myServiceRole \
  --ec2-attributes file://ec2_attributes.json \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Inhalt von `ec2_attributes.json`:

```
[
  {
    "SubnetId": "subnet-xxxxx",
```

```

    "KeyName": "myKey",
    "InstanceProfile": "myRole",
    "EmrManagedMasterSecurityGroup": "sg-master1",
    "EmrManagedSlaveSecurityGroup": "sg-slave1",
    "ServiceAccessSecurityGroup": "sg-service-access",
    "AdditionalMasterSecurityGroups": ["sg-addMaster1", "sg-addMaster2", "sg-
addMaster3", "sg-addMaster4"],
    "AdditionalSlaveSecurityGroups": ["sg-addSlave1", "sg-addSlave2", "sg-
addSlave3", "sg-addSlave4"]
  }
]

```

Beispiel 12: Um das Debuggen zu aktivieren und ein Protokoll anzugeben URI

Im folgenden `create-cluster` Beispiel wird der `--enable-debugging` Parameter verwendet, mit dem Sie Protokolldateien mithilfe des Debugging-Tools in der EMR Amazon-Konsole einfacher anzeigen können. Der `--log-uri` Parameter ist mit `--enable-debugging` erforderlich.

```

aws emr create-cluster \
  --enable-debugging \
  --log-uri s3://myBucket/myLog \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate

```

Beispiel 13: Um beim Erstellen eines Clusters Tags hinzuzufügen

Tags sind Schlüssel-Wert-Paare, die Ihnen helfen, Cluster zu identifizieren und zu verwalten. Im folgenden `create-cluster` Beispiel wird der `--tags` Parameter verwendet, um drei Tags für einen Cluster zu erstellen, eines mit dem Schlüsselnamen `name` und dem Wert `Shirley Rodriguez`, ein zweites mit dem Schlüsselnamen `age` und dem Wert `29` und ein drittes Tag mit dem Schlüsselnamen `department` und dem Wert `Analytics`

```

aws emr create-cluster \
  --tags name="Shirley Rodriguez" age=29 department="Analytics" \
  --release-label emr-5.32.0 \
  --instance-type m5.xlarge \
  --instance-count 3 \
  --use-default-roles

```

Das folgende Beispiel listet die auf einen Cluster angewendeten Tags auf.

```
aws emr describe-cluster \
  --cluster-id j-XXXXXXXXYY \
  --query Cluster.Tags
```

Beispiel 14: Um eine Sicherheitskonfiguration zu verwenden, die Verschlüsselung und andere Sicherheitsfunktionen aktiviert

Im folgenden `create-cluster` Beispiel wird der `--security-configuration` Parameter verwendet, um eine Sicherheitskonfiguration für einen EMR Cluster anzugeben. Sie können Sicherheitskonfigurationen mit EMR Amazon-Version 4.8.0 oder höher verwenden.

```
aws emr create-cluster \
  --instance-type m4.large \
  --release-label emr-5.9.0 \
  --security-configuration mySecurityConfiguration
```

Beispiel 15: Um einen Cluster mit zusätzlichen EBS Speichervolumes zu erstellen, die für die Instanzgruppen konfiguriert sind

Bei der Angabe zusätzlicher EBS Volumes sind die folgenden Argumente erforderlich: `VolumeType`, `SizeInGB` if `EbsBlockDeviceConfigs` ist angegeben.

Im folgenden `create-cluster` Beispiel wird ein Cluster mit mehreren EBS Volumes erstellt, die an EC2 Instances in der Core-Instanzgruppe angehängt sind.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --use-default-roles \
  --instance-
groups InstanceGroupType=MASTER, InstanceCount=1, InstanceType=d2.xlarge
'InstanceGroupType=CORE, InstanceCount=2, InstanceType=d2.xlarge, EbsConfiguration={EbsOptimiz
{VolumeSpecification={VolumeType=io1, SizeInGB=100, Iops=100}, VolumesPerInstance=4}}'
\
  --auto-terminate
```

Im folgenden Beispiel wird ein Cluster mit mehreren EBS Volumes erstellt, die an EC2 Instanzen in der Master-Instanzgruppe angehängt sind.

```
aws emr create-cluster \
```

```

--release-label emr-5.9.0 \
--use-default-roles \
--instance-groups 'InstanceGroupType=MASTER, InstanceCount=1,
InstanceType=d2.xlarge, EbsConfiguration={EbsOptimized=true,
EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=io1, SizeInGB=100,
Iops=100}},
{VolumeSpecification={VolumeType=standard, SizeInGB=50}, VolumesPerInstance=3}]}' InstanceGroup
\
--auto-terminate

```

Beispiel 16: Um einen Cluster mit einer automatischen Skalierungsrichtlinie zu erstellen

Mit Amazon EMR Version 4.0 und höher können Sie automatische Skalierungsrichtlinien an Kern- und Task-Instance-Gruppen anhängen. Die automatische Skalierungsrichtlinie fügt EC2 Instances als Reaktion auf eine CloudWatch Amazon-Metrik dynamisch hinzu und entfernt sie. Weitere Informationen finden Sie unter Using Automatic Scaling in Amazon EMR <<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html>>`_ im Amazon Management Guide. EMR

Wenn Sie eine automatische Skalierungsrichtlinie anhängen, müssen Sie auch die Standardrolle für die automatische Skalierung angeben. `--auto-scaling-role EMR_AutoScaling_DefaultRole`

Das folgende `create-cluster` Beispiel spezifiziert die automatische Skalierungsrichtlinie für die CORE Instanzgruppe mithilfe des `AutoScalingPolicy` Arguments mit einer eingebetteten JSON Struktur, die die Konfiguration der Skalierungsrichtlinie spezifiziert. Bei Instanzgruppen mit einer eingebetteten JSON Struktur muss die gesamte Sammlung von Argumenten in einfache Anführungszeichen eingeschlossen sein. Die Verwendung von einfachen Anführungszeichen ist für Instanzgruppen ohne eingebettete JSON Struktur optional.

```

aws emr create-cluster
--release-label emr-5.9.0 \
--use-default-roles --auto-scaling-role EMR_AutoScaling_DefaultRole \
--instance-
groups 'InstanceGroupType=MASTER, InstanceType=d2.xlarge, InstanceCount=1
InstanceGroupType=CORE, InstanceType=d2.xlarge, InstanceCount=2, AutoScalingPolicy={Constrain

```

Im folgenden Beispiel wird eine JSON Datei, `instancegroupconfig.json`, verwendet, um die Konfiguration aller Instanzgruppen in einem Cluster anzugeben. Die JSON Datei spezifiziert die Konfiguration der automatischen Skalierungsrichtlinie für die Kerninstanzgruppe.


```
aws emr create-cluster \  
  --release-label emr-5.9.0 \  
  --service-role EMR_DefaultRole \  
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \  
  --instance-groups file://myfolder/instancegroupconfig.json \  
  --auto-scaling-role EMR_AutoScaling_DefaultRole
```

Inhalt von `instancegroupconfig.json`:

```
[  
  {  
    "InstanceCount": 1,  
    "Name": "MyMasterIG",  
    "InstanceGroupType": "MASTER",  
    "InstanceType": "m4.large"  
  },  
  {  
    "InstanceCount": 2,  
    "Name": "MyCoreIG",  
    "InstanceGroupType": "CORE",  
    "InstanceType": "m4.large",  
    "AutoScalingPolicy": {  
      "Constraints": {  
        "MinCapacity": 2,  
        "MaxCapacity": 10  
      },  
      "Rules": [  
        {  
          "Name": "Default-scale-out",  
          "Description": "Replicates the default scale-out rule in the  
console for YARN memory.",  
          "Action": {  
            "SimpleScalingPolicyConfiguration": {  
              "AdjustmentType": "CHANGE_IN_CAPACITY",  
              "ScalingAdjustment": 1,  
              "CoolDown": 300  
            }  
          },  
          "Trigger": {  
            "CloudWatchAlarmDefinition": {  
              "ComparisonOperator": "LESS_THAN",  
              "EvaluationPeriods": 1,  
              "MetricName": "YARNMemoryAvailablePercentage",
```

```

    "Namespace": "AWS/ElasticMapReduce",
    "Period": 300,
    "Threshold": 15,
    "Statistic": "AVERAGE",
    "Unit": "PERCENT",
    "Dimensions": [
      {
        "Key": "JobFlowId",
        "Value": "${emr.clusterId}"
      }
    ]
  }
}
]

```

Beispiel 17: Fügen Sie beim Erstellen eines Clusters benutzerdefinierte JAR Schritte hinzu

Im folgenden `create-cluster` Beispiel werden Schritte hinzugefügt, indem eine in Amazon S3 gespeicherte JAR Datei angegeben wird. Die Schritte leiten die Arbeit an einen Cluster weiter. Die in der JAR Datei definierte Hauptfunktion wird ausgeführt, nachdem EC2 Instanzen bereitgestellt, alle Bootstrap-Aktionen ausgeführt und Anwendungen installiert wurden. Die Schritte werden mit `Type=CUSTOM_JAR` angegeben.

Für benutzerdefinierte JAR Schritte ist der `Jar=` Parameter erforderlich, der den Pfad und den Dateinamen von `angibtJAR`. Optionale Parameter sind `TypeName`, `ActionOnFailure`, `Args`, und `MainClass`. Wenn die Hauptklasse nicht angegeben ist, sollte die JAR Datei dies `MainClass` in ihrer Manifestdatei angeben.

```

aws emr create-cluster \
  --steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
myBucket/
mytest.jar,Args=arg1,arg2,arg3 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
myBucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3 \
  --release-label emr-5.3.1 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
  \
  --auto-terminate

```

Beispiel 18: Um Streaming-Schritte beim Erstellen eines Clusters hinzuzufügen

In den folgenden `create-cluster` Beispielen wird einem Cluster ein Streaming-Schritt hinzugefügt, der beendet wird, nachdem alle Schritte ausgeführt wurden. Streaming-Schritte erfordern Parameter `Type` und `Args`. Die optionalen Parameter für Streaming-Schritte sind `Name` und `ActionOnFailure`.

Das folgende Beispiel spezifiziert den Inline-Schritt.

```
aws emr create-cluster \
  --steps Type=STREAMING,Name='Streaming Program',ActionOnFailure=CONTINUE,Args=[-
files,s3://elasticmapreduce/samples/wordcount/wordSplitter.py,-
mapper,wordSplitter.py,-reducer,aggregate,-input,s3://elasticmapreduce/samples/
wordcount/input,-output,s3://mybucket/wordcount/output] \
  --release-label emr-5.3.1 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate
```

Im folgenden Beispiel wird eine lokal gespeicherte JSON Konfigurationsdatei mit dem Namen `multiplefiles.json` verwendet. Die JSON Konfiguration spezifiziert mehrere Dateien. Um mehrere Dateien innerhalb eines Schritts anzugeben, müssen Sie eine JSON Konfigurationsdatei verwenden, um den Schritt anzugeben. JSON Argumente müssen Optionen und Werte als eigene Elemente in der Liste enthalten.

```
aws emr create-cluster \
  --steps file:///./multiplefiles.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate
```

Inhalt von `multiplefiles.json`:

```
[
  {
    "Name": "JSON Streaming Step",
    "Args": [
      "-files",
      "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
```

```

        "-mapper",
        "wordSplitter.py",
        "-reducer",
        "aggregate",
        "-input",
        "s3://elasticmapreduce/samples/wordcount/input",
        "-output",
        "s3://mybucket/wordcount/output"
    ],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
}
]

```

Beispiel 19: So fügen Sie beim Erstellen eines Clusters Hive-Schritte hinzu

Im folgenden Beispiel werden beim Erstellen eines Clusters Hive-Schritte hinzugefügt. Hive-Schritte erfordern Parameter `Type` und `Args`. Die optionalen Parameter für Hive-Schritte sind `Name` und `ActionOnFailure`.

```

aws emr create-cluster \
  --steps Type=HIVE,Name='Hive
  program',ActionOnFailure=CONTINUE,ActionOnFailure=TERMINATE_CLUSTER,Args=[-
  f,s3://elasticmapreduce/samples/hive-ads/libs/model-build.q,-d,INPUT=s3://
  elasticmapreduce/samples/hive-ads/tables,-d,OUTPUT=s3://mybucket/hive-ads/
  output/2014-04-18/11-07-32,-d,LIBS=s3://elasticmapreduce/samples/hive-ads/libs] \
  --applications Name=Hive \
  --release-label emr-5.3.1 \
  --instance-
  groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

Beispiel 20: Um Pig-Schritte beim Erstellen eines Clusters hinzuzufügen

Im folgenden Beispiel werden Pig-Schritte hinzugefügt, wenn ein Cluster erstellt wird. Die erforderlichen Parameter für Pig-Schritte sind `Type` und `Args`. Die optionalen Parameter für Pig-Schritte sind `Name` und `ActionOnFailure`.

```

aws emr create-cluster \
  --steps Type=PIG,Name='Pig program',ActionOnFailure=CONTINUE,Args=[-f,s3://
  elasticmapreduce/samples/pig-apache/do-reports2.pig,-p,INPUT=s3://elasticmapreduce/
  samples/pig-apache/input,-p,OUTPUT=s3://mybucket/pig-apache/output] \
  --applications Name=Pig \

```

```

--release-label emr-5.3.1 \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

Beispiel 21: Um Bootstrap-Aktionen hinzuzufügen

Im folgenden `create-cluster` Beispiel werden zwei Bootstrap-Aktionen ausgeführt, die als Skripts definiert sind und in Amazon S3 gespeichert sind.

```

aws emr create-cluster \
  --bootstrap-actions Path=s3://mybucket/
myscript1,Name=BootstrapAction1,Args=[arg1,arg2] Path=s3://mybucket/
myscript2,Name=BootstrapAction2,Args=[arg1,arg2] \
  --release-label emr-5.3.1 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate

```

Beispiel 22: Um eine EMRFS konsistente Ansicht zu ermöglichen und die Einstellungen für `RetryCount` und `RetryPeriod` anzupassen

Im folgenden `create-cluster` Beispiel werden die Anzahl der Wiederholungen und der Wiederholungszeitraum für eine EMRFS konsistente Ansicht angegeben. Das Argument `Consistent=true` ist erforderlich.

```

aws emr create-cluster \
  --instance-type m4.large \
  --release-label emr-5.9.0 \
  --emrfs Consistent=true,RetryCount=6,RetryPeriod=30

```

Im folgenden Beispiel wird dieselbe EMRFS Konfiguration wie im vorherigen Beispiel angegeben, wobei eine lokal gespeicherte JSON Konfigurationsdatei mit dem Namen verwendet wird. `emrfsconfig.json`

```

aws emr create-cluster \
  --instance-type m4.large \
  --release-label emr-5.9.0 \
  --emrfs file://emrfsconfig.json

```

Inhalt von `emrfsconfig.json`:

```
{
  "Consistent": true,
  "RetryCount": 6,
  "RetryPeriod": 30
}
```

Beispiel 23: Um einen Cluster mit konfigurierbarem Kerberos zu erstellen

In den folgenden `create-cluster` Beispielen wird ein Cluster mithilfe einer Sicherheitskonfiguration mit aktiviertem Kerberos erstellt und Kerberos-Parameter für den verwendeten Cluster eingerichtet. `--kerberos-attributes`

Mit dem folgenden Befehl werden Kerberos-Attribute für den Inline-Cluster angegeben.

```
aws emr create-cluster \
  --instance-type m3.xlarge \
  --release-label emr-5.10.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --security-configuration mySecurityConfiguration \
  --kerberos-
attributes Realm=EC2.INTERNAL,KdcAdminPassword=123,CrossRealmTrustPrincipalPassword=123
```

Der folgende Befehl gibt dieselben Attribute an, verweist jedoch auf eine lokal gespeicherte JSON Datei mit dem Namen `kerberos_attributes.json`. In diesem Beispiel wird die Datei in demselben Verzeichnis gespeichert, in dem Sie den Befehl ausführen. Sie können auch auf eine in Amazon S3 gespeicherte Konfigurationsdatei verweisen.

```
aws emr create-cluster \
  --instance-type m3.xlarge \
  --release-label emr-5.10.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --security-configuration mySecurityConfiguration \
  --kerberos-attributes file://kerberos_attributes.json
```

Inhalt von `kerberos_attributes.json`:

```
{
  "Realm": "EC2.INTERNAL",
```

```
"KdcAdminPassword": "123",
"CrossRealmTrustPrincipalPassword": "123",
}
```

Im folgenden `create-cluster` Beispiel wird ein EMR Amazon-Cluster erstellt, der die `--instance-groups` Konfiguration verwendet und über eine verwaltete Skalierungsrichtlinie verfügt.

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE,
  --managed-scaling-policy
  ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

Das folgende `create-cluster` Beispiel erstellt einen EMR Amazon-Cluster, der die `--log-encryption-kms-key-id` verwendet, um die KMS Schlüssel-ID zu definieren, die für die Protokollverschlüsselung verwendet wird.

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --log-uri s3://myBucket/myLog \
  --log-encryption-kms-key-id arn:aws:kms:us-east-1:110302272565:key/
dd559181-283e-45d7-99d1-66da348c4d33 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE,
```

Im folgenden `create-cluster` Beispiel wird ein EMR Amazon-Cluster erstellt, der die Konfiguration `--placement-group-configs` verwendet, um Master-Knoten mithilfe einer Platzierungsstrategie in einem Hochverfügbarkeitscluster (HA) innerhalb einer EC2 Platzierungsgruppe zu SPREAD platzieren.

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m4.large InstanceGroupType=CORE,
  \
```

```
--placement-group-configs InstanceRole=MASTER
```

Im folgenden `create-cluster` Beispiel wird ein EMR Amazon-Cluster erstellt, der die Konfiguration „`--auto-termination-policy`“ verwendet, um einen Schwellenwert für die automatische Beendigung des Leerlaufs für den Cluster festzulegen.

```
aws emr create-cluster \
  --release-label emr-5.34.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE
\
  --auto-termination-policy IdleTimeout=100
```

Das folgende `create-cluster` Beispiel erstellt einen EMR Amazon-Cluster, der „`--os-release-label`“ verwendet, um eine Amazon Linux-Version für den Clusterstart zu definieren

```
aws emr create-cluster \
  --release-label emr-6.6.0 \
  --os-release-label 2.0.20220406.1 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE
```

Beispiel 24: So geben Sie die Attribute eines EBS Root-Volumes an: Größe, IOPS und Durchsatz für Cluster-Instances, die mit den EMR Versionen 6.15.0 und höher erstellt wurden

Im folgenden `create-cluster` Beispiel wird ein EMR Amazon-Cluster erstellt, der Root-Volume-Attribute verwendet, um Root-Volume-Spezifikationen für die EC2 Instances zu konfigurieren.

```
aws emr create-cluster \
  --name "Cluster with My Custom AMI" \
  --custom-ami-id ami-a518e6df \
  --ebs-root-volume-size 20 \
  --ebs-root-volume-iops 3000 \
  --ebs-root-volume-throughput 125 \
  --release-label emr-6.15.0 \
  --use-default-roles \
```



```
--instance-count 2 \  
--instance-type m4.Large
```

- API-Einheiten finden Sie [CreateClusterExamples](#) unter AWS CLI Befehlsreferenz.

create-default-roles

Das folgende Codebeispiel zeigt die Verwendung `create-default-roles`.

AWS CLI

1. Um die IAM Standardrolle für zu erstellen EC2

Befehl:

```
aws emr create-default-roles
```

Ausgabe:

If the role already exists then the command returns nothing.

If the role does not exist then the output will be:

```
[  
  {  
    "RolePolicy": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": [  
            "cloudwatch:*",  
            "dynamodb:*",  
            "ec2:Describe*",  
            "elasticmapreduce:Describe*",  
            "elasticmapreduce:ListBootstrapActions",  
            "elasticmapreduce:ListClusters",  
            "elasticmapreduce:ListInstanceGroups",  
            "elasticmapreduce:ListInstances",  
            "elasticmapreduce:ListSteps",  
            "kinesis:CreateStream",  
            "kinesis>DeleteStream",  
            "kinesis:DescribeStream",  
            "kinesis:GetRecords",
```

```

        "kinesis:GetShardIterator",
        "kinesis:MergeShards",
        "kinesis:PutRecord",
        "kinesis:SplitShard",
        "rds:Describe*",
        "s3:*",
        "sdb:*",
        "sns:*",
        "sqs:*"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
},
"Role": {
    "AssumeRolePolicyDocument": {
        "Version": "2008-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AR0AIQ5SIQUGL5KMYBJX6",
    "CreateDate": "2015-06-09T17:09:04.602Z",
    "RoleName": "EMR_EC2_DefaultRole",
    "Path": "/",
    "Arn": "arn:aws:iam::176430881729:role/EMR_EC2_DefaultRole"
}
},
{
    "RolePolicy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:AuthorizeSecurityGroupIngress",
                    "ec2:CancelSpotInstanceRequests",

```

```

        "ec2:CreateSecurityGroup",
        "ec2:CreateTags",
        "ec2>DeleteTags",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeKeyPairs",
        "ec2:DescribePrefixLists",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcs",
        "ec2:ModifyImageAttribute",
        "ec2:ModifyInstanceAttribute",
        "ec2:RequestSpotInstances",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListInstanceProfiles",
        "iam:ListRolePolicies",
        "iam:PassRole",
        "s3:CreateBucket",
        "s3:Get*",
        "s3:List*",
        "sdb:BatchPutAttributes",
        "sdb:Select",
        "sqs:CreateQueue",
        "sqs>Delete*",
        "sqs:GetQueue*",
        "sqs:ReceiveMessage"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
},
"Role": {

```

```

    "AssumeRolePolicyDocument": {
      "Version": "2008-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "elasticmapreduce.amazonaws.com"
          }
        }
      ]
    },
    "RoleId": "AROAI3SRVPPVSRDLARBPY",
    "CreateDate": "2015-06-09T17:09:10.401Z",
    "RoleName": "EMR_DefaultRole",
    "Path": "/",
    "Arn": "arn:aws:iam::176430881729:role/EMR_DefaultRole"
  }
}
]

```

- API-Einheiten finden Sie [CreateDefaultRoles](#) in der AWS CLI Befehlsreferenz.

create-security-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-security-configuration`.

AWS CLI

1. Um eine Sicherheitskonfiguration zu erstellen, bei der die Verschlüsselung während der Übertragung PEM für den Zertifikatsanbieter und die Verschlüsselung im Ruhezustand mit SSE-S3 für die S3-Verschlüsselung und AWS-KMS für den lokalen Festplattenschlüsselanbieter aktiviert ist

Befehl:

```

aws emr create-security-configuration --name MySecurityConfig --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption" : true,
    "EnableAtRestEncryption" : true,

```

```

        "InTransitEncryptionConfiguration" : {
            "TLSCertificateConfiguration" : {
                "CertificateProviderType" : "PEM",
                "S3Object" : "s3://mycertstore/artifacts/
MyCerts.zip"
            }
        },
        "AtRestEncryptionConfiguration" : {
            "S3EncryptionConfiguration" : {
                "EncryptionMode" : "SSE-S3"
            },
            "LocalDiskEncryptionConfiguration" : {
                "EncryptionKeyProviderType" : "AwsKms",
                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
    }
}'

```

Ausgabe:

```

{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}

```

JSON-Äquivalent (Inhalt von security_configuration.json):

```

{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": true,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "PEM",
        "S3Object": "s3://mycertstore/artifacts/MyCerts.zip"
      }
    },
    "AtRestEncryptionConfiguration": {
      "S3EncryptionConfiguration": {
        "EncryptionMode": "SSE-S3"
      }
    }
  }
}

```

```

        "LocalDiskEncryptionConfiguration": {
            "EncryptionKeyProviderType": "AwsKms",
            "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
        }
    }
}

```

Befehl (mit `security_configuration.json`):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

Ausgabe:

```

{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}

```

2. Um eine Sicherheitskonfiguration mit aktiviertem Kerberos mithilfe von KDC Cluster-Dedicated und Cross-Realm Trust zu erstellen

Befehl:

```

aws emr create-security-configuration --name MySecurityConfig --security-
configuration '{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
          "Domain": "ad.domain.com",
          "AdminServer": "ad.domain.com",
          "KdcServer": "ad.domain.com"
        }
      }
    }
  }
}

```

```
}'
```

Ausgabe:

```
{  
  "CreationDateTime": 1490225558.982,  
  "Name": "MySecurityConfig"  
}
```

JSON-Äquivalent (Inhalt von security_configuration.json):

```
{  
  "AuthenticationConfiguration": {  
    "KerberosConfiguration": {  
      "Provider": "ClusterDedicatedKdc",  
      "ClusterDedicatedKdcConfiguration": {  
        "TicketLifetimeInHours": 24,  
        "CrossRealmTrustConfiguration": {  
          "Realm": "AD.DOMAIN.COM",  
          "Domain": "ad.domain.com",  
          "AdminServer": "ad.domain.com",  
          "KdcServer": "ad.domain.com"  
        }  
      }  
    }  
  }  
}
```

Befehl (mit security_configuration.json):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-  
configuration file://./security_configuration.json
```

Ausgabe:

```
{  
  "CreationDateTime": 1490225558.982,  
  "Name": "MySecurityConfig"  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [CreateSecurityConfiguration](#).AWS CLI

delete-security-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-security-configuration`.

AWS CLI

Um eine Sicherheitskonfiguration in der aktuellen Region zu löschen

Befehl:

```
aws emr delete-security-configuration --name MySecurityConfig
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [DeleteSecurityConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-cluster

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster`.

AWS CLI

Befehl:

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

Ausgabe:

```
For release-label based uniform instance groups cluster:
{
  "Cluster": {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1436475075.199,
        "CreationDateTime": 1436474656.563,
      },
      "State": "WAITING",
      "StateChangeReason": {
        "Message": "Waiting for steps to run"
      }
    }
  }
}
```



```
    }
  },
  "Ec2InstanceAttributes": {
    "ServiceAccessSecurityGroup": "sg-xxxxxxx",
    "EmrManagedMasterSecurityGroup": "sg-xxxxxxx",
    "IamInstanceProfile": "EMR_EC2_DefaultRole",
    "Ec2KeyName": "myKey",
    "Ec2AvailabilityZone": "us-east-1c",
    "EmrManagedSlaveSecurityGroup": "sg-yyyyyyyyy"
  },
  "Name": "My Cluster",
  "ServiceRole": "EMR_DefaultRole",
  "Tags": [],
  "TerminationProtected": true,
  "UnhealthyNodeReplacement": true,
  "ReleaseLabel": "emr-4.0.0",
  "NormalizedInstanceHours": 96,
  "InstanceGroups": [
    {
      "RequestedInstanceCount": 2,
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1436475074.245,
          "CreationDateTime": 1436474656.564,
          "EndDateTime": 1436638158.387
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      },
      "Name": "CORE",
      "InstanceGroupType": "CORE",
      "Id": "ig-YYYYYYY",
      "Configurations": [],
      "InstanceType": "m3.large",
      "Market": "ON_DEMAND",
      "RunningInstanceCount": 2
    },
    {
      "RequestedInstanceCount": 1,
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1436475074.245,
```

```

        "CreationDateTime": 1436474656.564,
        "EndDateTime": 1436638158.387
    },
    "State": "RUNNING",
    "StateChangeReason": {
        "Message": "",
    }
},
"Name": "MASTER",
"InstanceGroupType": "MASTER",
"Id": "ig-XXXXXXXXX",
"Configurations": [],
"InstanceType": "m3.large",
"Market": "ON_DEMAND",
"RunningInstanceCount": 1
}
],
"Applications": [
    {
        "Name": "Hadoop"
    }
],
"VisibleToAllUsers": true,
"BootstrapActions": [],
"MasterPublicDnsName": "ec2-54-147-144-78.compute-1.amazonaws.com",
"AutoTerminate": false,
"Id": "j-XXXXXXXXX",
"Configurations": [
    {
        "Properties": {
            "fs.s3.consistent.retryPeriodSeconds": "20",
            "fs.s3.enableServerSideEncryption": "true",
            "fs.s3.consistent": "false",
            "fs.s3.consistent.retryCount": "2"
        },
        "Classification": "emrfs-site"
    }
]
}
}
}

```

For release-label based instance fleet cluster:

```
{
```

```
"Cluster": {
  "Status": {
    "Timeline": {
      "ReadyDateTime": 1487897289.705,
      "CreationDateTime": 1487896933.942
    },
    "State": "WAITING",
    "StateChangeReason": {
      "Message": "Waiting for steps to run"
    }
  },
  "Ec2InstanceAttributes": {
    "EmrManagedMasterSecurityGroup": "sg-xxxxx",
    "RequestedEc2AvailabilityZones": [],
    "RequestedEc2SubnetIds": [],
    "IamInstanceProfile": "EMR_EC2_DefaultRole",
    "Ec2AvailabilityZone": "us-east-1a",
    "EmrManagedSlaveSecurityGroup": "sg-xxxxx"
  },
  "Name": "My Cluster",
  "ServiceRole": "EMR_DefaultRole",
  "Tags": [],
  "TerminationProtected": false,
  "UnhealthyNodeReplacement": false,
  "ReleaseLabel": "emr-5.2.0",
  "NormalizedInstanceHours": 472,
  "InstanceCollectionType": "INSTANCE_FLEET",
  "InstanceFleets": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1487897212.74,
          "CreationDateTime": 1487896933.948
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      },
      "ProvisionedSpotCapacity": 1,
      "Name": "MASTER",
      "InstanceFleetType": "MASTER",
      "LaunchSpecifications": {
        "SpotSpecification": {
```

```

        "TimeoutDurationMinutes": 60,
        "TimeoutAction": "TERMINATE_CLUSTER"
    }
},
"TargetSpotCapacity": 1,
"ProvisionedOnDemandCapacity": 0,
"InstanceTypeSpecifications": [
    {
        "BidPrice": "0.5",
        "InstanceType": "m3.xlarge",
        "WeightedCapacity": 1
    }
],
"Id": "if-xxxxxxx",
"TargetOnDemandCapacity": 0
}
],
"Applications": [
    {
        "Version": "2.7.3",
        "Name": "Hadoop"
    }
],
"ScaleDownBehavior": "TERMINATE_AT_INSTANCE_HOUR",
"VisibleToAllUsers": true,
"BootstrapActions": [],
"MasterPublicDnsName": "ec2-xxx-xx-xxx-xx.compute-1.amazonaws.com",
"AutoTerminate": false,
"Id": "j-xxxxx",
"Configurations": []
}
}

```

For ami based uniform instance group cluster:

```

{
  "Cluster": {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1399400564.432,
        "CreationDateTime": 1399400268.62
      },
      "State": "WAITING",
      "StateChangeReason": {

```

```
        "Message": "Waiting for steps to run"
      }
    },
    "Ec2InstanceAttributes": {
      "IamInstanceProfile": "EMR_EC2_DefaultRole",
      "Ec2AvailabilityZone": "us-east-1c"
    },
    "Name": "My Cluster",
    "Tags": [],
    "TerminationProtected": true,
    "UnhealthyNodeReplacement": true,
    "RunningAmiVersion": "2.5.4",
    "InstanceGroups": [
      {
        "RequestedInstanceCount": 1,
        "Status": {
          "Timeline": {
            "ReadyDateTime": 1399400558.848,
            "CreationDateTime": 1399400268.621
          },
          "State": "RUNNING",
          "StateChangeReason": {
            "Message": ""
          }
        },
        "Name": "Master instance group",
        "InstanceGroupType": "MASTER",
        "InstanceType": "m1.small",
        "Id": "ig-ABCD",
        "Market": "ON_DEMAND",
        "RunningInstanceCount": 1
      },
      {
        "RequestedInstanceCount": 2,
        "Status": {
          "Timeline": {
            "ReadyDateTime": 1399400564.439,
            "CreationDateTime": 1399400268.621
          },
          "State": "RUNNING",
          "StateChangeReason": {
            "Message": ""
          }
        }
      },
    ]
  },
}
```

```

        "Name": "Core instance group",
        "InstanceGroupType": "CORE",
        "InstanceType": "m1.small",
        "Id": "ig-DEF",
        "Market": "ON_DEMAND",
        "RunningInstanceCount": 2
      }
    ],
    "Applications": [
      {
        "Version": "1.0.3",
        "Name": "hadoop"
      }
    ],
    "BootstrapActions": [],
    "VisibleToAllUsers": false,
    "RequestedAmiVersion": "2.4.2",
    "LogUri": "s3://myLogUri/",
    "AutoTerminate": false,
    "Id": "j-XXXXXXXX"
  }
}

```

- API-Einheiten finden Sie [DescribeCluster](#) in der AWS CLI Befehlsreferenz.

describe-step

Das folgende Codebeispiel zeigt die Verwendung `describe-step`.

AWS CLI

Der folgende Befehl beschreibt einen Schritt mit der Schritt-ID `s-3LZC0QUT43AM` in einem Cluster mit der Cluster-ID `j-3SD91U2E1L2QX`:

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

Ausgabe:

```

{
  "Step": {
    "Status": {
      "Timeline": {

```

```

        "EndTime": 1433200470.481,
        "CreationDateTime": 1433199926.597,
        "StartDateTime": 1433200404.959
    },
    "State": "COMPLETED",
    "StateChangeReason": {}
},
"Config": {
    "Args": [
        "s3://us-west-2.elasticmapreduce/libs/hive/hive-script",
        "--base-path",
        "s3://us-west-2.elasticmapreduce/libs/hive/",
        "--install-hive",
        "--hive-versions",
        "0.13.1"
    ],
    "Jar": "s3://us-west-2.elasticmapreduce/libs/script-runner/script-
runner.jar",
    "Properties": {}
},
"Id": "s-3LZC0QUT43AM",
"ActionOnFailure": "TERMINATE_CLUSTER",
"Name": "Setup hive"
}
}

```

- API-Einheiten finden Sie [DescribeStep](#) in der AWS CLI Befehlsreferenz.

get

Das folgende Codebeispiel zeigt die Verwendung `get`.

AWS CLI

Im Folgenden wird das `hadoop-examples.jar` Archiv von der Master-Instance in einem Cluster mit der Cluster-ID heruntergeladen `j-3SD91U2E1L2QX`:

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /home/hadoop-examples.jar --dest ~
```

- API-Einheiten finden Sie unter [Get](#) in AWS CLI Command Reference.

list-clusters

Das folgende Codebeispiel zeigt die Verwendung `list-clusters`.

AWS CLI

Der folgende Befehl listet alle aktiven EMR Cluster in der aktuellen Region auf:

```
aws emr list-clusters --active
```

Ausgabe:

```
{
  "Clusters": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1433200405.353,
          "CreationDateTime": 1433199926.596
        },
        "State": "WAITING",
        "StateChangeReason": {
          "Message": "Waiting after step completed"
        }
      },
      "NormalizedInstanceHours": 6,
      "Id": "j-3SD91U2E1L2QX",
      "Name": "my-cluster"
    }
  ]
}
```

- API-Einheiten finden Sie [ListClusters](#) in der AWS CLI Befehlsreferenz.

list-instance-fleets

Das folgende Codebeispiel zeigt die Verwendung `list-instance-fleets`.

AWS CLI

Um Konfigurationsdetails von Instance-Flotten in einem Cluster abzurufen

In diesem Beispiel werden die Details der Instanzflotten im angegebenen Cluster aufgeführt.

Befehl:

```
list-instance-fleets --cluster-id 'j-12ABCDEFGH134JK'
```

Ausgabe:

```
{
  "InstanceFleets": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1488759094.637,
          "CreationDateTime": 1488758719.817
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      },
      "ProvisionedSpotCapacity": 6,
      "Name": "CORE",
      "InstanceFleetType": "CORE",
      "LaunchSpecifications": {
        "SpotSpecification": {
          "TimeoutDurationMinutes": 60,
          "TimeoutAction": "TERMINATE_CLUSTER"
        }
      },
      "ProvisionedOnDemandCapacity": 2,
      "InstanceTypeSpecifications": [
        {
          "BidPrice": "0.5",
          "InstanceType": "m3.xlarge",
          "WeightedCapacity": 2
        }
      ],
      "Id": "if-1ABC2DEFGHIJ3"
    },
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1488759058.598,
          "CreationDateTime": 1488758719.811
        }
      }
    }
  ]
}
```

```

        },
        "State": "RUNNING",
        "StateChangeReason": {
            "Message": ""
        }
    },
    "ProvisionedSpotCapacity": 0,
    "Name": "MASTER",
    "InstanceFleetType": "MASTER",
    "ProvisionedOnDemandCapacity": 1,
    "InstanceTypeSpecifications": [
        {
            "BidPriceAsPercentageOfOnDemandPrice": 100.0,
            "InstanceType": "m3.xlarge",
            "WeightedCapacity": 1
        }
    ],
    "Id": "if-2ABC4DEFGHIJ4"
}
]
}

```

- API-Einheiten finden Sie [ListInstanceFleets](#) in der AWS CLI Befehlsreferenz.

list-instances

Das folgende Codebeispiel zeigt die Verwendung `list-instances`.

AWS CLI

Der folgende Befehl listet alle Instances in einem Cluster mit der Cluster-ID `aufj-3C6XNQ39VR9WL`:

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

Ausgabe:

```

For a uniform instance group based cluster
{
  "Instances": [
    {
      "Status": {

```

```
    "Timeline": {
      "ReadyDateTime": 1433200400.03,
      "CreationDateTime": 1433199960.152
    },
    "State": "RUNNING",
    "StateChangeReason": {}
  },
  "Ec2InstanceId": "i-f19ecfee",
  "PublicDnsName": "ec2-52-52-41-150.us-west-2.compute.amazonaws.com",
  "PrivateDnsName": "ip-172-21-11-216.us-west-2.compute.internal",
  "PublicIpAddress": "52.52.41.150",
  "Id": "ci-3NNHQUQ2TWB6Y",
  "PrivateIpAddress": "172.21.11.216"
},
{
  "Status": {
    "Timeline": {
      "ReadyDateTime": 1433200400.031,
      "CreationDateTime": 1433199949.102
    },
    "State": "RUNNING",
    "StateChangeReason": {}
  },
  "Ec2InstanceId": "i-1feee4c2",
  "PublicDnsName": "ec2-52-63-246-32.us-west-2.compute.amazonaws.com",
  "PrivateDnsName": "ip-172-31-24-130.us-west-2.compute.internal",
  "PublicIpAddress": "52.63.246.32",
  "Id": "ci-GAOCMKNKDCV7",
  "PrivateIpAddress": "172.21.11.215"
},
{
  "Status": {
    "Timeline": {
      "ReadyDateTime": 1433200400.031,
      "CreationDateTime": 1433199949.102
    },
    "State": "RUNNING",
    "StateChangeReason": {}
  },
  "Ec2InstanceId": "i-15cfeee3",
  "PublicDnsName": "ec2-52-25-246-63.us-west-2.compute.amazonaws.com",
  "PrivateDnsName": "ip-172-31-24-129.us-west-2.compute.internal",
  "PublicIpAddress": "52.25.246.63",
  "Id": "ci-2W3TDFFB47UAD",
```

```

        "PrivateIpAddress": "172.21.11.214"
    }
]
}

```

For a fleet based cluster:

```

{
  "Instances": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1487810810.878,
          "CreationDateTime": 1487810588.367,
          "EndDateTime": 1488022990.924
        },
        "State": "TERMINATED",
        "StateChangeReason": {
          "Message": "Instance was terminated."
        }
      },
      "Ec2InstanceId": "i-xxxxx",
      "InstanceFleetId": "if-xxxxx",
      "EbsVolumes": [],
      "PublicDnsName": "ec2-xx-xxx-xxx-xxx.compute-1.amazonaws.com",
      "InstanceType": "m3.xlarge",
      "PrivateDnsName": "ip-xx-xx-xxx-xx.ec2.internal",
      "Market": "SPOT",
      "PublicIpAddress": "xx.xx.xxx.xxx",
      "Id": "ci-xxxxx",
      "PrivateIpAddress": "10.47.191.80"
    }
  ]
}

```

- API-Einheiten finden Sie [ListInstances](#) in der AWS CLI Befehlsreferenz.

list-security-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-security-configurations`.

AWS CLI

Um Sicherheitskonfigurationen in der aktuellen Region aufzulisten

Befehl:

```
aws emr list-security-configurations
```

Ausgabe:

```
{
  "SecurityConfigurations": [
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-1"
    },
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-2"
    }
  ]
}
```

- API-Einheiten finden Sie [ListSecurityConfigurations](#) in der AWS CLI Befehlsreferenz.

list-steps

Das folgende Codebeispiel zeigt die Verwendung `list-steps`.

AWS CLI

Der folgende Befehl listet alle Schritte in einem Cluster mit der Cluster-ID auf `j-3SD91U2E1L2QX`:

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- API-Einheiten finden Sie [ListSteps](#) in der AWS CLI Befehlsreferenz.

modify-cluster-attributes

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-attributes`.

AWS CLI

Mit dem folgenden Befehl wird die Sichtbarkeit eines EMR Clusters mit der ID `j-301CDNY0J5XM4` für alle Benutzer festgelegt:

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-users
```

- API-Einheiten finden Sie [ModifyClusterAttributes](#) in der AWS CLI Befehlsreferenz.

modify-instance-fleet

Das folgende Codebeispiel zeigt die Verwendung `modify-instance-fleet`.

AWS CLI

Um die Zielkapazitäten einer Instance-Flotte zu ändern

In diesem Beispiel werden die Zielkapazitäten On-Demand und Spot für die angegebene Instance-Flotte auf 1 geändert.

Befehl:

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGH134JK' --instance-fleet InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- API-Einheiten finden Sie [ModifyInstanceFleet](#) unter AWS CLI Befehlsreferenz.

put

Das folgende Codebeispiel zeigt die Verwendung `put`.

AWS CLI

Der folgende Befehl lädt eine Datei mit dem Namen `healthcheck.sh` auf die Master-Instance in einem Cluster mit der Cluster-ID `j-3SD91U2E1L2QX` hoch:

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- API-Einheiten finden Sie unter AWS CLI Befehlsreferenz [einfügen](#).

remove-tags

Das folgende Codebeispiel zeigt die Verwendung `remove-tags`.

AWS CLI

Der folgende Befehl entfernt ein Tag mit dem Schlüssel `prod` aus einem Cluster mit der Cluster-ID `j-3SD91U2E1L2QX`:

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- API-Einheiten finden Sie [RemoveTags](#) in der AWS CLI Befehlsreferenz.

schedule-hbase-backup

Das folgende Codebeispiel zeigt die Verwendung `schedule-hbase-backup`.

AWS CLI

Hinweis: Dieser Befehl kann nur mit HBase AMI Version 2.x und 3.x verwendet werden

1. Um ein vollständiges HBase Backup zu planen >>>>>>
`06ab6d6e13564b5733d75abaf3b599f93cf39a23`

Befehl:

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type full --dir  
s3://myBucket/backup --interval 10 --unit hours --start-time  
2014-04-21T05:26:10Z --consistent
```

Ausgabe:

```
None
```

2. Um ein inkrementelles Backup HBase zu planen

Befehl:

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type incremental  
--dir s3://myBucket/backup --interval 30 --unit minutes --start-time  
2014-04-21T05:26:10Z --consistent
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [ScheduleHbaseBackup](#) in der AWS CLI Befehlsreferenz.

socks

Das folgende Codebeispiel zeigt die Verwendung socks.

AWS CLI

Der folgende Befehl öffnet eine Socks-Verbindung mit der Master-Instance in einem Cluster mit der Cluster-ID `j-3SD91U2E1L2QX`:

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

Die Option Schlüsselpaardatei verwendet einen lokalen Pfad zu einer privaten Schlüsseldatei.

- API-Einheiten finden Sie unter [Socks](#) in der AWS CLI Befehlsreferenz.

ssh

Das folgende Codebeispiel zeigt die Verwendung ssh.

AWS CLI

Der folgende Befehl öffnet eine SSH-Verbindung mit der Master-Instance in einem Cluster mit der Cluster-ID: `j-3SD91U2E1L2QX`

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

Die Option Schlüsselpaardatei verwendet einen lokalen Pfad zu einer privaten Schlüsseldatei.

Ausgabe:

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i /home/local/user/.ssh/mykey.pem hadoop@ec2-52-52-41-150.us-west-2.compute.amazonaws.com
Warning: Permanently added 'ec2-52-52-41-150.us-west-2.compute.amazonaws.com,52.52.41.150' (ECDSA) to the list of known hosts.
Last login: Mon Jun  1 23:15:38 2015
```



```

_|  _|_ )
_| (   /  Amazon Linux AMI
_| \|_ | _|

```

```

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
26 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.

```

```

-----

Welcome to Amazon Elastic MapReduce running Hadoop and Amazon Linux.

```

```

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.

```

```

The Hadoop UI can be accessed via the following commands:

```

```

ResourceManager    lynx http://ip-172-21-11-216:9026/
NameNode           lynx http://ip-172-21-11-216:9101/

```

```

-----

[hadoop@ip-172-31-16-216 ~]$

```

- API-Einheiten finden Sie unter [Ssh](#) in der AWS CLI Befehlsreferenz.

Amazon EMR an EKS Beispielen mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie die AWS Command Line Interface mit Amazon EMR on verwenden EKS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

update-role-trust-policy

Das folgende Codebeispiel zeigt die Verwendung `update-role-trust-policy`.

AWS CLI

Um die Vertrauensrichtlinie einer IAM Rolle zu aktualisieren, die mit EMR Amazon verwendet werden soll EKS

Dieser Beispielbefehl aktualisiert die Vertrauensrichtlinie einer Rolle namens `example_iam_role`, sodass sie mit Amazon EMR on EKS mit dem Namespace `example_namespace` aus einem Cluster namens `example_cluster` verwendet werden kann. EKS

Befehl:

```
aws emr-containers update-role-trust-policy \  
  --cluster example_cluster \  
  --namespace example_namespace \  
  --role-name example_iam_role
```

Ausgabe:

```
If the trust policy has already been updated, then the output will be:  
Trust policy statement already exists for role example_iam_role. No  
changes were made!
```

```
If the trust policy has not been updated yet, then the output will be:  
Successfully updated trust policy of role example_iam_role.
```

- Einzelheiten [UpdateRoleTrustPolicy](#) finden Sie in der Befehlsreferenz. API AWS CLI

EventBridge Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren EventBridge.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-rule`.

AWS CLI

Um eine CloudWatch Ereignisregel zu löschen

In diesem Beispiel wird die Regel mit dem Namen `EC2InstanceStateChanges` gelöscht:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- API-Einheiten finden Sie [DeleteRule](#) in der AWS CLI Befehlsreferenz.

describe-rule

Das folgende Codebeispiel zeigt die Verwendung `describe-rule`.

AWS CLI

Um Informationen über eine CloudWatch Ereignisregel anzuzeigen

In diesem Beispiel werden Informationen zu der Regel mit dem Namen `DailyLambdaFunction` angezeigt:

```
aws events describe-rule --name "DailyLambdaFunction"
```

- API-Einheiten finden Sie [DescribeRule](#) unter AWS CLI Befehlsreferenz.

disable-rule

Das folgende Codebeispiel zeigt die Verwendung `disable-rule`.

AWS CLI

Um eine CloudWatch Ereignisregel zu deaktivieren

In diesem Beispiel wird die genannte DailyLambdaFunction Regel deaktiviert. Die Regel wird nicht gelöscht:

```
aws events disable-rule --name "DailyLambdaFunction"
```

- APIEinzelheiten finden Sie [DisableRule](#) in der AWS CLI Befehlsreferenz.

enable-rule

Das folgende Codebeispiel zeigt die Verwendung `enable-rule`.

AWS CLI

Um eine CloudWatch Ereignisregel zu aktivieren

In diesem Beispiel wird die genannte Regel aktiviert DailyLambdaFunction, die zuvor deaktiviert wurde:

```
aws events enable-rule --name "DailyLambdaFunction"
```

- APIEinzelheiten finden Sie [EnableRule](#) in der AWS CLI Befehlsreferenz.

list-rule-names-by-target

Das folgende Codebeispiel zeigt die Verwendung `list-rule-names-by-target`.

AWS CLI

So zeigen Sie alle Regeln mit einem bestimmten Ziel an

In diesem Beispiel werden alle Regeln angezeigt, deren Ziel die Lambda-Funktion MyFunctionName "" ist:

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

- APIEinzelheiten finden Sie [ListRuleNamesByTarget](#) in der AWS CLI Befehlsreferenz.

list-rules

Das folgende Codebeispiel zeigt die Verwendung `list-rules`.

AWS CLI

Um eine Liste aller CloudWatch Event-Regeln anzuzeigen

In diesem Beispiel werden alle CloudWatch Event-Regeln in der Region angezeigt:

```
aws events list-rules
```

Um eine Liste von CloudWatch Event-Regeln anzuzeigen, die mit einer bestimmten Zeichenfolge beginnen.

In diesem Beispiel werden alle CloudWatch Event-Regeln in der Region angezeigt, deren Name mit „Täglich“ beginnt:

```
aws events list-rules --name-prefix "Daily"
```

- API-Einheiten finden Sie [ListRules](#) unter AWS CLI Befehlsreferenz.

list-targets-by-rule

Das folgende Codebeispiel zeigt die Verwendung `list-targets-by-rule`.

AWS CLI

Um alle Ziele für eine CloudWatch Ereignisregel anzuzeigen

In diesem Beispiel werden alle Ziele der Regel mit dem Namen angezeigt `DailyLambdaFunction`:

```
aws events list-targets-by-rule --rule "DailyLambdaFunction"
```

- API-Einheiten finden Sie [ListTargetsByRule](#) in der AWS CLI Befehlsreferenz.

put-events

Das folgende Codebeispiel zeigt die Verwendung `put-events`.

AWS CLI

Um ein benutzerdefiniertes Ereignis an CloudWatch Events zu senden

In diesem Beispiel wird ein benutzerdefiniertes Ereignis an CloudWatch Events gesendet. Das Ereignis ist in der Datei `putevents.json` enthalten:

```
aws events put-events --entries file://putevents.json
```

Die Datei `putevents.json` hat folgenden Inhalt:

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  },
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  }
]
```

- API-Einheiten finden Sie [PutEvents](#) in der AWS CLI Befehlsreferenz.

put-rule

Das folgende Codebeispiel zeigt die Verwendung `put-rule`.

AWS CLI

Um Regeln für CloudWatch Ereignisse zu erstellen

In diesem Beispiel wird eine Regel erstellt, die jeden Tag um 9:00 Uhr (UTC) ausgelöst wird. Wenn Sie `put-targets` verwenden, um eine Lambda-Funktion als Ziel dieser Regel hinzuzufügen, können Sie die Lambda-Funktion jeden Tag zur angegebenen Zeit ausführen:

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 * * ? *)"
```

In diesem Beispiel wird eine Regel erstellt, die ausgelöst wird, wenn eine EC2 Instanz in der Region ihren Status ändert:

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"source\": [\"aws.ec2\"], \"detail-type\": [\"EC2 Instance State-change Notification\"]}" --role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

In diesem Beispiel wird eine Regel erstellt, die ausgelöst wird, wenn eine EC2 Instanz in der Region gestoppt oder beendet wird:

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-pattern "{\"source\": [\"aws.ec2\"], \"detail-type\": [\"EC2 Instance State-change Notification\"], \"detail\": {\"state\": [\"stopped\", \"terminated\"]}}" --role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- API-Einheiten finden Sie [PutRule](#) unter AWS CLI Befehlsreferenz.

put-targets

Das folgende Codebeispiel zeigt die Verwendung `put-targets`.

AWS CLI

Um Ziele für CloudWatch Event-Regeln hinzuzufügen

Im folgenden Beispiel wird eine Lambda-Funktion als Ziel einer Regel hinzugefügt:

```
aws events put-targets --rule DailyLambdaFunction --targets "Id"="1", "Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Im folgenden Beispiel wird ein Amazon-Kinesis-Stream als Ziel festgelegt, sodass Ereignisse, die von dieser Regel erfasst werden, an den Stream weitergeleitet werden:

```
aws events put-targets --rule EC2InstanceStateChanges --targets  
  "Id"="1", "Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/  
  MyStream", "RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Im folgenden Beispiel werden zwei Amazon-Kinesis-Streams als Ziele für eine Regel festgelegt:

```
aws events put-targets --rule DailyLambdaFunction --targets  
  "Id"="Target1", "Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/  
  MyStream1", "RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda"  
  "Id"="Target2", "Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/  
  MyStream2", "RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- API-Einheiten finden Sie [PutTargets](#) in der AWS CLI Befehlsreferenz.

remove-targets

Das folgende Codebeispiel zeigt die Verwendung `remove-targets`.

AWS CLI

So entfernen Sie ein Ziel für ein Ereignis

In diesem Beispiel wird der Amazon Kinesis Kinesis-Stream mit dem Namen MyStream 1 als Ziel der Regel DailyLambdaFunction entfernt. Bei DailyLambdaFunction seiner Erstellung wurde dieser Stream als Ziel mit der ID Target1 festgelegt:

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- API-Einheiten finden Sie [RemoveTargets](#) in der AWS CLI Befehlsreferenz.

test-event-pattern

Das folgende Codebeispiel zeigt die Verwendung `test-event-pattern`.

AWS CLI

Um zu überprüfen, ob ein Ereignismuster mit einem angegebenen Ereignis übereinstimmt

In diesem Beispiel wird getestet, ob das Muster „source:com.mycompany.myapp“ mit dem angegebenen Ereignis übereinstimmt. In diesem Beispiel wäre die Ausgabe „true“:


```
aws events test-event-pattern --event-pattern "{\"source\":[\"com.mycompany.myapp\n\"]}" --event "{\"id\":\"1\", \"source\":\"com.mycompany.myapp\", \"detail-type\":\n\"myDetailType\", \"account\":\"123456789012\", \"region\":\"us-east-1\", \"time\":\n\"2017-04-11T20:11:04Z\"}"
```

- API-Einheiten finden Sie [TestEventPattern](#) in der AWS CLI Befehlsreferenz.

Beispiele für Firewall Manager mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Firewall Manager Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-admin-account

Das folgende Codebeispiel zeigt die Verwendung `associate-admin-account`.

AWS CLI

So richten Sie das Firewall Manager Manager-Administratorkonto ein

Im folgenden `associate-admin-account` Beispiel wird das Administratorkonto für Firewall Manager eingerichtet.

```
aws fms associate-admin-account \  
  --admin-account 123456789012
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten des AWS Firewall Manager-Administratorkontos](#) im AWS WAFEntwicklerhandbuch für AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [AssociateAdminAccount](#)in der AWS CLI Befehlsreferenz.

delete-notification-channel

Das folgende Codebeispiel zeigt die Verwendung**delete-notification-channel**.

AWS CLI

So entfernen Sie die SNS Themeninformationen für Firewall Manager Manager-Protokolle

Im folgenden `delete-notification-channel` Beispiel werden die SNS Themeninformationen entfernt.

```
aws fms delete-notification-channel
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Konfiguration von SNS Amazon-Benachrichtigungen und CloudWatch Amazon-Alarmen](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [DeleteNotificationChannel](#)in der AWS CLI Befehlsreferenz.

delete-policy

Das folgende Codebeispiel zeigt die Verwendung**delete-policy**.

AWS CLI

So löschen Sie eine Firewall Manager Manager-Richtlinie

Im folgenden `delete-policy` Beispiel wird die Richtlinie mit der angegebenen ID zusammen mit all ihren Ressourcen entfernt.

```
aws fms delete-policy \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --delete-all-policy-resources
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Firewall Manager-Richtlinien](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [DeletePolicy](#)in der AWS CLI Befehlsreferenz.

disassociate-admin-account

Das folgende Codebeispiel zeigt die Verwendung `disassociate-admin-account`.

AWS CLI

So entfernen Sie das Firewall Manager Manager-Administratorkonto

Im folgenden `disassociate-admin-account` Beispiel wird die aktuelle Administratorkontenverknüpfung aus Firewall Manager entfernt.

```
aws fms disassociate-admin-account
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten des AWS Firewall Manager-Administratorkontos](#) im AWS WAFEntwicklerhandbuch für AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [DisassociateAdminAccount](#)in der AWS CLI Befehlsreferenz.

get-admin-account

Das folgende Codebeispiel zeigt die Verwendung `get-admin-account`.

AWS CLI

So rufen Sie das Firewall Manager Manager-Administratorkonto ab

Im folgenden `get-admin-account` Beispiel wird das Administratorkonto abgerufen.

```
aws fms get-admin-account
```

Ausgabe:

```
{
  "AdminAccount": "123456789012",
  "RoleStatus": "READY"
```

```
}
```

Weitere Informationen finden Sie unter [Voraussetzungen für AWS Firewall Manager](#) im AWS WAFEntwicklerhandbuch für AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [GetAdminAccount](#) in der AWS CLI Befehlsreferenz.

get-compliance-detail

Das folgende Codebeispiel zeigt die Verwendung `get-compliance-detail`.

AWS CLI

Um die Compliance-Informationen für ein Konto abzurufen

Im folgenden `get-compliance-detail` Beispiel werden Compliance-Informationen für die angegebene Richtlinie und das angegebene Mitgliedskonto abgerufen.

```
aws fms get-compliance-detail \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --member-account 123456789012
```

Ausgabe:

```
{  
  "PolicyComplianceDetail": {  
    "EvaluationLimitExceeded": false,  
    "IssueInfoMap": {},  
    "MemberAccount": "123456789012",  
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "PolicyOwner": "123456789012",  
    "Violators": []  
  }  
}
```

Weitere Informationen finden Sie unter [Ressourcenkonformität mit einer Richtlinie anzeigen](#) im AWS WAFEntwicklerhandbuch von AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [GetComplianceDetail](#) in der AWS CLI Befehlsreferenz.

get-notification-channel

Das folgende Codebeispiel zeigt die Verwendung `get-notification-channel`.

AWS CLI

So rufen Sie die SNS Themeninformationen für Firewall Manager Manager-Protokolle ab

Im folgenden `get-notification-channel` Beispiel werden die SNS Themeninformationen abgerufen.

```
aws fms get-notification-channel
```

Ausgabe:

```
{
  "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:us-west-2-fms",
  "SnsRoleName": "arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS"
}
```

Weitere Informationen finden [Sie unter Konfiguration von SNS Amazon-Benachrichtigungen und CloudWatch Amazon-Alarmen](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [GetNotificationChannel](#) in der AWS CLI Befehlsreferenz.

get-policy

Das folgende Codebeispiel zeigt die Verwendung `get-policy`.

AWS CLI

So rufen Sie eine Firewall Manager Manager-Richtlinie ab

Im folgenden `get-policy` Beispiel wird die Richtlinie mit der angegebenen ID abgerufen.

```
aws fms get-policy \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "Policy": {
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```

    "PolicyName": "test",
    "PolicyUpdateToken": "1:p+2RpKR4wPFx7mcrL1U0QQ==",
    "SecurityServicePolicyData": {
      "Type": "SECURITY_GROUPS_COMMON",
      "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_COMMON\",
\\\"revertManualSecurityGroupChanges\\\":true,\\\"exclusiveResourceSecurityGroupManagement
\\\":false,\\\"securityGroups\\\":[\\\"id\\\":\\\"sg-045c43ccc9724e63e\\\"]}\"
    },
    "ResourceType": "AWS::EC2::Instance",
    "ResourceTags": [],
    "ExcludeResourceTags": false,
    "RemediationEnabled": false
  },
  "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/d1ac59b8-938e-42b3-
b2e0-7c620422ddc2"
}

```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Firewall Manager-Richtlinien](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [GetPolicy](#) in der AWS CLI Befehlsreferenz.

list-compliance-status

Das folgende Codebeispiel zeigt die Verwendung `list-compliance-status`.

AWS CLI

Um die Informationen zur Einhaltung der Richtlinien für Mitgliedskonten abzurufen

Im folgenden `list-compliance-status` Beispiel werden Informationen zur Einhaltung der Mitgliedskonten für die angegebene Richtlinie abgerufen.

```

aws fms list-compliance-status \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "PolicyComplianceStatusList": [
    {
      "PolicyOwner": "123456789012",

```

```
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyName": "test",
    "MemberAccount": "123456789012",
    "EvaluationResults": [
      {
        "ComplianceStatus": "COMPLIANT",
        "ViolatorCount": 0,
        "EvaluationLimitExceeded": false
      },
      {
        "ComplianceStatus": "NON_COMPLIANT",
        "ViolatorCount": 2,
        "EvaluationLimitExceeded": false
      }
    ],
    "LastUpdated": 1576283774.0,
    "IssueInfoMap": {}
  }
]
```

Weitere Informationen finden Sie unter [Ressourcenkonformität mit einer Richtlinie anzeigen](#) im AWS WAFEntwicklerhandbuch von AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [ListComplianceStatus](#) in der AWS CLI Befehlsreferenz.

list-member-accounts

Das folgende Codebeispiel zeigt die Verwendung `list-member-accounts`.

AWS CLI

Um die Mitgliedskonten in der Organisation abzurufen

Das folgende `list-member-accounts` Beispiel listet alle Mitgliedskonten auf, die sich in der Organisation des Firewall Manager Manager-Administrators befinden.

```
aws fms list-member-accounts
```

Ausgabe:

```
{
```

```
"MemberAccounts": [  
  "222222222222",  
  "333333333333",  
  "444444444444"  
]  
}
```

Weitere Informationen finden Sie unter [AWS Firewall Manager](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [ListMemberAccounts](#) in der AWS CLI Befehlsreferenz.

list-policies

Das folgende Codebeispiel zeigt die Verwendung `list-policies`.

AWS CLI

So rufen Sie alle Firewall Manager Manager-Richtlinien ab

Im folgenden `list-policies` Beispiel wird die Liste der Richtlinien für das Konto abgerufen. In diesem Beispiel ist die Ausgabe auf zwei Ergebnisse pro Anfrage beschränkt. Bei jedem Aufruf wird ein Wert zurückgegeben `NextToken`, der beim nächsten `list-policies` Aufruf als Wert für den `--starting-token` Parameter verwendet werden kann, um die nächste Gruppe von Ergebnissen für die Liste abzurufen.

```
aws fms list-policies \  
  --max-items 2
```

Ausgabe:

```
{  
  "PolicyList": [  
    {  
      "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "PolicyName": "test",  
      "ResourceType": "AWS::EC2::Instance",  
      "SecurityServiceType": "SECURITY_GROUPS_COMMON",  
      "RemediationEnabled": false
```



```

    },
    {
      "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "PolicyId": "457c9b21-fc94-406c-ae63-21217395ba72",
      "PolicyName": "test",
      "ResourceType": "AWS::EC2::Instance",
      "SecurityServiceType": "SECURITY_GROUPS_COMMON",
      "RemediationEnabled": false
    }
  ],
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Firewall Manager-Richtlinien](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [ListPolicies](#) in der AWS CLI Befehlsreferenz.

put-notification-channel

Das folgende Codebeispiel zeigt die Verwendungput-notification-channel.

AWS CLI

So legen Sie die SNS Themeninformationen für Firewall Manager Manager-Protokolle fest

Im folgenden put-notification-channel Beispiel werden die SNS Themeninformationen festgelegt.

```

aws fms put-notification-channel \
  --sns-topic-arn arn:aws:sns:us-west-2:123456789012:us-west-2-fms \
  --sns-role-name arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Konfiguration von SNS Amazon-Benachrichtigungen und CloudWatch Amazon-Alarmen](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [PutNotificationChannel](#) in der AWS CLI Befehlsreferenz.

put-policy

Das folgende Codebeispiel zeigt die Verwendung `put-policy`.

AWS CLI

So erstellen Sie eine Firewall Manager Manager-Richtlinie

Im folgenden `put-policy` Beispiel wird eine Firewall Manager Manager-Sicherheitsgruppenrichtlinie erstellt.

```
aws fms put-policy \  
  --cli-input-json file://policy.json
```

Inhalt von `policy.json`:

```
{  
  "Policy": {  
    "PolicyName": "test",  
    "SecurityServicePolicyData": {  
      "Type": "SECURITY_GROUPS_USAGE_AUDIT",  
      "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_USAGE_AUDIT\",  
\"deleteUnusedSecurityGroups\":false,\"coalesceRedundantSecurityGroups\":true}"  
    },  
    "ResourceType": "AWS::EC2::SecurityGroup",  
    "ResourceTags": [],  
    "ExcludeResourceTags": false,  
    "RemediationEnabled": false  
  },  
  "TagList": [  
    {  
      "Key": "foo",  
      "Value": "foo"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "Policy": {  
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "PolicyName": "test",  
  }  
}
```

```

    "PolicyUpdateToken": "1:X9QGexP7HASD1sFp+G31Iw==",
    "SecurityServicePolicyData": {
      "Type": "SECURITY_GROUPS_USAGE_AUDIT",
      "ManagedServiceData": "{\"type\": \"SECURITY_GROUPS_USAGE_AUDIT\",
\\\"deleteUnusedSecurityGroups\\\":false,\\\"coalesceRedundantSecurityGroups\\\":true,
\\\"optionalDelayForUnusedInMinutes\\\":null}"
    },
    "ResourceType": "AWS::EC2::SecurityGroup",
    "ResourceTags": [],
    "ExcludeResourceTags": false,
    "RemediationEnabled": false
  },
  "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}

```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Firewall Manager-Richtlinien](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [PutPolicy](#) in der AWS CLI Befehlsreferenz.

AWS FIS Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS FIS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-experiment-template

Das folgende Codebeispiel zeigt die Verwendung `create-experiment-template`.

AWS CLI

Um eine Experimentvorlage zu erstellen

Im folgenden `create-experiment-template` Beispiel wird eine Versuchsvorlage in Ihrem AWS FIS Konto erstellt.

```
aws fis create-experiment-template \  
  --cli-input-json file://myfile.json
```

Inhalt von `myfile.json`:

```
{  
  "description": "experimentTemplate",  
  "stopConditions": [  
    {  
      "source": "aws:cloudwatch:alarm",  
      "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"  
    }  
  ],  
  "targets": {  
    "Instances-Target-1": {  
      "resourceType": "aws:ec2:instance",  
      "resourceArns": [  
        "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"  
      ],  
      "selectionMode": "ALL"  
    }  
  },  
  "actions": {  
    "reboot": {  
      "actionId": "aws:ec2:reboot-instances",  
      "description": "reboot",  
      "parameters": {},  
      "targets": {  
        "Instances": "Instances-Target-1"  
      }  
    }  
  },  
  "roleArn": "arn:aws:iam::123456789012:role/myRole"  
}
```

Ausgabe:

```
{
  "experimentTemplate": {
    "id": "ABCDE1fgHIJkLmNop",
    "description": "experimentTemplate",
    "targets": {
      "Instances-Target-1": {
        "resourceType": "aws:ec2:instance",
        "resourceArns": [
          "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
        ],
        "selectionMode": "ALL"
      }
    },
    "actions": {
      "reboot": {
        "actionId": "aws:ec2:reboot-instances",
        "description": "reboot",
        "parameters": {},
        "targets": {
          "Instances": "Instances-Target-1"
        }
      }
    },
    "stopConditions": [
      {
        "source": "aws:cloudwatch:alarm",
        "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
      }
    ],
    "creationTime": 1616434850.659,
    "lastUpdateTime": 1616434850.659,
    "roleArn": "arn:aws:iam::123456789012:role/myRole",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Erstellen einer Versuchsvorlage](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateExperimentTemplate](#) unter AWS CLI Befehlsreferenz.

delete-experiment-template

Das folgende Codebeispiel zeigt die Verwendung `delete-experiment-template`.

AWS CLI

Um eine Experimentvorlage zu löschen

Im folgenden `delete-experiment-template` Beispiel wird die angegebene Experimentvorlage gelöscht.

```
aws fis delete-experiment-template \  
  --id ABCDE1fgHIJkLmNop
```

Ausgabe:

```
{  
  "experimentTemplate": {  
    "id": "ABCDE1fgHIJkLmNop",  
    "description": "myExperimentTemplate",  
    "targets": {  
      "Instances-Target-1": {  
        "resourceType": "aws:ec2:instance",  
        "resourceArns": [  
          "arn:aws:ec2:us-west-2:123456789012:instance/  
i-12a3b4c56d78e9012"  
        ],  
        "selectionMode": "ALL"  
      }  
    },  
    "actions": {  
      "testaction": {  
        "actionId": "aws:ec2:stop-instances",  
        "parameters": {},  
        "targets": {  
          "Instances": "Instances-Target-1"  
        }  
      }  
    },  
    "stopConditions": [  
      {  
        "source": "none"  
      }  
    ]  
  }  
}
```

```
    ],
    "creationTime": 1616017191.124,
    "lastUpdateTime": 1616017859.607,
    "roleArn": "arn:aws:iam::123456789012:role/FISRole"
  }
}
```

Weitere Informationen finden Sie unter [Löschen einer Experimentvorlage](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteExperimentTemplate](#) unter AWS CLI Befehlsreferenz.

get-action

Das folgende Codebeispiel zeigt die Verwendung `get-action`.

AWS CLI

Um Details zur Aktion zu erhalten

Im folgenden `get-action` Beispiel werden die Details der angegebenen Aktion abgerufen.

```
aws fis get-action \
  --id aws:ec2:stop-instances
```

Ausgabe:

```
{
  "action": {
    "id": "aws:ec2:stop-instances",
    "description": "Stop the specified EC2 instances.",
    "parameters": {
      "startInstancesAfterDuration": {
        "description": "The time to wait before restarting the instances (ISO 8601 duration).",
        "required": false
      }
    },
    "targets": {
      "Instances": {
        "resourceType": "aws:ec2:instance"
      }
    }
  },
}
```

```

    "tags": {}
  }
}

```

Weitere Informationen finden Sie unter [Aktionen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [GetAction](#) in der AWS CLI Befehlsreferenz.

get-experiment-template

Das folgende Codebeispiel zeigt die Verwendung `get-experiment-template`.

AWS CLI

Um Details zur Experimentvorlage zu erhalten

Im folgenden `get-experiment-template` Beispiel werden die Details der angegebenen Experimentvorlage abgerufen.

```

aws fis get-experiment-template \
  --id ABCDE1fgHIJKLmNop

```

Ausgabe:

```

{
  "experimentTemplate": {
    "id": "ABCDE1fgHIJKLmNop",
    "description": "myExperimentTemplate",
    "targets": {
      "Instances-Target-1": {
        "resourceType": "aws:ec2:instance",
        "resourceArns": [
          "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"
        ],
        "selectionMode": "ALL"
      }
    },
    "actions": {
      "testaction": {
        "actionId": "aws:ec2:stop-instances",
        "parameters": {},

```



```
        "targets": {
            "Instances": "Instances-Target-1"
        }
    },
    "stopConditions": [
        {
            "source": "none"
        }
    ],
    "creationTime": 1616017191.124,
    "lastUpdateTime": 1616017331.51,
    "roleArn": "arn:aws:iam::123456789012:role/FISRole",
    "tags": {
        "key": "value"
    }
}
}
```

Weitere Informationen finden Sie unter [Experimentvorlagen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [GetExperimentTemplate](#) in der AWS CLI Befehlsreferenz.

get-experiment

Das folgende Codebeispiel zeigt die Verwendung `get-experiment`.

AWS CLI

Um Details zum Experiment zu erhalten

Im folgenden `get-experiment` Beispiel werden die Details des angegebenen Experiments abgerufen.

```
aws fis get-experiment \
  --id ABC12DeFGhI3jKLMNOP
```

Ausgabe:

```
{
  "experiment": {
    "id": "ABC12DeFGhI3jKLMNOP",
```

```
"experimentTemplateId": "ABCDE1fgHIJkLmNop",
"roleArn": "arn:aws:iam::123456789012:role/myRole",
"state": {
  "status": "completed",
  "reason": "Experiment completed."
},
"targets": {
  "Instances-Target-1": {
    "resourceType": "aws:ec2:instance",
    "resourceArns": [
      "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
    ],
    "selectionMode": "ALL"
  }
},
"actions": {
  "reboot": {
    "actionId": "aws:ec2:reboot-instances",
    "parameters": {},
    "targets": {
      "Instances": "Instances-Target-1"
    },
    "state": {
      "status": "completed",
      "reason": "Action was completed."
    }
  }
},
"stopConditions": [
  {
    "source": "none"
  }
],
"creationTime": 1616432509.662,
"startTime": 1616432509.962,
"endTime": 1616432522.307,
"tags": {}
}
```

Weitere Informationen finden Sie unter [Experimente für AWS FIS](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [GetExperiment](#) in der AWS CLI Befehlsreferenz.

list-actions

Das folgende Codebeispiel zeigt die Verwendung `list-actions`.

AWS CLI

Um Aktionen aufzulisten

Das folgende `list-actions` Beispiel listet die verfügbaren Aktionen auf.

```
aws fis list-actions
```

Ausgabe:

```
{
  "actions": [
    {
      "id": "aws:ec2:reboot-instances",
      "description": "Reboot the specified EC2 instances.",
      "targets": {
        "Instances": {
          "resourceType": "aws:ec2:instance"
        }
      },
      "tags": {}
    },
    {
      "id": "aws:ec2:stop-instances",
      "description": "Stop the specified EC2 instances.",
      "targets": {
        "Instances": {
          "resourceType": "aws:ec2:instance"
        }
      },
      "tags": {}
    },
    {
      "id": "aws:ec2:terminate-instances",
      "description": "Terminate the specified EC2 instances.",
      "targets": {
        "Instances": {
```

```

        "resourceType": "aws:ec2:instance"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:ecs:drain-container-instances",
    "description": "Drain percentage of underlying EC2 instances on an ECS
cluster.",
    "targets": {
      "Clusters": {
        "resourceType": "aws:ecs:cluster"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:eks:terminate-nodegroup-instances",
    "description": "Terminates a percentage of the underlying EC2 instances
in an EKS cluster.",
    "targets": {
      "Nodegroups": {
        "resourceType": "aws:eks:nodegroup"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:inject-api-internal-error",
    "description": "Cause an AWS service to return internal error responses
for specific callers and operations.",
    "targets": {
      "Roles": {
        "resourceType": "aws:iam:role"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:inject-api-throttle-error",
    "description": "Cause an AWS service to return throttled responses for
specific callers and operations.",
    "targets": {
      "Roles": {

```

```
        "resourceType": "aws:iam:role"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:inject-api-unavailable-error",
    "description": "Cause an AWS service to return unavailable error
responses for specific callers and operations.",
    "targets": {
      "Roles": {
        "resourceType": "aws:iam:role"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:wait",
    "description": "Wait for the specified duration. Stop condition
monitoring will continue during this time.",
    "tags": {}
  },
  {
    "id": "aws:rds:failover-db-cluster",
    "description": "Failover a DB Cluster to one of the replicas.",
    "targets": {
      "Clusters": {
        "resourceType": "aws:rds:cluster"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:rds:reboot-db-instances",
    "description": "Reboot the specified DB instances.",
    "targets": {
      "DBInstances": {
        "resourceType": "aws:rds:db"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:ssm:send-command",
```

```
        "description": "Run the specified SSM document.",
        "targets": {
            "Instances": {
                "resourceType": "aws:ec2:instance"
            }
        },
        "tags": {}
    }
]
}
```

Weitere Informationen finden Sie unter [Aktionen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [ListActions](#) in der AWS CLI Befehlsreferenz.

list-experiment-templates

Das folgende Codebeispiel zeigt die Verwendung `list-experiment-templates`.

AWS CLI

Um Versuchsvorlagen aufzulisten

Das folgende `list-experiment-templates` Beispiel listet die Versuchsvorlagen in Ihrem AWS Konto auf.

```
aws fis list-experiment-templates
```

Ausgabe:

```
{
  "experimentTemplates": [
    {
      "id": "ABCDE1fgHIJkLmNop",
      "description": "myExperimentTemplate",
      "creationTime": 1616017191.124,
      "lastUpdateTime": 1616017191.124,
      "tags": {
        "key": "value"
      }
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Experimentvorlagen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [ListExperimentTemplates](#) in der AWS CLI Befehlsreferenz.

list-experiments

Das folgende Codebeispiel zeigt die Verwendung `list-experiments`.

AWS CLI

Um Experimente aufzulisten

Das folgende `list-experiments` Beispiel listet die Experimente in Ihrem AWS Konto auf.

```
aws fis list-experiments
```

Ausgabe:

```
{
  "experiments": [
    {
      "id": "ABCdeF1GHijKLM23N0",
      "experimentTemplateId": "ABCDE1fgHIJkLmNop",
      "state": {
        "status": "running",
        "reason": "Experiment is running."
      },
      "creationTime": 1616017341.197,
      "tags": {
        "key": "value"
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Experimente](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [ListExperiments](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für die angegebene Ressource auf.

```
aws fis list-tags-for-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Ausgabe:

```
{  
  "tags": {  
    "key1": "value1",  
    "key2": "value2"  
  }  
}
```

Weitere Informationen finden Sie unter [Taggen Ihrer AWS FIS Ressourcen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-experiment

Das folgende Codebeispiel zeigt die Verwendung `start-experiment`.

AWS CLI

Um ein Experiment zu starten

Im folgenden `start-experiment` Beispiel wird das angegebene Experiment gestartet.

```
aws fis start-experiment \  
  --experiment-name my-experiment
```



```
--experiment-template-id ABCDE1fgHIJkLmNop
```

Ausgabe:

```
{
  "experiment": {
    "id": "ABC12DeFGhI3jKLMNOP",
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",
    "roleArn": "arn:aws:iam::123456789012:role/myRole",
    "state": {
      "status": "initiating",
      "reason": "Experiment is initiating."
    },
    "targets": {
      "Instances-Target-1": {
        "resourceType": "aws:ec2:instance",
        "resourceArns": [
          "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
        ],
        "selectionMode": "ALL"
      }
    },
    "actions": {
      "reboot": {
        "actionId": "aws:ec2:reboot-instances",
        "parameters": {},
        "targets": {
          "Instances": "Instances-Target-1"
        },
        "state": {
          "status": "pending",
          "reason": "Initial state"
        }
      }
    },
    "stopConditions": [
      {
        "source": "none"
      }
    ],
    "creationTime": 1616432464.025,
    "startTime": 1616432464.374,
```

```
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Experimente für AWS FIS](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [StartExperiment](#) in der AWS CLI Befehlsreferenz.

stop-experiment

Das folgende Codebeispiel zeigt die Verwendung `stop-experiment`.

AWS CLI

Um ein Experiment zu beenden

Im folgenden `stop-experiment` Beispiel wird die Ausführung des angegebenen Experiments beendet.

```
aws fis stop-experiment \
  --id ABC12DeFGhI3jKLMNOP
```

Ausgabe:

```
{
  "experiment": {
    "id": "ABC12DeFGhI3jKLMNOP",
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",
    "roleArn": "arn:aws:iam::123456789012:role/myRole",
    "state": {
      "status": "stopping",
      "reason": "Stopping Experiment."
    },
    "targets": {
      "Instances-Target-1": {
        "resourceType": "aws:ec2:instance",
        "resourceArns": [
          "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"
        ]
      }
    }
  }
}
```

```
        "selectionMode": "ALL"
      }
    },
    "actions": {
      "reboot": {
        "actionId": "aws:ec2:reboot-instances",
        "parameters": {},
        "targets": {
          "Instances": "Instances-Target-1"
        },
        "startAfter": [
          "wait"
        ],
        "state": {
          "status": "pending",
          "reason": "Initial state."
        }
      },
      "wait": {
        "actionId": "aws:fis:wait",
        "parameters": {
          "duration": "PT5M"
        },
        "state": {
          "status": "running",
          "reason": ""
        }
      }
    },
    "stopConditions": [
      {
        "source": "none"
      }
    ],
    "creationTime": 1616432680.927,
    "startTime": 1616432681.177,
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Experimente für AWS FIS](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [StopExperiment](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird die angegebene Ressource markiert.

```
aws fis tag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP \  
  --tags key1=value1,key2=value2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Taggen Ihrer AWS FIS Ressourcen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung einer Ressource aufzuheben

Im folgenden `untag-resource` Beispiel werden die Tags aus der angegebenen Ressource entfernt.

```
aws fis untag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Taggen Ihrer AWS FIS Ressourcen](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-experiment-template

Das folgende Codebeispiel zeigt die Verwendung `update-experiment-template`.

AWS CLI

Um eine Experimentvorlage zu aktualisieren

Im folgenden `update-experiment-template` Beispiel wird die Beschreibung der angegebenen Experimentvorlage aktualisiert.

```
aws fis update-experiment-template \  
  --id ABCDE1fgHIJkLmNop \  
  ---description myExperimentTemplate
```

Ausgabe:

```
{  
  "experimentTemplate": {  
    "id": "ABCDE1fgHIJkLmNop",  
    "description": "myExperimentTemplate",  
    "targets": {  
      "Instances-Target-1": {  
        "resourceType": "aws:ec2:instance",  
        "resourceArns": [  
          "arn:aws:ec2:us-west-2:123456789012:instance/  
i-12a3b4c56d78e9012"  
        ],  
        "selectionMode": "ALL"  
      }  
    },  
    "actions": {  
      "testaction": {  
        "actionId": "aws:ec2:stop-instances",  
        "parameters": {},  
        "targets": {  
          "Instances": "Instances-Target-1"  
        }  
      }  
    },  
    "stopConditions": [  
      {  
        "source": "none"  
      }  
    ]  
  }  
}
```

```
    }
  ],
  "creationTime": 1616017191.124,
  "lastUpdateTime": 1616017859.607,
  "roleArn": "arn:aws:iam::123456789012:role/FISRole",
  "tags": {
    "key": "value"
  }
}
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer Experimentvorlage](#) im AWS Fault Injection Simulator-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateExperimentTemplate](#) in der AWS CLI Befehlsreferenz.

GameLift Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden GameLift.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-build

Das folgende Codebeispiel zeigt die Verwendung `create-build`.

AWS CLI

Beispiel 1: Um einen Spiel-Build aus Dateien in einem S3-Bucket zu erstellen

Im folgenden `create-build` Beispiel wird eine benutzerdefinierte Spiele-Build-Ressource erstellt. Es verwendet komprimierte Dateien, die an einem S3-Speicherort in einem AWS Konto gespeichert sind, das Sie kontrollieren. In diesem Beispiel wird davon ausgegangen, dass Sie bereits eine IAM Rolle erstellt haben, die Amazon die GameLift Erlaubnis erteilt, auf den S3-Standort zuzugreifen. Da die Anfrage kein Betriebssystem spezifiziert, ist die neue Build-Ressource standardmäßig auf `WINDOWS_2012` eingestellt.

```
aws gamelift create-build \  
  --storage-location file://storage-loc.json \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678
```

Inhalt von `storage-loc.json`:

```
{  
  "Bucket": "MegaFrogRaceServer_NA_build_files"  
  "Key": "MegaFrogRaceServer_build_123.zip"  
  "RoleArn": "arn:aws:iam::123456789012:role/gamelift"  
}
```

Ausgabe:

```
{  
  "Build": {  
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "CreationTime": 1496708916.18,  
    "Name": "MegaFrogRaceServer.NA",  
    "OperatingSystem": "WINDOWS_2012",  
    "SizeOnDisk": 479303,  
    "Status": "INITIALIZED",  
    "Version": "12345.678"  
  },  
  "StorageLocation": {  
    "Bucket": "MegaFrogRaceServer_NA_build_files",  
    "Key": "MegaFrogRaceServer_build_123.zip"  
  }  
}
```

Beispiel 2: Um eine Spiele-Build-Ressource für das manuelle Hochladen von Dateien zu erstellen GameLift

Das folgende `create-build` Beispiel erstellt eine neue Build-Ressource. Es erhält auch einen Speicherort und temporäre Anmeldeinformationen, mit denen Sie Ihren Spiel-Build manuell an den GameLift Speicherort in Amazon S3 hochladen können. Sobald Sie Ihren Build erfolgreich hochgeladen haben, validiert der GameLift Service den Build und aktualisiert den Status des neuen Builds.

```
aws gamelift create-build \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678 \  
  --operating-system AMAZON_LINUX
```

Ausgabe:

```
{  
  "Build": {  
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "CreationTime": 1496708916.18,  
    "Name": "MegaFrogRaceServer.NA",  
    "OperatingSystem": "AMAZON_LINUX",  
    "SizeOnDisk": 0,  
    "Status": "INITIALIZED",  
    "Version": "12345.678"  
  },  
  "StorageLocation": {  
    "Bucket": "gamelift-builds-us-west-2",  
    "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  },  
  "UploadCredentials": {  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",  
    "SessionToken": "AgoGb3JpZ2luENZ...EXAMPLETOKEN=="  
  }  
}
```

Weitere Informationen finden Sie unter [Hochladen eines benutzerdefinierten Server-Builds auf GameLift](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [CreateBuild](#) in der AWS CLI Befehlsreferenz.

create-fleet

Das folgende Codebeispiel zeigt die Verwendung `create-fleet`.

AWS CLI

Beispiel 1: Um eine einfache Linux-Flotte zu erstellen

Im folgenden `create-fleet` Beispiel wird eine minimal konfigurierte Flotte von On-Demand-Linux-Instances erstellt, um einen benutzerdefinierten Server-Build zu hosten. Sie können die Konfiguration abschließen, indem Sie `update-fleet`

```
aws gamelift create-fleet \
  --name MegaFrogRaceServer.NA.v2 \
  --description 'Hosts for v2 North America' \
  --build-id build-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
  --certificate-configuration 'CertificateType=GENERATED' \
  --ec2-instance-type c4.large \
  --fleet-type ON_DEMAND \
  --runtime-configuration 'ServerProcesses=[{LaunchPath=/local/game/release-na/MegaFrogRace_Server.exe, ConcurrentExecutions=1}]'
```

Ausgabe:

```
{
  "FleetAttributes": {
    "BuildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
    "CertificateConfiguration": {
      "CertificateType": "GENERATED"
    },
    "CreationTime": 1496365885.44,
    "Description": "Hosts for v2 North America",
    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetType": "ON_DEMAND",
    "InstanceType": "c4.large",
    "MetricGroups": ["default"],
    "Name": "MegaFrogRace.NA.v2",
    "NewGameSessionProtectionPolicy": "NoProtection",
```

```

    "OperatingSystem": "AMAZON_LINUX",
    "ServerLaunchPath": "/local/game/release-na/MegaFrogRace_Server.exe",
    "Status": "NEW"
  }
}

```

Beispiel 2: So erstellen Sie eine Windows-Standardflotte

Im folgenden `create-fleet` Beispiel wird eine minimal konfigurierte Flotte von Windows-Spot-Instances erstellt, um einen benutzerdefinierten Server-Build zu hosten. Sie können die Konfiguration abschließen, indem Sie `update-fleet`

```

aws gamelift create-fleet \
  --name MegaFrogRace.NA.v2 \
  --description 'Hosts for v2 North America' \
  --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff \
  --certificate-configuration 'CertificateType=GENERATED' \
  --ec2-instance-type c4.large \
  --fleet-type SPOT \
  --runtime-configuration 'ServerProcesses=[{LaunchPath=C:\game
\Bin64.Release.Dedicated\MegaFrogRace_Server.exe, ConcurrentExecutions=1}]'

```

Ausgabe:

```

{
  "FleetAttributes": {
    "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
    "CertificateConfiguration": {
      "CertificateType": "GENERATED"
    },
    "CreationTime": 1496365885.44,
    "Description": "Hosts for v2 North America",
    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetType": "SPOT",
    "InstanceType": "c4.large",
    "MetricGroups": ["default"],
    "Name": "MegaFrogRace.NA.v2",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "WINDOWS_2012",
    "ServerLaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",

```

```

    "Status": "NEW"
  }
}

```

Beispiel 3: Um eine vollständig konfigurierte Flotte zu erstellen

Im folgenden `create-fleet` Beispiel wird eine Flotte von Windows-Spot-Instances für einen benutzerdefinierten Serverbuild erstellt, wobei die am häufigsten verwendeten Konfigurationseinstellungen bereitgestellt werden.

```

aws gamelift create-fleet \
  --name MegaFrogRace.NA.v2 \
  --description 'Hosts for v2 North America' \
  --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff \
  --certificate-configuration 'CertificateType=GENERATED' \
  --ec2-instance-type c4.large \
  --ec2-inbound-permissions
'FromPort=33435,ToPort=33435,IpRange=10.24.34.0/23,Protocol=UDP' \
  --fleet-type SPOT \
  --new-game-session-protection-policy FullProtection \
  --runtime-configuration file://runtime-config.json \
  --metric-groups default \
  --instance-role-arn 'arn:aws:iam::444455556666:role/GameLiftS3Access'

```

Inhalt von `runtime-config.json`:

```

GameSessionActivationTimeoutSeconds=300,
MaxConcurrentGameSessionActivations=2,
ServerProcesses=[
  {LaunchPath=C:\game\Bin64.Release.Dedicated\MegaFrogRace_Server.exe,Parameters=-
debug,ConcurrentExecutions=1},
  {LaunchPath=C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe,ConcurrentExecutions=1}]

```

Ausgabe:

```

{
  "FleetAttributes": {
    "InstanceRoleArn": "arn:aws:iam::444455556666:role/GameLiftS3Access",
    "Status": "NEW",
    "InstanceType": "c4.large",

```

```

    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "Description": "Hosts for v2 North America",
    "FleetType": "SPOT",
    "OperatingSystem": "WINDOWS_2012",
    "Name": "MegaFrogRace.NA.v2",
    "CreationTime": 1569309011.11,
    "MetricGroups": [
        "default"
    ],
    "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
    "ServerLaunchParameters": "abc",
    "ServerLaunchPath": "C:\\game\\Bin64.Release.Dedicated\\
\MegaFrogRace_Server.exe",
    "NewGameSessionProtectionPolicy": "FullProtection",
    "CertificateConfiguration": {
        "CertificateType": "GENERATED"
    }
}
}
}

```

Beispiel 4: So erstellen Sie eine Realtime Server-Flotte

Im folgenden `create-fleet` Beispiel wird eine Flotte von Spot-Instances mit einem Echtzeit-Konfigurationskript erstellt, das auf Amazon GameLift hochgeladen wurde. Alle Realtime-Server werden auf Linux-Computern bereitgestellt. Gehen Sie für dieses Beispiel davon aus, dass das hochgeladene Echtzeit-Skript mehrere Skriptdateien enthält, wobei die `Init()` Funktion, die sich in der Skriptdatei befindet, aufgerufen wird. `MainScript.js` Wie gezeigt, wird diese Datei in der Laufzeitkonfiguration als Startskript identifiziert.

```

aws gamelift create-fleet \
  --name MegaFrogRace.NA.realtime \
  --description 'Mega Frog Race Realtime fleet' \
  --script-id script-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
  --ec2-instance-type c4.large \
  --fleet-type SPOT \
  --certificate-configuration 'CertificateType=GENERATED' --runtime-configuration
'ServerProcesses=[{LaunchPath=/local/game/MainScript.js,Parameters=#map
Winter444,ConcurrentExecutions=5}]'

```

Ausgabe:

```
{
  "FleetAttributes": {
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "Status": "NEW",
    "CreationTime": 1569310745.212,
    "InstanceType": "c4.large",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "CertificateConfiguration": {
      "CertificateType": "GENERATED"
    },
    "Name": "MegaFrogRace.NA.realtime",
    "ScriptId": "script-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetType": "SPOT",
    "MetricGroups": [
      "default"
    ],
    "Description": "Mega Frog Race Realtime fleet",
    "OperatingSystem": "AMAZON_LINUX"
  }
}
```

- API-Einheiten finden Sie [CreateFleet](#) in der AWS CLI Befehlsreferenz.

create-game-session-queue

Das folgende Codebeispiel zeigt die Verwendung `create-game-session-queue`.

AWS CLI

Beispiel 1: Um eine Warteschlange für eine geordnete Spielsitzung einzurichten

Im folgenden `create-game-session-queue` Beispiel wird eine neue Warteschlange für Spielsitzungen mit Zielen in zwei Regionen erstellt. Außerdem wird die Warteschlange so konfiguriert, dass die Spielsitzung nach 10 Minuten Wartezeit auf die Platzierung ein Timeout anfordert. Da keine Latenzrichtlinien definiert sind, wird GameLift versucht, alle Spielsitzungen mit dem ersten aufgelisteten Ziel zu platzieren.

```
aws gamelift create-game-session-queue \
  --name MegaFrogRaceServer-NA \
  --destinations file://destinations.json \
```

```
--timeout-in-seconds 600
```

Inhalt von `destinations.json`:

```
{
  "Destinations": [
    {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },
    {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }
  ]
}
```

Ausgabe:

```
{
  "GameSessionQueues": [
    {
      "Name": "MegaFrogRaceServer-NA",
      "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:123456789012:gamesessionqueue/MegaFrogRaceServer-NA",
      "TimeoutInSeconds": 600,
      "Destinations": [
        {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},
        {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}
      ]
    }
  ]
}
```

Beispiel 2: Um eine Warteschlange für Spielsitzungen mit Latenzrichtlinien für Spieler einzurichten

Im folgenden `create-game-session-queue` Beispiel wird eine neue Warteschlange für Spielsitzungen mit zwei Latenzrichtlinien für Spieler erstellt. Die erste Richtlinie legt eine Latenzgrenze von 100 ms fest, die während der ersten Minute eines Platzierungsversuchs für eine Spielsitzung durchgesetzt wird. Die zweite Richtlinie erhöht die Latenzbegrenzung auf 200 ms, bis das Timeout der Platzierungsanfrage bei 3 Minuten liegt.

```
aws gamelift create-game-session-queue \  
  --name MegaFrogRaceServer-NA \  
  --latency-requirements [ ... ]
```

```
--destinations file://destinations.json \  
--player-latency-policies file://latency-policies.json \  
--timeout-in-seconds 180
```

Inhalt von `destinations.json`:

```
{  
  "Destinations": [  
    { "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },  
    { "DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }  
  ]  
}
```

Inhalt von `latency-policies.json`:

```
{  
  "PlayerLatencyPolicies": [  
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},  
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":  
60}  
  ]  
}
```

Ausgabe:

```
{  
  "GameSessionQueue": {  
    "Name": "MegaFrogRaceServer-NA",  
    "GameSessionQueueArn": "arn:aws:gamelift:us-west-2:111122223333:gamesessionqueue/MegaFrogRaceServer-NA",  
    "TimeoutInSeconds": 600,  
    "PlayerLatencyPolicies": [  
      {  
        "MaximumIndividualPlayerLatencyMilliseconds": 100,  
        "PolicyDurationSeconds": 60  
      },  
      {  
        "MaximumIndividualPlayerLatencyMilliseconds": 200  
      }  
    ]  
  }  
}
```

```
    "Destinations": [  
      {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},  
      {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}  
    ],  
  }  
}
```

Weitere Informationen finden Sie unter [Create a Queue](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [CreateGameSessionQueue](#) in der AWS CLI Befehlsreferenz.

delete-build

Das folgende Codebeispiel zeigt die Verwendung `delete-build`.

AWS CLI

Um einen benutzerdefinierten Spiel-Build zu löschen

Im folgenden `delete-build` Beispiel wird ein Build aus Ihrem GameLift Amazon-Konto entfernt. Nachdem der Build gelöscht wurde, können Sie ihn nicht mehr zum Erstellen neuer Flotten verwenden. Dieser Vorgang kann nicht rückgängig gemacht werden.

```
aws gamelift delete-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBuild](#) in der AWS CLI Befehlsreferenz.

delete-fleet

Das folgende Codebeispiel zeigt die Verwendung `delete-fleet`.

AWS CLI

Um eine Flotte zu löschen, die nicht mehr verwendet wird

Im folgenden `delete-fleet` Beispiel wird eine Flotte entfernt, die auf null Instanzen herunterskaliert wurde. Wenn die Flottenkapazität größer als Null ist, schlägt die Anforderung mit einem HTTP 400-Fehler fehl.


```
aws gamelift delete-fleet \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Manage GameLift Fleets](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DeleteFleet](#) in der AWS CLI Befehlsreferenz.

delete-game-session-queue

Das folgende Codebeispiel zeigt die Verwendung `delete-game-session-queue`.

AWS CLI

Um eine Warteschlange für eine Spielsitzung zu löschen

Im folgenden `delete-game-session-queue` Beispiel wird eine angegebene Warteschlange für Spielsitzungen gelöscht.

```
aws gamelift delete-game-session-queue \  
  --name MegaFrogRace-NA
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteGameSessionQueue](#) in der AWS CLI Befehlsreferenz.

describe-build

Das folgende Codebeispiel zeigt die Verwendung `describe-build`.

AWS CLI

Um Informationen zu einem benutzerdefinierten Spiel-Build zu erhalten

Im folgenden `describe-build` Beispiel werden Eigenschaften für eine Build-Ressource für einen Spieleserver abgerufen.

```
aws gamelift describe-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "Build": {
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "CreationTime": 1496708916.18,
    "Name": "My_Game_Server_Build_One",
    "OperatingSystem": "AMAZON_LINUX",
    "SizeOnDisk": 1304924,
    "Status": "READY",
    "Version": "12345.678"
  }
}
```

Weitere Informationen finden Sie unter [Hochladen eines benutzerdefinierten Server-Builds auf GameLift](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DescribeBuild](#) in der AWS CLI Befehlsreferenz.

describe-ec2-instance-limits

Das folgende Codebeispiel zeigt die Verwendung `describe-ec2-instance-limits`.

AWS CLI

Um Service-Limits für einen EC2 Instance-Typ abzurufen

Im folgenden `describe-ec2-instance-limits` Beispiel werden die maximal zulässigen Instanzen und die aktuell verwendeten Instanzen für den angegebenen EC2 Instanztyp in der aktuellen Region angezeigt. Das Ergebnis zeigt, dass nur fünf der zwanzig zulässigen Instanzen verwendet werden.

```
aws gamelift describe-ec2-instance-limits \
  --ec2-instance-type m5.large
```

Ausgabe:

```
{
  "EC2InstanceLimits": [
```

```
{
  "EC2InstanceType": "m5.large",
  "CurrentInstances": 5,
  "InstanceLimit": 20
}
```

Weitere Informationen finden [Sie unter Choose Computing Resources](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie unter [DescribeEc2 InstanceLimits](#) in der AWS CLI Befehlsreferenz.

describe-fleet-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet-attributes`.

AWS CLI

Beispiel 1: Um Attribute für eine Liste von Flotten anzuzeigen

Im folgenden `describe-fleet-attributes` Beispiel werden Flottenattribute für zwei angegebene Flotten abgerufen. Wie gezeigt, werden die angeforderten Flotten mit demselben Build bereitgestellt, eine für On-Demand-Instances und eine für Spot-Instances, mit einigen geringfügigen Konfigurationsunterschieden.

```
aws gamelift describe-fleet-attributes \
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Ausgabe:

```
{
  "FleetAttributes": [
    {
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "FleetType": "ON_DEMAND",
      "InstanceType": "c4.large",
      "Description": "On-demand hosts for v2 North America",
      "Name": "MegaFrogRaceServer.NA.v2-od",
    }
  ]
}
```

```

    "CreationTime": 1568836191.995,
    "Status": "ACTIVE",
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
    "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
    "ServerLaunchParameters": "+gamelift_start_server",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "WINDOWS_2012",
    "MetricGroups": [
        "default"
    ],
    "CertificateConfiguration": {
        "CertificateType": "DISABLED"
    }
},
{
    "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222",
    "FleetType": "SPOT",
    "InstanceType": "c4.large",
    "Description": "On-demand hosts for v2 North America",
    "Name": "MegaFrogRaceServer.NA.v2-spot",
    "CreationTime": 1568838275.379,
    "Status": "ACTIVATING",
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
    "ServerLaunchPath": "C:\\game\\MegaFrogRace_Server.exe",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "WINDOWS_2012",
    "MetricGroups": [
        "default"
    ],
    "CertificateConfiguration": {
        "CertificateType": "GENERATED"
    }
}
]
}

```

Beispiel 2: Um Attribute für alle Flotten anzufordern

Im Folgenden werden Flottenattribute für alle Flotten mit beliebigem Status `describe-fleet-attributes` zurückgegeben. Dieses Beispiel veranschaulicht die Verwendung von Paginierungsparametern, um jeweils eine Flotte zurückzugeben.

```
aws gamelift describe-fleet-attributes \
  --limit 1
```

Ausgabe:

```
{
  "FleetAttributes": [
    {
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "FleetType": "SPOT",
      "InstanceType": "c4.large",
      "Description": "On-demand hosts for v2 North America",
      "Name": "MegaFrogRaceServer.NA.v2-spot",
      "CreationTime": 1568838275.379,
      "Status": "ACTIVATING",
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "ServerLaunchPath": "C:\\\\game\\\\MegaFrogRace_Server.exe",
      "NewGameSessionProtectionPolicy": "NoProtection",
      "OperatingSystem": "WINDOWS_2012",
      "MetricGroups": [
        "default"
      ],
      "CertificateConfiguration": {
        "CertificateType": "GENERATED"
      }
    }
  ],
  "NextToken":
  "eyJhd3NBdW50SWQlOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"
}
```

Die Ausgabe enthält einen `NextToken` Wert, den Sie verwenden können, wenn Sie den Befehl ein zweites Mal aufrufen. Übergeben Sie den Wert an den `--next-token` Parameter, um

anzugeben, wo die Ausgabe abgerufen werden soll. Der folgende Befehl gibt das zweite Ergebnis in der Ausgabe zurück.

```
aws gamelift describe-fleet-attributes \  
  --limit 1 \  
  --next-  
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Wiederholen Sie den Vorgang, bis die Antwort keinen NextToken Wert mehr enthält.

Weitere Informationen finden Sie unter [Setting Up GameLift Flotten](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DescribeFleetAttributes](#) in der AWS CLI Befehlsreferenz.

describe-fleet-capacity

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet-capacity`.

AWS CLI

Um den Kapazitätsstatus für eine Liste von Flotten anzuzeigen

Im folgenden `describe-fleet-capacity` Beispiel wird die aktuelle Kapazität für zwei angegebene Flotten abgerufen.

```
aws gamelift describe-fleet-capacity \  
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Ausgabe:

```
{  
  "FleetCapacity": [  
    {  
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "InstanceType": "c5.large",  
      "InstanceCounts": {  
        "DESIRED": 10,  
        "MINIMUM": 1,  
        "MAXIMUM": 20,  
        "PENDING": 0,  
      }  
    }  
  ]  
}
```

```
        "ACTIVE": 10,
        "IDLE": 3,
        "TERMINATING": 0
    }
},
{
    "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "InstanceType": "c5.large",
    "InstanceCounts": {
        "DESIRED": 13,
        "MINIMUM": 1,
        "MAXIMUM": 20,
        "PENDING": 0,
        "ACTIVE": 15,
        "IDLE": 2,
        "TERMINATING": 2
    }
}
]
}
```

Weitere Informationen finden Sie unter [GameLift Metrics for Fleets](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DescribeFleetCapacity](#) in der AWS CLI Befehlsreferenz.

describe-fleet-events

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet-events`.

AWS CLI

Um Ereignisse für einen bestimmten Zeitraum anzufordern

Im folgenden `describe-fleet-events` Beispiel werden Details zu allen flottenbezogenen Ereignissen angezeigt, die während des angegebenen Zeitraums aufgetreten sind.

```
aws gamelift describe-fleet-events \  
  --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111 \  
  --start-time 1579647600 \  
  --end-time 1579649400 \  
  --output text
```

```
--limit 5
```

Ausgabe:

```
{
  "Events": [
    {
      "EventId": "a37b6892-5d07-4d3b-8b47-80244ecf66b9",
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "EventCode": "FLEET_STATE_ACTIVE",
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
state to ACTIVE",
      "EventTime": 1579649342.191
    },
    {
      "EventId": "67da4ec9-92a3-4d95-886a-5d6772c24063",
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "EventCode": "FLEET_STATE_ACTIVATING",
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
state to ACTIVATING",
      "EventTime": 1579649321.427
    },
    {
      "EventId": "23813a46-a9e6-4a53-8847-f12e6a8381ac",
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "EventCode": "FLEET_STATE_BUILDING",
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
state to BUILDING",
      "EventTime": 1579649321.243
    },
    {
      "EventId": "3bf217d0-1d44-42f9-9202-433ed475d2e8",
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "EventCode": "FLEET_STATE_VALIDATING",
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
state to VALIDATING",
      "EventTime": 1579649197.449
    },
    {
      "EventId": "2ecd0130-5986-44eb-99a7-62df27741084",
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "EventCode": "FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND",
      "Message": "Failed to find a valid path",
    }
  ]
}
```


Im folgenden `describe-fleet-port-settings` Beispiel werden Verbindungseinstellungen für eine angegebene Flotte abgerufen.

```
aws gamelift describe-fleet-port-settings \  
  --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111
```

Ausgabe:

```
{  
  "InboundPermissions": [  
    {  
      "FromPort": 33400,  
      "ToPort": 33500,  
      "IpRange": "0.0.0.0/0",  
      "Protocol": "UDP"  
    },  
    {  
      "FromPort": 1900,  
      "ToPort": 2000,  
      "IpRange": "0.0.0.0/0",  
      "Protocol": "TCP"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Setting Up GameLift Flotten](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DescribeFleetPortSettings](#) in der AWS CLI Befehlsreferenz.

describe-fleet-utilization

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet-utilization`.

AWS CLI

Beispiel 1: Um Nutzungsdaten für eine Liste von Flotten anzuzeigen

Im folgenden `describe-fleet-utilization` Beispiel werden aktuelle Nutzungsinformationen für eine angegebene Flotte abgerufen.

```
aws gamelift describe-fleet-utilization \  
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111
```

Ausgabe:

```
{  
  "FleetUtilization": [  
    {  
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ActiveServerProcessCount": 100,  
      "ActiveGameSessionCount": 62,  
      "CurrentPlayerSessionCount": 329,  
      "MaximumPlayerSessionCount": 1000  
    }  
  ]  
}
```

Beispiel 2: Um Nutzungsdaten für alle Flotten anzufordern

Im Folgenden werden Flottennutzungsdaten für alle Flotten mit beliebigem Status `describe-fleet-utilization` zurückgegeben. In diesem Beispiel werden Paginierungsparameter verwendet, um Daten für zwei Flotten gleichzeitig zurückzugeben.

```
aws gamelift describe-fleet-utilization \  
  --limit 2
```

Ausgabe:

```
{  
  "FleetUtilization": [  
    {  
      "FleetId": "fleet-1111aaaa-22bb-33cc-44dd-5555eeee66ff",  
      "ActiveServerProcessCount": 100,  
      "ActiveGameSessionCount": 13,  
      "CurrentPlayerSessionCount": 98,  
      "MaximumPlayerSessionCount": 1000  
    },  
    {  
      "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",  
      "ActiveServerProcessCount": 100,  
      "ActiveGameSessionCount": 62,  
      "CurrentPlayerSessionCount": 100,  
      "MaximumPlayerSessionCount": 1000  
    }  
  ]  
}
```

```

        "CurrentPlayerSessionCount": 329,
        "MaximumPlayerSessionCount": 1000
    }
],
"NextToken":
"eyJhd3NBWY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}

```

Rufen Sie den Befehl ein zweites Mal auf und übergeben Sie den NextToken Wert als Argument an den `--next-token` Parameter, um die nächsten beiden Ergebnisse zu sehen.

```

aws gamelift describe-fleet-utilization \
  --limit 2 \
  --next-
token eyJhd3NBWY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS

```

Wiederholen Sie den Vorgang, bis die Antwort keinen NextToken Wert mehr in der Ausgabe enthält.

Weitere Informationen finden Sie unter [GameLift Metrics for Fleets](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DescribeFleetUtilization](#) in der AWS CLI Befehlsreferenz.

describe-game-session-queues

Das folgende Codebeispiel zeigt die Verwendung `describe-game-session-queues`.

AWS CLI

Um Warteschlangen für Spielsitzungen anzuzeigen

Im folgenden `describe-game-session-queues` Beispiel werden Eigenschaften für zwei angegebene Warteschlangen abgerufen.

```

aws gamelift describe-game-session-queues \
  --names MegaFrogRace-NA MegaFrogRace-EU

```

Ausgabe:

```

{
  "GameSessionQueues": [{

```

```
    "Destinations": [{
      "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    {
      "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    }
  ],
  "Name": "MegaFrogRace-NA",
  "TimeoutInSeconds": 600,
  "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/
MegaFrogRace-NA",
  "PlayerLatencyPolicies": [{
    "MaximumIndividualPlayerLatencyMilliseconds": 200
  },
  {
    "MaximumIndividualPlayerLatencyMilliseconds": 100,
    "PolicyDurationSeconds": 60
  }
  ],
  "FilterConfiguration": {
    "AllowedLocations": ["us-west-2", "ap-south-1", "us-east-1"]
  },
  "PriorityConfiguration": {
    "PriorityOrder": ["LOCATION", "FLEET_TYPE", "DESTINATION"],
    "LocationOrder": ["us-west-2", "ap-south-1", "us-east-1"]
  }
},
{
  "Destinations": [{
    "DestinationArn": "arn:aws:gamelift:eu-west-3::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  }],
  "Name": "MegaFrogRace-EU",
  "TimeoutInSeconds": 600,
  "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/
MegaFrogRace-EU"
}
]
```

Weitere Informationen finden Sie unter [Using Multi-Region-Warteschlangen](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [DescribeGameSessionQueues](#) in der AWS CLI Befehlsreferenz.

describe-runtime-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-runtime-configuration`.

AWS CLI

Um die Laufzeitkonfiguration für eine Flotte anzufordern

Im folgenden `describe-runtime-configuration` Beispiel werden Details zur aktuellen Laufzeitkonfiguration für eine angegebene Flotte abgerufen.

```
aws gamelift describe-runtime-configuration \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "RuntimeConfiguration": {  
    "ServerProcesses": [  
      {  
        "LaunchPath": "C:\\game\\Bin64.Release.Dedicated  
\\MegaFrogRace_Server.exe",  
        "Parameters": "+gamelift_start_server",  
        "ConcurrentExecutions": 3  
      },  
      {  
        "LaunchPath": "C:\\game\\Bin64.Release.Dedicated  
\\MegaFrogRace_Server.exe",  
        "Parameters": "+gamelift_start_server +debug",  
        "ConcurrentExecutions": 1  
      }  
    ],  
    "MaxConcurrentGameSessionActivations": 2147483647,  
    "GameSessionActivationTimeoutSeconds": 300  
  }  
}
```



```

        "CreationTime": 1495528748.555,
        "Name": "My_Game_Server_Build_Two",
        "OperatingSystem": "AMAZON_LINUX_2",
        "SizeOnDisk": 8567781,
        "Status": "FAILED",
        "Version": "23456.789"
    }
],
"NextToken":
"eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}

```

Sie können den Befehl dann erneut mit dem folgenden `--next-token` Parameter aufrufen, um die nächsten beiden Builds zu sehen.

```

aws gamelift list-builds \
  --limit 2
  --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS

```

Wiederholen Sie den Vorgang, bis die Antwort keinen `NextToken` Wert mehr enthält.

Beispiel 2: Um eine Liste von benutzerdefinierten Spiel-Builds abzurufen, die den Status „Fehler“ haben

Im folgenden `list-builds` Beispiel werden Eigenschaften für alle Spielserver-Builds in der aktuellen Region abgerufen, die derzeit den Status `FAILED` haben.

```

aws gamelift list-builds \
  --status FAILED

```

Ausgabe:

```

{
  "Builds": [
    {
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "CreationTime": 1495528748.555,
      "Name": "My_Game_Server_Build_Two",
      "OperatingSystem": "AMAZON_LINUX_2",

```



```

        "SizeOnDisk": 8567781,
        "Status": "FAILED",
        "Version": "23456.789"
    }
]
}

```

- API-Einheiten finden Sie [ListBuilds](#) in der AWS CLI Befehlsreferenz.

list-fleets

Das folgende Codebeispiel zeigt die Verwendung `list-fleets`.

AWS CLI

Beispiel 1: Um eine Liste aller Flotten in einer Region abzurufen

Im folgenden `list-fleets` Beispiel wird die Flotte IDs aller Flotten in der aktuellen Region angezeigt. In diesem Beispiel werden Paginierungsparameter verwendet, um zwei Flotten IDs gleichzeitig abzurufen. Die Antwort enthält ein `next-token` Attribut, das angibt, dass mehr Ergebnisse abgerufen werden müssen.

```
aws gamelift list-fleets \
  --limit 2
```

Ausgabe:

```
{
  "FleetIds": [
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  ],
  "NextToken":
  "eyJhd3NBZ2NvdW50SWQlOnsic3NjMmMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"
}
```

Sie können den `NextToken` Wert aus der vorherigen Antwort im nächsten Befehl übergeben, wie hier gezeigt, um die nächsten beiden Ergebnisse zu erhalten.

```
aws gamelift list-fleets \
  --limit 2 \
```

```
--next-  
token eyJhd3NBY2NvdW50SWQl0nsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRLZj
```

Beispiel 2: Um eine Liste aller Flotten in einer Region mit einem bestimmten Build oder Skript abzurufen

Im folgenden `list-builds` Beispiel werden die Flotten abgerufen, IDs die mit dem angegebenen Spiel-Build eingesetzt wurden. Wenn Sie mit Realtime Servers arbeiten, können Sie anstelle einer Build-ID eine Skript-ID angeben. Da in diesem Beispiel der Grenzparameter nicht angegeben ist, können die Ergebnisse bis zu 16 Flotten IDs umfassen.

```
aws gamelift list-fleets \  
--build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "FleetIds": [  
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"  
  ]  
}
```

- API-Einheiten finden Sie [ListFleets](#) in der AWS CLI Befehlsreferenz.

request-upload-credentials

Das folgende Codebeispiel zeigt die Verwendung `request-upload-credentials`.

AWS CLI

Um die Zugangsdaten für das Hochladen eines Builds zu aktualisieren

Im folgenden `create-build` Beispiel werden neue, gültige Zugangsdaten für das Hochladen einer GameLift Build-Datei an einen Amazon S3 S3-Speicherort abgerufen. Anmeldeinformationen haben eine begrenzte Lebensdauer. Sie erhalten die Build-ID aus der Antwort auf die ursprüngliche `CreateBuild` Anfrage.

```
aws gamelift request-upload-credentials \  

```

```
--build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "StorageLocation": {
    "Bucket": "gamelift-builds-us-west-2",
    "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "UploadCredentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "SessionToken": "AgoGb3JpZ21uENZ...EXAMPLETOKEN=="
  }
}
```

Weitere Informationen finden Sie unter [Hochladen eines benutzerdefinierten Server-Builds auf GameLift](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [RequestUploadCredentials](#) in der AWS CLI Befehlsreferenz.

start-fleet-actions

Das folgende Codebeispiel zeigt die Verwendung `start-fleet-actions`.

AWS CLI

Um die automatische Skalierungsaktivität der Flotte neu zu starten

Im folgenden `start-fleet-actions` Beispiel werden alle Skalierungsrichtlinien wieder verwendet, die für die angegebene Flotte definiert sind, aber durch den Aufruf `stop-fleet-actions` von ```` gestoppt wurden. Nach dem Start beginnen die Skalierungsrichtlinien sofort mit der Erfassung ihrer jeweiligen Metriken.

```
aws gamelift start-fleet-actions \
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --actions AUTO_SCALING
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [StartFleetActions](#) in der AWS CLI Befehlsreferenz.

stop-fleet-actions

Das folgende Codebeispiel zeigt die Verwendung `stop-fleet-actions`.

AWS CLI

Um die automatische Skalierung einer Flotte zu stoppen

Im folgenden `stop-fleet-actions` Beispiel wird die Verwendung aller Skalierungsrichtlinien beendet, die für die angegebene Flotte definiert sind. Nachdem die Richtlinien ausgesetzt wurden, bleibt die Flottenkapazität bei der Anzahl der aktiven Instances, sofern Sie sie nicht manuell anpassen.

```
aws gamelift start-fleet-actions \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --actions AUTO_SCALING
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [StopFleetActions](#) in der AWS CLI Befehlsreferenz.

update-build

Das folgende Codebeispiel zeigt die Verwendung `update-build`.

AWS CLI

Um einen benutzerdefinierten Spiel-Build zu aktualisieren

Im folgenden `update-build` Beispiel werden der Name und die Versionsinformationen geändert, die einer angegebenen Build-Ressource zugeordnet sind. Das zurückgegebene Build-Objekt bestätigt, dass die Änderungen erfolgreich vorgenommen wurden.

```
aws gamelift update-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --name MegaFrogRaceServer.NA.east \  
  --build-version 12345.east
```

Ausgabe:

```
{
```

```

    "Build": {
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "CreationTime": 1496708916.18,
      "Name": "MegaFrogRaceServer.NA.east",
      "OperatingSystem": "AMAZON_LINUX_2",
      "SizeOnDisk": 1304924,
      "Status": "READY",
      "Version": "12345.east"
    }
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren Ihrer Build-Dateien](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [UpdateBuild](#) in der AWS CLI Befehlsreferenz.

update-game-session-queue

Das folgende Codebeispiel zeigt die Verwendung `update-game-session-queue`.

AWS CLI

Um die Warteschlangenkonfiguration einer Spielsitzung zu aktualisieren

Das folgende `update-game-session-queue` Beispiel fügt ein neues Ziel hinzu und aktualisiert die Spielerlatenzrichtlinien für eine bestehende Warteschlange für Spielsitzungen.

```

aws gamelift update-game-session-queue \
  --name MegaFrogRace-NA \
  --destinations file://destinations.json \
  --player-latency-policies file://latency-policies.json

```

Inhalt von `destinations.json`:

```

{
  "Destinations": [
    {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
    {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
  ]
}

```

```

    {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
  ]
}

```

Inhalt von `latency-policies.json`:

```

{
  "PlayerLatencyPolicies": [
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},
    {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
120},
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
120}
  ]
}

```

Ausgabe:

```

{
  "GameSessionQueue": {
    "Destinations": [
      {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
      {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
      {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
    ],
    "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:111122223333:gamesessionqueue/MegaFrogRace-NA",
    "Name": "MegaFrogRace-NA",
    "TimeoutInSeconds": 600,
    "PlayerLatencyPolicies": [
      {"MaximumIndividualPlayerLatencyMilliseconds": 200},
      {"MaximumIndividualPlayerLatencyMilliseconds": 150,
"PolicyDurationSeconds": 120},
      {"MaximumIndividualPlayerLatencyMilliseconds": 100,
"PolicyDurationSeconds": 120}
    ]
  }
}

```

Weitere Informationen finden Sie unter [Using Multi-Region-Warteschlangen](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [UpdateGameSessionQueue](#) in der AWS CLI Befehlsreferenz.

upload-build

Das folgende Codebeispiel zeigt die Verwendung `upload-build`.

AWS CLI

Beispiel 1: Um einen Linux-Gameserver-Build hochzuladen

Das folgende `upload-build` Beispiel lädt Build-Dateien für Linux-Gameserver aus einem Dateiverzeichnis in den GameLift Dienst hoch und erstellt eine Build-Ressource.

```
aws gamelift upload-build \  
  --name MegaFrogRaceServer.NA \  
  --build-version 2.0.1 \  
  --build-root ~/MegaFrogRace_Server/release-na \  
  --operating-system AMAZON_LINUX_2 \  
  --server-sdk-version 4.0.2
```

Ausgabe:

```
Uploading ~/MegaFrogRace_Server/release-na: 16.0 KiB / 74.6 KiB (21.45%)  
Uploading ~/MegaFrogRace_Server/release-na: 32.0 KiB / 74.6 KiB (42.89%)  
Uploading ~/MegaFrogRace_Server/release-na: 48.0 KiB / 74.6 KiB (64.34%)  
Uploading ~/MegaFrogRace_Server/release-na: 64.0 KiB / 74.6 KiB (85.79%)  
Uploading ~/MegaFrogRace_Server/release-na: 74.6 KiB / 74.6 KiB (100.00%)  
Successfully uploaded ~/MegaFrogRace_Server/release-na to AWS GameLift  
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Beispiel 2: Um einen Windows-Gameserver-Build hochzuladen

Im folgenden `upload-build` Beispiel werden Build-Dateien für Windows-Gameserver aus einem Verzeichnis in den GameLift Dienst hochgeladen und ein Build-Datensatz erstellt.

```
aws gamelift upload-build \  
  --name MegaFrogRaceServer.NA \  
  --build-version 2.0.1 \  
  --build-root ~/MegaFrogRace_Server/release-na \  
  --operating-system AMAZON_LINUX_2 \  
  --server-sdk-version 4.0.2
```

```
--build-root C:\MegaFrogRace_Server\release-na \  
--operating-system WINDOWS_2012  
--server-sdk-version 4.0.2
```

Ausgabe:

```
Uploading C:\MegaFrogRace_Server\release-na: 16.0 KiB / 74.6 KiB (21.45%)  
Uploading C:\MegaFrogRace_Server\release-na: 32.0 KiB / 74.6 KiB (42.89%)  
Uploading C:\MegaFrogRace_Server\release-na: 48.0 KiB / 74.6 KiB (64.34%)  
Uploading C:\MegaFrogRace_Server\release-na: 64.0 KiB / 74.6 KiB (85.79%)  
Uploading C:\MegaFrogRace_Server\release-na: 74.6 KiB / 74.6 KiB (100.00%)  
Successfully uploaded C:\MegaFrogRace_Server\release-na to AWS GameLift  
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Weitere Informationen finden Sie unter [Hochladen eines benutzerdefinierten Server-Builds auf GameLift](#) im Amazon GameLift Developer Guide.

- API-Einheiten finden Sie [UploadBuild](#) in der AWS CLI Befehlsreferenz.

Global Accelerator-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Global Accelerator Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-custom-routing-endpoints

Das folgende Codebeispiel zeigt die Verwendung `add-custom-routing-endpoints`.

AWS CLI

Um einer Endpunktgruppe einen VPC Subnetzendpoint für einen benutzerdefinierten Routingbeschleuniger hinzuzufügen

Im folgenden `add-custom-routing-endpoints` Beispiel wird einer Endpunktgruppe für einen benutzerdefinierten Routingbeschleuniger ein VPC Subnetzendpoint hinzugefügt.

```
aws globalaccelerator add-custom-routing-endpoints \
  --endpoint-group-
  arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
  abcd-1234abcdefg/0123vxyz/endpoint-group/4321abcd \
  --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

Ausgabe:

```
{
  "EndpointDescriptions": [
    {
      "EndpointId": "subnet-1234567890abcdef0"
    }
  ],
  "EndpointGroupArn": "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefg/0123vxyz/endpoint-group/4321abcd"
}
```

Weitere Informationen finden Sie unter [VPCSubnetzendpunkte für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator Developer Guide AWS](#) .

- APIEinzelheiten finden Sie [AddCustomRoutingEndpoints](#) in AWS CLI der Befehlsreferenz.

advertise-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `advertise-byoip-cidr`.

AWS CLI

Um für einen Adressbereich zu werben

Im folgenden `advertise-byoip-cidr` Beispiel werden Sie AWS aufgefordert, für einen Adressbereich zu werben, den Sie für die Verwendung mit Ihren AWS Ressourcen bereitgestellt haben.

```
aws globalaccelerator advertise-byoip-cidr \  
  --cidr 198.51.100.0/24
```

Ausgabe:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_ADVERTISING"  
  }  
}
```

Weitere Informationen finden Sie unter [Bring Your Own IP Address in AWS Global Accelerator](#) im AWS Global Accelerator Developer Guide.

- API-Einheiten finden Sie [AdvertiseByoipCidr](#) in der AWS CLI Befehlsreferenz.

allow-custom-routing-traffic

Das folgende Codebeispiel zeigt die Verwendung `allow-custom-routing-traffic`.

AWS CLI

Um Traffic zu bestimmten EC2 Amazon-Instance-Zielen in einem VPC Subnetz für einen benutzerdefinierten Routing-Beschleuniger zuzulassen

Das folgende `allow-custom-routing-traffic` Beispiel gibt an, dass Datenverkehr zu bestimmten IP-Adressen und Ports der EC2 Amazon-Instance (Ziel) für einen VPC Subnetzpunkt in einem benutzerdefinierten Routing Accelerator zugelassen ist, der Datenverkehr empfangen kann.

```
aws globalaccelerator allow-custom-routing-traffic \  
  --endpoint-group-  
  arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
  abcd-1234abcdefg/Listener/0123vxyz/endpoint-group/ab8888example \  
  --endpoint-id subnet-abcd123example \  
  \
```

```
--destination-addresses "172.31.200.6" "172.31.200.7" \  
--destination-ports 80 81
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [VPCSubnetzendpunkte für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator Developer Guide AWS](#) .

- APIEinzelheiten finden Sie [AllowCustomRoutingTraffic](#) in AWS CLI der Befehlsreferenz.

create-accelerator

Das folgende Codebeispiel zeigt die Verwendung `create-accelerator`.

AWS CLI

Um einen Beschleuniger zu erstellen

Im folgenden `create-accelerator` Beispiel wird ein Accelerator mit zwei Tags mit zwei BYOIP statischen IP-Adressen erstellt. Sie müssen die US-West-2 (Oregon) Region angeben, um einen Accelerator zu erstellen oder zu aktualisieren.

```
aws globalaccelerator create-accelerator \  
  --name ExampleAccelerator \  
  --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \  
  --ip-addresses 192.0.2.250 198.51.100.52
```

Ausgabe:

```
{  
  "Accelerator": {  
    "AcceleratorArn":  
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefg",  
    "IpAddressType": "IPV4",  
    "Name": "ExampleAccelerator",  
    "Enabled": true,  
    "Status": "IN_PROGRESS",  
    "IpSets": [  
      {  
        "IpAddresses": [  

```

```

        "192.0.2.250",
        "198.51.100.52"
    ],
    "IpFamily": "IPv4"
}
],
"DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
"CreatedTime": 1542394847.0,
"LastModifiedTime": 1542394847.0
}
}

```

Weitere Informationen finden Sie unter [Accelerators in AWS Global Accelerator](#) im AWS Global Accelerator Developer Guide.

- API-Einheiten finden Sie [CreateAccelerator](#) in der AWS CLI Befehlsreferenz.

create-custom-routing-accelerator

Das folgende Codebeispiel zeigt die Verwendung `create-custom-routing-accelerator`.

AWS CLI

Um einen benutzerdefinierten Routing-Beschleuniger zu erstellen

Im folgenden `create-custom-routing-accelerator` Beispiel wird ein benutzerdefinierter Routingbeschleuniger mit den Tags `Name` und `erstelltProject`.

```

aws globalaccelerator create-custom-routing-accelerator \
  --name ExampleCustomRoutingAccelerator \
  --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \
  --ip-addresses 192.0.2.250 198.51.100.52

```

Ausgabe:

```

{
  "Accelerator": {
    "AcceleratorArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleCustomRoutingAccelerator",

```

```

    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
          "198.51.100.52"
        ],
        "IpFamily": "IPv4"
      }
    ],
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
    "CreatedTime": 1542394847.0,
    "LastModifiedTime": 1542394847.0
  }
}

```

Weitere Informationen finden Sie unter [Benutzerdefinierte Routing-Beschleuniger in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [CreateCustomRoutingAccelerator](#) in der AWS CLI Befehlsreferenz.

create-custom-routing-endpoint-group

Das folgende Codebeispiel zeigt die Verwendung `create-custom-routing-endpoint-group`.

AWS CLI

Um eine Endpunktgruppe für einen benutzerdefinierten Routingbeschleuniger zu erstellen

Im folgenden `create-custom-routing-endpoint-group` Beispiel wird eine Endpunktgruppe für einen benutzerdefinierten Routing Accelerator erstellt.

```

aws globalaccelerator create-custom-routing-endpoint-group \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
  --endpoint-group-region us-east-2 \
  --destination-configurations "FromPort=80,ToPort=81,Protocols=TCP,UDP"

```

Ausgabe:

```
{
```

```

    "EndpointGroup": {
      "EndpointGroupArn":
        "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
        abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd",
      "EndpointGroupRegion": "us-east-2",
      "DestinationDescriptions": [
        {
          "FromPort": 80,
          "ToPort": 81,
          "Protocols": [
            "TCP",
            "UDP"
          ]
        }
      ],
      "EndpointDescriptions": []
    }
  }
}

```

Weitere Informationen finden Sie unter [Endpunktgruppen für benutzerdefinierte Routing-Beschleuniger in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [CreateCustomRoutingEndpointGroup](#) in der AWS CLI Befehlsreferenz.

create-custom-routing-listener

Das folgende Codebeispiel zeigt die Verwendung `create-custom-routing-listener`.

AWS CLI

Um einen Listener für einen benutzerdefinierten Routing-Beschleuniger zu erstellen

Im folgenden `create-custom-routing-listener` Beispiel wird ein Listener mit einem Portbereich von 5000 bis 10000 für einen benutzerdefinierten Routing-Beschleuniger erstellt.

```

aws globalaccelerator create-custom-routing-listener \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefgh \
  --port-ranges FromPort=5000,ToPort=10000

```

Ausgabe:

```
{
  "Listener": {
    "PortRange": [
      "FromPort": 5000,
      "ToPort": 10000
    ],
    "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz"
  }
}
```

Weitere Informationen finden Sie unter [Listener für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [CreateCustomRoutingListener](#) in der AWS CLI Befehlsreferenz.

create-endpoint-group

Das folgende Codebeispiel zeigt die Verwendung `create-endpoint-group`.

AWS CLI

Um eine Endpunktgruppe zu erstellen

Im folgenden `create-endpoint-group` Beispiel wird eine Endpunktgruppe mit einem Endpunkt erstellt.

```
aws globalaccelerator create-endpoint-group \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
  --endpoint-group-region us-east-1 \
  --endpoint-configurations EndpointId=i-1234567890abcdef0,Weight=128
```

Ausgabe:

```
{
  "EndpointGroup": {
    "TrafficDialPercentage": 100.0,
    "EndpointDescriptions": [
      {
        "Weight": 128,
```

```

        "EndpointId": "i-1234567890abcdef0"
      }
    ],
    "EndpointGroupArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefggh/listener/0123vxyz/endpoint-group/098765zyxwvu",
    "EndpointGroupRegion": "us-east-1"
  }
}

```

Weitere Informationen finden Sie unter [Endpunktgruppen in AWS Global Accelerator](#) im AWS Global Accelerator Developer Guide.

- API-Einheiten finden Sie [CreateEndpointGroup](#) in der AWS CLI Befehlsreferenz.

create-listener

Das folgende Codebeispiel zeigt die Verwendung `create-listener`.

AWS CLI

Um einen Listener zu erstellen

Im folgenden `create-listener` Beispiel wird ein Listener mit zwei Ports erstellt.

```

aws globalaccelerator create-listener \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefggh \
  --port-ranges FromPort=80,ToPort=80 FromPort=81,ToPort=81 \
  --protocol TCP

```

Ausgabe:

```

{
  "Listener": {
    "PortRanges": [
      {
        "ToPort": 80,
        "FromPort": 80
      },
      {
        "ToPort": 81,

```



```

        "FromPort": 81
      }
    ],
    "ClientAffinity": "NONE",
    "Protocol": "TCP",
    "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz"
  }
}

```

Weitere Informationen finden Sie unter [Listeners in AWS Global Accelerator im Global Accelerator Developer Guide](#).AWS

- API-Einheiten finden Sie [CreateListener](#) in der AWS CLI Befehlsreferenz.

deny-custom-routing-traffic

Das folgende Codebeispiel zeigt die Verwendung `deny-custom-routing-traffic`.

AWS CLI

Um eine Zieladresse anzugeben, die keinen Datenverkehr in einem benutzerdefinierten Routingbeschleuniger empfangen kann

Im folgenden `deny-custom-routing-traffic` Beispiel werden Zieladressen in einem Subnetzendpunkt angegeben, der keinen Datenverkehr für einen benutzerdefinierten Routingbeschleuniger empfangen kann. Um mehr als eine Zieladresse anzugeben, trennen Sie die Adressen durch ein Leerzeichen. Bei einem erfolgreichen `deny-custom-routing-traffic` Aufruf erfolgt keine Antwort.

```

aws globalaccelerator deny-custom-routing-traffic \
  --endpoint-group-
arn "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab8888example" \
  --endpoint-id "subnet-abcd123example" \
  --destination-addresses "198.51.100.52"

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [VPC-Subnetzendpunkte für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [DenyCustomRoutingTraffic](#) in AWS CLI der Befehlsreferenz.

deprovision-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `deprovision-byoip-cidr`.

AWS CLI

Um die Bereitstellung eines Adressbereichs aufzuheben

Im folgenden `deprovision-byoip-cidr` Beispiel wird der angegebene Adressbereich freigegeben, den Sie für die Verwendung mit Ihren AWS Ressourcen bereitgestellt haben.

```
aws globalaccelerator deprovision-byoip-cidr \
  --cidr "198.51.100.0/24"
```

Ausgabe:

```
{
  "ByoipCidr": {
    "Cidr": "198.51.100.0/24",
    "State": "PENDING_DEPROVISIONING"
  }
}
```

Weitere Informationen finden Sie unter [Bring your own IP address in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [DeprovisionByoipCidr](#) in der AWS CLI Befehlsreferenz.

describe-accelerator-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-accelerator-attributes`.

AWS CLI

Um die Eigenschaften eines Beschleunigers zu beschreiben

Im folgenden `describe-accelerator-attributes` Beispiel werden die Attributdetails für einen Beschleuniger abgerufen.

```
aws globalaccelerator describe-accelerator-attributes \
```

```
--accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Ausgabe:

```
{
  "AcceleratorAttributes": {
    "FlowLogsEnabled": true
    "FlowLogsS3Bucket": flowlogs-abc
    "FlowLogsS3Prefix": bucketprefix-abc
  }
}
```

Weitere Informationen finden Sie unter [Accelerators in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [DescribeAcceleratorAttributes](#) in der AWS CLI Befehlsreferenz.

describe-accelerator

Das folgende Codebeispiel zeigt die Verwendung `describe-accelerator`.

AWS CLI

Um einen Beschleuniger zu beschreiben

Im folgenden `describe-accelerator` Beispiel werden die Details zum angegebenen Beschleuniger abgerufen.

```
aws globalaccelerator describe-accelerator \  
--accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Ausgabe:

```
{
  "Accelerator": {
    "AcceleratorArn":
    "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
```

```
    "Name": "ExampleAccelerator",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
          "198.51.100.52"
        ],
        "IpFamily": "IPv4"
      }
    ],
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
    "CreatedTime": 1542394847,
    "LastModifiedTime": 1542395013
  }
}
```

Weitere Informationen finden Sie unter [Accelerators in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [DescribeAccelerator](#) in der AWS CLI Befehlsreferenz.

describe-custom-routing-accelerator-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-routing-accelerator-attributes`.

AWS CLI

Um die Attribute eines benutzerdefinierten Routing-Beschleunigers zu beschreiben

Im folgenden `describe-custom-routing-accelerator-attributes` Beispiel werden die Attribute für einen benutzerdefinierten Routing-Beschleuniger beschrieben.

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Ausgabe:

```
{
```

```
"AcceleratorAttributes": {
  "FlowLogsEnabled": false
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Routing-Beschleuniger in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [DescribeCustomRoutingAcceleratorAttributes](#) in der AWS CLI Befehlsreferenz.

describe-custom-routing-accelerator

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-routing-accelerator`.

AWS CLI

Um einen benutzerdefinierten Routing-Beschleuniger zu beschreiben

Im folgenden `describe-custom-routing-accelerator` Beispiel werden die Details zum angegebenen benutzerdefinierten Routingbeschleuniger abgerufen.

```
aws globalaccelerator describe-custom-routing-accelerator \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Ausgabe:

```
{
  "Accelerator": {
    "AcceleratorArn":
      "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleCustomRoutingAccelerator",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
```

```

        "198.51.100.52"
      ],
      "IpFamily": "IPv4"
    }
  ],
  "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
  "CreatedTime": 1542394847,
  "LastModifiedTime": 1542395013
}
}

```

Weitere Informationen finden Sie unter [Benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator Developer Guide](#).AWS

- API-Einheiten finden Sie [DescribeCustomRoutingAccelerator](#) in der AWS CLI Befehlsreferenz.

describe-custom-routing-endpoint-group

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-routing-endpoint-group`.

AWS CLI

Um eine Endpunktgruppe für einen benutzerdefinierten Routingbeschleuniger zu beschreiben

Das folgende `describe-custom-routing-endpoint-group` Beispiel beschreibt eine Endpunktgruppe für einen benutzerdefinierten Routing Accelerator.

```

aws globalaccelerator describe-custom-routing-endpoint-group \
  --endpoint-group-
  arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
  abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab8888example

```

Ausgabe:

```

{
  "EndpointGroup": {
    "EndpointGroupArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
    abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab8888example",
    "EndpointGroupRegion": "us-east-2",
    "DestinationDescriptions": [

```

```
{
  {
    "FromPort": 5000,
    "ToPort": 10000,
    "Protocols": [
      "UDP"
    ]
  },
  "EndpointDescriptions": [
    {
      "EndpointId": "subnet-1234567890abcdef0"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Endpunktgruppen für benutzerdefinierte Routing-Beschleuniger in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [DescribeCustomRoutingEndpointGroup](#) in der AWS CLI Befehlsreferenz.

describe-custom-routing-listener

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-routing-listener`.

AWS CLI

Um einen Listener für einen benutzerdefinierten Routing-Beschleuniger zu beschreiben

Das folgende `describe-custom-routing-listener` Beispiel beschreibt einen Listener für einen benutzerdefinierten Routing-Beschleuniger.

```
aws globalaccelerator describe-custom-routing-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Ausgabe:

```
{
  "Listener": {
```

```

    "PortRanges": [
      "FromPort": 5000,
      "ToPort": 10000
    ],
    "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234"
  }
}

```

Weitere Informationen finden Sie unter [Listener für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [DescribeCustomRoutingListener](#) in der AWS CLI Befehlsreferenz.

describe-endpoint-group

Das folgende Codebeispiel zeigt die Verwendung `describe-endpoint-group`.

AWS CLI

Um eine Endpunktgruppe zu beschreiben

Im folgenden `describe-endpoint-group` Beispiel werden Details zu einer Endpunktgruppe mit den folgenden Endpunkten abgerufen: eine EC2 Amazon-Instance, eine ALB, eine und eine NLB

```

aws globalaccelerator describe-endpoint-group \
  --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab88888example

```

Ausgabe:

```

{
  "EndpointGroup": {
    "TrafficDialPercentage": 100.0,
    "EndpointDescriptions": [
      {
        "Weight": 128,
        "EndpointId": "i-1234567890abcdef0"
      }
    ]
  }
}

```



```

    },
    {
      "Weight": 128,
      "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
    },
    {
      "Weight": 128,
      "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
    }
  ],
  "EndpointGroupArn":
  "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
  "EndpointGroupRegion": "us-east-1"
}
}

```

Weitere Informationen finden Sie unter [Endpunktgruppen in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [DescribeEndpointGroup](#) in der AWS CLI Befehlsreferenz.

describe-listener

Das folgende Codebeispiel zeigt die Verwendung `describe-listener`.

AWS CLI

Um einen Zuhörer zu beschreiben

Das folgende `describe-listener` Beispiel beschreibt einen Listener.

```

aws globalaccelerator describe-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234

```

Ausgabe:

```
{
```

```
"Listener": {
  "ListenerArn":
  "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
  "PortRanges": [
    {
      "FromPort": 80,
      "ToPort": 80
    }
  ],
  "Protocol": "TCP",
  "ClientAffinity": "NONE"
}
```

Weitere Informationen finden Sie unter [Listeners in AWS Global Accelerator im Global Accelerator Developer Guide](#).AWS

- APIEinzelheiten finden Sie [DescribeListener](#) in der AWS CLI Befehlsreferenz.

list-accelerators

Das folgende Codebeispiel zeigt die Verwendung `list-accelerators`.

AWS CLI

Um Ihre Beschleuniger aufzulisten

Das folgende `list-accelerators` Beispiel listet die Accelerators in Ihrem AWS Konto auf. Dieses Konto hat zwei Beschleuniger.

```
aws globalaccelerator list-accelerators
```

Ausgabe:

```
{
  "Accelerators": [
    {
      "AcceleratorArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
      "Name": "TestAccelerator",
```


list-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `list-byoip-cidr`.

AWS CLI

Um Ihre Adressbereiche aufzulisten

Im folgenden `list-byoip-cidr` Beispiel werden die Adressbereiche für Bring Your Own IP Address (BYOIP) aufgeführt, die Sie für die Verwendung mit Global Accelerator bereitgestellt haben.

```
aws globalaccelerator list-byoip-cidrs
```

Ausgabe:

```
{
  "ByoipCidrs": [
    {
      "Cidr": "198.51.100.0/24",
      "State": "READY"
    }
    {
      "Cidr": "203.0.113.25/24",
      "State": "READY"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Bring your own IP address in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [ListByoipCidr](#) in der AWS CLI Befehlsreferenz.

list-custom-routing-accelerators

Das folgende Codebeispiel zeigt die Verwendung `list-custom-routing-accelerators`.

AWS CLI

Um Ihre benutzerdefinierten Routing-Beschleuniger aufzulisten


```

    {
      "FromPort": 80,
      "ToPort": 80,
      "Protocols": [
        "TCP",
        "UDP"
      ]
    }
  ]
  "EndpointDescriptions": [
    {
      "EndpointId": "subnet-abcd123example"
    }
  ]
}
]
}

```

Weitere Informationen finden Sie unter [Endpunktgruppen für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator Developer Guide](#).AWS

- API-Einheiten finden Sie [ListCustomRoutingEndpointGroups](#) in der AWS CLI Befehlsreferenz.

list-custom-routing-listeners

Das folgende Codebeispiel zeigt die Verwendung `list-custom-routing-listeners`.

AWS CLI

Um Listener für benutzerdefinierte Routing-Beschleuniger aufzulisten

Das folgende `list-custom-routing-listeners` Beispiel listet die Listener für einen benutzerdefinierten Routing-Beschleuniger auf.

```

aws globalaccelerator list-custom-routing-listeners \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Ausgabe:

```

{
  "Listeners": [

```

```
{
  "ListenerArn":
  "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
  "PortRanges": [
    {
      "FromPort": 5000,
      "ToPort": 10000
    }
  ],
  "Protocol": "TCP"
}
]
```

Weitere Informationen finden Sie unter [Listener für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [ListCustomRoutingListeners](#) in der AWS CLI Befehlsreferenz.

list-custom-routing-port-mappings-by-destination

Das folgende Codebeispiel zeigt die Verwendung `list-custom-routing-port-mappings-by-destination`.

AWS CLI

Um die Portzuordnungen für ein bestimmtes benutzerdefiniertes Routing Accelerator-Ziel aufzulisten

Das folgende `list-custom-routing-port-mappings-by-destination` Beispiel stellt die Portzuordnungen für einen bestimmten EC2 Zielsever (an der Zieladresse) für einen benutzerdefinierten Routingbeschleuniger bereit.

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \
  --endpoint-id subnet-abcd123example \
  --destination-address 198.51.100.52
```

Ausgabe:

```
{
```



```

    "DestinationPortMappings": [
      {
        "AcceleratorArn":
          "arn:aws:globalaccelerator::402092451327:accelerator/24ea29b8-
          d750-4489-8919-3095f3c4b0a7",
        "AcceleratorSocketAddresses": [
          {
            "IpAddress": "192.0.2.250",
            "Port": 65514
          },
          {
            "IpAddress": "192.10.100.99",
            "Port": 65514
          }
        ],
        "EndpointGroupArn":
          "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
          abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example",
        "EndpointId": "subnet-abcd123example",
        "EndpointGroupRegion": "us-west-2",
        "DestinationSocketAddress": {
          "IpAddress": "198.51.100.52",
          "Port": 80
        },
        "IpAddressType": "IPv4",
        "DestinationTrafficState": "ALLOW"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [So funktionieren benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [ListCustomRoutingPortMappingsByDestination](#) in der AWS CLI Befehlsreferenz.

list-custom-routing-port-mappings

Das folgende Codebeispiel zeigt die Verwendung `list-custom-routing-port-mappings`.

AWS CLI

Um die Portzuordnungen in einem benutzerdefinierten Routing-Beschleuniger aufzulisten

Das folgende `list-custom-routing-port-mappings` Beispiel enthält eine unvollständige Liste der Portzuordnungen in einem benutzerdefinierten Routingbeschleuniger.

```
aws globalaccelerator list-custom-routing-port-mappings \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Ausgabe:

```
{
  "PortMappings": [
    {
      "AcceleratorPort": 40480,
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
      "EndpointId": "subnet-1234567890abcdef0",
      "DestinationSocketAddress": {
        "IpAddress": "192.0.2.250",
        "Port": 80
      },
      "Protocols": [
        "TCP",
        "UDP"
      ],
      "DestinationTrafficState": "ALLOW"
    }
    {
      "AcceleratorPort": 40481,
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
      "EndpointId": "subnet-1234567890abcdef0",
      "DestinationSocketAddress": {
        "IpAddress": "192.0.2.251",
        "Port": 80
      },
      "Protocols": [
        "TCP",
        "UDP"
      ],
      "DestinationTrafficState": "ALLOW"
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [So funktionieren benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [ListCustomRoutingPortMappings](#) in der AWS CLI Befehlsreferenz.

list-endpoint-groups

Das folgende Codebeispiel zeigt die Verwendung `list-endpoint-groups`.

AWS CLI

Um Endpunktgruppen aufzulisten

Das folgende `list-endpoint-groups` Beispiel listet die Endpunktgruppen für einen Listener auf. Dieser Listener hat zwei Endpunktgruppen.

```
aws globalaccelerator --region us-west-2 list-endpoint-groups \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Ausgabe:

```
{
  "EndpointGroups": [
    {
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
      "EndpointGroupRegion": "eu-central-1",
      "EndpointDescriptions": [],
      "TrafficDialPercentage": 100.0,
      "HealthCheckPort": 80,
      "HealthCheckProtocol": "TCP",
      "HealthCheckIntervalSeconds": 30,
      "ThresholdCount": 3
    }
    {
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab99999example",
```

```

        "EndpointGroupRegion": "us-east-1",
        "EndpointDescriptions": [],
        "TrafficDialPercentage": 50.0,
        "HealthCheckPort": 80,
        "HealthCheckProtocol": "TCP",
        "HealthCheckIntervalSeconds": 30,
        "ThresholdCount": 3
    }
]
}

```

Weitere Informationen finden Sie unter [Endpunktgruppen in AWS Global Accelerator](#) im AWS Global Accelerator Developer Guide.

- API-Einheiten finden Sie [ListEndpointGroups](#) in der AWS CLI Befehlsreferenz.

list-listeners

Das folgende Codebeispiel zeigt die Verwendung `list-listeners`.

AWS CLI

Um Zuhörer aufzulisten

Das folgende `list-listeners` Beispiel listet die Listener für einen Accelerator auf.

```

aws globalaccelerator list-listeners \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Ausgabe:

```

{
  "Listeners": [
    {
      "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
      "PortRanges": [
        {
          "FromPort": 80,
          "ToPort": 80
        }
      ]
    }
  ]
}

```

```
    ],
    "Protocol": "TCP",
    "ClientAffinity": "NONE"
  }
]
```

Weitere Informationen finden Sie unter [Listeners in AWS Global Accelerator im Global Accelerator Developer Guide](#).AWS

- API-Einheiten finden Sie [ListListeners](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für einen Beschleuniger aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für einen bestimmten Beschleuniger auf.

```
aws globalaccelerator list-tags-for-resource \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "Project",
      "Value": "A123456"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

provision-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `provision-byoip-cidr`.

AWS CLI

Um einen Adressbereich bereitzustellen

Im folgenden `provision-byoip-cidr` Beispiel wird der angegebene Adressbereich zur Verwendung mit Ihren AWS Ressourcen bereitgestellt.

```
aws globalaccelerator provision-byoip-cidr \  
  --cidr 192.0.2.250/24 \  
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Ausgabe:

```
{  
  "ByoipCidr": {  
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_PROVISIONING"  
  }  
}
```

Weitere Informationen finden Sie unter [Bring your own IP address in AWS Global Accelerator](#) im AWS Global Accelerator Developer Guide.

- API-Einheiten finden Sie [ProvisionByoipCidr](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einen Beschleuniger zu taggen

Im folgenden `tag-resource` Beispiel werden einem Accelerator die Tags `Name` und `Project` hinzugefügt, zusammen mit den entsprechenden Werten für jeden.

```
aws globalaccelerator tag-resource \  
  --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
  abcd-1234-abcd-1234abcdefgh \
```

```
--tags Key="Name",Value="Example Name" Key="Project",Value="Example Project"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einem Accelerator zu entfernen

Im folgenden `untag-resource` Beispiel werden die Tags `Name` und `Project` aus einem Accelerator entfernt.

```
aws globalaccelerator untag-resource \  
  --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --tag-keys Key="Name" Key="Project"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-accelerator-attributes

Das folgende Codebeispiel zeigt die Verwendung `update-accelerator-attributes`.

AWS CLI

Um die Attribute eines Accelerators zu aktualisieren

Im folgenden `update-accelerator-attributes` Beispiel wird ein Accelerator aktualisiert, um Flow-Logs zu aktivieren. Sie müssen die `US-West-2 (Oregon)` Region angeben, um Accelerator-Attribute zu erstellen oder zu aktualisieren.

```
aws globalaccelerator update-accelerator-attributes \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --flow-logs-enabled \  
  --flow-logs-s3-bucket flowlogs-abc \  
  --flow-logs-s3-prefix bucketprefix-abc
```

Ausgabe:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": true  
    "FlowLogsS3Bucket": flowlogs-abc  
    "FlowLogsS3Prefix": bucketprefix-abc  
  }  
}
```

Weitere Informationen finden Sie unter [Accelerators in AWS Global Accelerator](#) im AWS Global Accelerator Developer Guide.

- API-Einheiten finden Sie [UpdateAcceleratorAttributes](#) in der AWS CLI Befehlsreferenz.

update-accelerator

Das folgende Codebeispiel zeigt die Verwendung `update-accelerator`.

AWS CLI

Um einen Beschleuniger zu aktualisieren

Im folgenden `update-accelerator` Beispiel wird ein Beschleuniger so geändert, dass der Name des Beschleunigers geändert wird. `ExampleAcceleratorNew` Sie müssen die US-West-2 (Oregon) Region angeben, um Beschleuniger zu erstellen oder zu aktualisieren.

```
aws globalaccelerator update-accelerator \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --name ExampleAcceleratorNew
```

Ausgabe:


```
{
  "Accelerator": {
    "AcceleratorArn":
      "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleAcceleratorNew",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
          "198.51.100.52"
        ],
        "IpFamily": "IPv4"
      }
    ],
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
    "CreatedTime": 1232394847,
    "LastModifiedTime": 1232395654
  }
}
```

Weitere Informationen finden Sie unter [Accelerators in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [UpdateAccelerator](#) in der AWS CLI Befehlsreferenz.

update-custom-routing-accelerator-attributes

Das folgende Codebeispiel zeigt die Verwendung `update-custom-routing-accelerator-attributes`.

AWS CLI

Um die Attribute eines benutzerdefinierten Routing Accelerators zu aktualisieren

Im folgenden `update-custom-routing-accelerator-attributes` Beispiel wird ein benutzerdefinierter Routing-Beschleuniger aktualisiert, sodass Flow-Logs aktiviert werden.

```
aws globalaccelerator update-custom-routing-accelerator-attributes \
```

```
--accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
--flow-logs-enabled \
--flow-logs-s3-bucket flowlogs-abc \
--flow-logs-s3-prefix bucketprefix-abc
```

Ausgabe:

```
{
  "AcceleratorAttributes": {
    "FlowLogsEnabled": true
    "FlowLogsS3Bucket": flowlogs-abc
    "FlowLogsS3Prefix": bucketprefix-abc
  }
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Routing-Beschleuniger in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [UpdateCustomRoutingAcceleratorAttributes](#) in der AWS CLI Befehlsreferenz.

update-custom-routing-accelerator

Das folgende Codebeispiel zeigt die Verwendung `update-custom-routing-accelerator`.

AWS CLI

Um einen benutzerdefinierten Routing-Beschleuniger zu aktualisieren

Im folgenden `update-custom-routing-accelerator` Beispiel wird ein benutzerdefinierter Routing-Beschleuniger geändert, um den Namen des Beschleunigers zu ändern.

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
  --name ExampleCustomRoutingAcceleratorNew
```

Ausgabe:

```
{
```

```

    "Accelerator": {
      "AcceleratorArn":
"arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
      "IpAddressType": "IPV4",
      "Name": "ExampleCustomRoutingAcceleratorNew",
      "Enabled": true,
      "Status": "IN_PROGRESS",
      "IpSets": [
        {
          "IpAddresses": [
            "192.0.2.250",
            "198.51.100.52"
          ],
          "IpFamily": "IPv4"
        }
      ],
      "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
      "CreatedTime": 1232394847,
      "LastModifiedTime": 1232395654
    }
  }
}

```

Weitere Informationen finden Sie unter [Benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator Developer Guide](#).AWS

- API-Einheiten finden Sie [UpdateCustomRoutingAccelerator](#) in der AWS CLI Befehlsreferenz.

update-custom-routing-listener

Das folgende Codebeispiel zeigt die Verwendung `update-custom-routing-listener`.

AWS CLI

Um einen Listener für einen benutzerdefinierten Routing-Beschleuniger zu aktualisieren

Im folgenden `update-custom-routing-listener` Beispiel wird ein Listener aktualisiert, um den Portbereich zu ändern.

```

aws globalaccelerator update-custom-routing-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \

```

```
--port-ranges FromPort=10000,ToPort=20000
```

Ausgabe:

```
{
  "Listener": {
    "ListenerArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz
    "PortRanges": [
      {
        "FromPort": 10000,
        "ToPort": 20000
      }
    ],
    "Protocol": "TCP"
  }
}
```

Weitere Informationen finden Sie unter [Listener für benutzerdefinierte Routing-Beschleuniger in Global Accelerator im AWS Global Accelerator AWS Developer Guide](#).

- API-Einheiten finden Sie [UpdateCustomRoutingListener](#) in der AWS CLI Befehlsreferenz.

update-endpoint-group

Das folgende Codebeispiel zeigt die Verwendung `update-endpoint-group`.

AWS CLI

Um eine Endpunktgruppe zu aktualisieren

Im folgenden `update-endpoint-group` Beispiel werden einer Endpunktgruppe drei Endpunkte hinzugefügt: eine Elastic IP-Adresse ALB, ein und ein NLB.

```
aws globalaccelerator update-endpoint-group \
  --endpoint-group-
  arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab8888example \
  --endpoint-configurations \
    EndpointId=eipalloc-eip01234567890abc,Weight=128 \
```


AWS CLI

Um einen Listener zu aktualisieren

Im folgenden `update-listener` Beispiel wird ein Listener aktualisiert, sodass der Port auf 100 geändert wird.

```
aws globalaccelerator update-listener \  
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \  
  --port-ranges FromPort=100,ToPort=100
```

Ausgabe:

```
{  
  "Listener": {  
    "ListenerArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz"  
    "PortRanges": [  
      {  
        "FromPort": 100,  
        "ToPort": 100  
      }  
    ],  
    "Protocol": "TCP",  
    "ClientAffinity": "NONE"  
  }  
}
```

Weitere Informationen finden Sie unter [Listeners in AWS Global Accelerator im Global Accelerator Developer Guide](#).AWS

- API-Einheiten finden Sie [UpdateListener](#) in der AWS CLI Befehlsreferenz.

withdraw-byoip-cidr

Das folgende Codebeispiel zeigt die Verwendung `withdraw-byoip-cidr`.

AWS CLI

Um einen Adressbereich zurückzuziehen

Im folgenden `withdraw-byoip-cidr` Beispiel wird ein Adressbereich aus AWS Global Accelerator entfernt, den Sie zuvor für die Verwendung mit Ihren AWS Ressourcen beworben haben.

```
aws globalaccelerator withdraw-byoip-cidr \  
  --cidr 192.0.2.250/24
```

Ausgabe:

```
{  
  "ByoipCidr": {  
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_WITHDRAWING"  
  }  
}
```

Weitere Informationen finden Sie unter [Bring your own IP address in AWS Global Accelerator im AWS Global Accelerator Developer Guide](#).

- API-Einheiten finden Sie [WithdrawByoipCidr](#) in der AWS CLI Befehlsreferenz.

AWS Glue Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Glue.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-stop-job-run

Das folgende Codebeispiel zeigt die Verwendung `batch-stop-job-run`.

AWS CLI

Um Jobläufe zu beenden

Das folgende `batch-stop-job-run` Beispiel stoppt die Ausführung eines Auftrags.

```
aws glue batch-stop-job-run \  
  --job-name "my-testing-job" \  
  --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```

Ausgabe:

```
{  
  "SuccessfulSubmissions": [  
    {  
      "JobName": "my-testing-job",  
      "JobRunId":  
"jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f"  
    }  
  ],  
  "Errors": [],  
  "ResponseMetadata": {  
    "RequestId": "66bd6b90-01db-44ab-95b9-6aeff0e73d88",  
    "HTTPStatusCode": 200,  
    "HTTPHeaders": {  
      "date": "Fri, 16 Oct 2020 20:54:51 GMT",  
      "content-type": "application/x-amz-json-1.1",  
      "content-length": "148",  
      "connection": "keep-alive",  
      "x-amzn-requestid": "66bd6b90-01db-44ab-95b9-6aeff0e73d88"  
    },  
    "RetryAttempts": 0  
  }  
}
```

Weitere Informationen finden Sie unter [Auftragsausführungen](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [BatchStopJobRun](#) in der AWS CLI Befehlsreferenz.

create-connection

Das folgende Codebeispiel zeigt die Verwendung `create-connection`.

AWS CLI

Um eine Verbindung für AWS Glue-Datenspeicher herzustellen

Das folgende `create-connection` Beispiel erstellt eine Verbindung im AWS Glue-Datenkatalog, die Verbindungsinformationen für einen Kafka-Datenspeicher bereitstellt.

```
aws glue create-connection \
  --connection-input '{ \
    "Name":"conn-kafka-custom", \
    "Description":"kafka connection with ssl to custom kafka", \
    "ConnectionType":"KAFKA", \
    "ConnectionProperties":{ \
      "KAFKA_BOOTSTRAP_SERVERS":"<Kafka-broker-server-url>:<SSL-Port>", \
      "KAFKA_SSL_ENABLED":"true", \
      "KAFKA_CUSTOM_CERT": "s3://bucket/prefix/cert-file.pem" \
    }, \
    "PhysicalConnectionRequirements":{ \
      "SubnetId":"subnet-1234", \
      "SecurityGroupIdList":["sg-1234"], \
      "AvailabilityZone":"us-east-1a"} \
  }' \
  --region us-east-1
  --endpoint https://glue.us-east-1.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verbindungen im AWS Glue-Datenkatalog definieren](#) im AWS Glue-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateConnection](#) in der AWS CLI Befehlsreferenz.

create-database

Das folgende Codebeispiel zeigt die Verwendung `create-database`.

AWS CLI

So erstellen Sie eine Datenbank

Das folgende `create-database` Beispiel erstellt eine Datenbank im AWS Glue-Datenkatalog.

```
aws glue create-database \  
  --database-input "{\"Name\":\"tempdb\"}" \  
  --profile my_profile \  
  --endpoint https://glue.us-east-1.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Definieren einer Datenbank in Ihrem Datenkatalog](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [CreateDatabase](#) unter AWS CLI Befehlsreferenz.

`create-job`

Das folgende Codebeispiel zeigt die Verwendung `create-job`.

AWS CLI

Einen Auftrag zur Datentransformation erstellen

Im folgenden Beispiel für `create-job` wird ein Streaming-Job erstellt, der ein in S3 gespeichertes Skript ausführt.

```
aws glue create-job \  
  --name my-testing-job \  
  --role AWSGlueServiceRoleDefault \  
  --command '{ \  
    "Name": "gluestreaming", \  
    "ScriptLocation": "s3://DOC-EXAMPLE-BUCKET/folder/" \  
  }' \  
  --region us-east-1 \  
  --output json \  
  --default-arguments '{ \  
    "--job-language": "scala", \  
    "--class": "GlueApp" \  
  }' \  
  \
```

```
--profile my-profile \  
--endpoint https://glue.us-east-1.amazonaws.com
```

Inhalt von `test_script.scala`:

```
import com.amazonaws.services.glue.ChoiceOption  
import com.amazonaws.services.glue.GlueContext  
import com.amazonaws.services.glue.MappingSpec  
import com.amazonaws.services.glue.ResolveSpec  
import com.amazonaws.services.glue.errors.CallSite  
import com.amazonaws.services.glue.util.GlueArgParser  
import com.amazonaws.services.glue.util.Job  
import com.amazonaws.services.glue.util.JsonOptions  
import org.apache.spark.SparkContext  
import scala.collection.JavaConverters._  
  
object GlueApp {  
  def main(sysArgs: Array[String]) {  
    val spark: SparkContext = new SparkContext()  
    val glueContext: GlueContext = new GlueContext(spark)  
    // @params: [JOB_NAME]  
    val args = GlueArgParser.getResolvedOptions(sysArgs,  
Seq("JOB_NAME").toArray)  
    Job.init(args("JOB_NAME"), glueContext, args.asJava)  
    // @type: DataSource  
    // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx  
= "datasource0"]  
    // @return: datasource0  
    // @inputs: []  
    val datasource0 = glueContext.getCatalogSource(database = "tempdb",  
tableName = "s3-source", redshiftTmpDir = "", transformationContext =  
"datasource0").getDynamicFrame()  
    // @type: ApplyMapping  
    // @args: [mapping = [("sensorid", "int", "sensorid", "int"),  
("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",  
"status", "string")], transformation_ctx = "applymapping1"]  
    // @return: applymapping1  
    // @inputs: [frame = datasource0]  
    val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",  
"int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",  
"int"), ("status", "string", "status", "string")), caseSensitive = false,  
transformationContext = "applymapping1")  
    // @type: SelectFields
```

```

    // @args: [paths = ["sensorid", "currenttemperature", "status"],
transformation_ctx = "selectfields2"]
    // @return: selectfields2
    // @inputs: [frame = applymapping1]
    val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
"currenttemperature", "status"), transformationContext = "selectfields2")
    // @type: ResolveChoice
    // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-
s3-sink", transformation_ctx = "resolvechoice3"]
    // @return: resolvechoice3
    // @inputs: [frame = selectfields2]
    val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
Some("my-s3-sink"), transformationContext = "resolvechoice3")
    // @type: DataSink
    // @args: [database = "tempdb", table_name = "my-s3-sink",
transformation_ctx = "datasink4"]
    // @return: datasink4
    // @inputs: [frame = resolvechoice3]
    val datasink4 = glueContext.getCatalogSink(database = "tempdb",
tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
"datasink4").writeDynamicFrame(resolvechoice3)
    Job.commit()
  }
}

```

Ausgabe:

```

{
  "Name": "my-testing-job"
}

```

Weitere Informationen finden Sie unter [Authoring Jobs in AWS Glue im AWS Glue Developer Guide](#).

- API-Einheiten finden Sie [CreateJob](#) in der AWS CLI Befehlsreferenz.

create-table

Das folgende Codebeispiel zeigt die Verwendung `create-table`.

AWS CLI

Beispiel 1: So erstellen Sie eine Tabelle für einen Kinesis-Datenstream

Das folgende `create-table` Beispiel erstellt eine Tabelle im AWS Glue-Datenkatalog, die einen Kinesis-Datenstrom beschreibt.

```
aws glue create-table \
  --database-name tempdb \
  --table-input '{"Name":"test-kinesis-input", "StorageDescriptor":{ \
    "Columns":[ \
      {"Name":"sensorid", "Type":"int"}, \
      {"Name":"currenttemperature", "Type":"int"}, \
      {"Name":"status", "Type":"string"} \
    ], \
    "Location":"my-testing-stream", \
    "Parameters":{ \
      "typeOfData":"kinesis", "streamName":"my-testing-stream", \
      "kinesisUrl":"https://kinesis.us-east-1.amazonaws.com" \
    }, \
    "SerdeInfo":{ \
      "SerializationLibrary":"org.openx.data.jsonserde.JsonSerDe"} \
  }, \
  "Parameters":{ \
    "classification":"json"} \
}' \
  --profile my-profile \
  --endpoint https://glue.us-east-1.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Definieren von Tabellen im AWS Glue-Datenkatalog](#) im AWS Glue-Entwicklerhandbuch.

Beispiel 2: So erstellen Sie eine Tabelle für einen Kafka-Datenspeicher

Das folgende `create-table` Beispiel erstellt eine Tabelle im AWS Glue-Datenkatalog, die einen Kafka-Datenspeicher beschreibt.

```
aws glue create-table \
  --database-name tempdb \
  --table-input '{"Name":"test-kafka-input", "StorageDescriptor":{ \
    "Columns":[ \
```

```

        {"Name": "sensorid", "Type": "int"}, \
        {"Name": "currenttemperature", "Type": "int"}, \
        {"Name": "status", "Type": "string"}
    ], \
    "Location": "glue-topic", \
    "Parameters": { \
        "typeOfData": "kafka", "topicName": "glue-topic", \
        "connectionName": "my-kafka-connection"
    }, \
    "SerdeInfo": { \
        "SerializationLibrary": "org.apache.hadoop.hive.serde2.OpenCSVSerde"
    }
} \
} \
"Parameters": { \
    "separatorChar": ","
} \
}' \
--profile my-profile \
--endpoint https://glue.us-east-1.amazonaws.com

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Definieren von Tabellen im AWS Glue-Datenkatalog](#) im AWS Glue-Entwicklerhandbuch.

Beispiel 3: So erstellen Sie eine Tabelle für einen AWS S3-Datenspeicher

Das folgende `create-table` Beispiel erstellt eine Tabelle im AWS Glue Data Catalog, die einen AWS Simple Storage Service (AWS S3) -Datenspeicher beschreibt.

```

aws glue create-table \
  --database-name tempdb \
  --table-input '{"Name": "s3-output", "StorageDescriptor": { \
    "Columns": [ \
      {"Name": "s1", "Type": "string"}, \
      {"Name": "s2", "Type": "int"}, \
      {"Name": "s3", "Type": "string"}
    ], \
    "Location": "s3://bucket-path/", \
    "SerdeInfo": { \
      "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
    }
  }, \
  "Parameters": { \
    "classification": "json"
  } \
} \

```

```
}' \  
--profile my-profile \  
--endpoint https://glue.us-east-1.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Definieren von Tabellen im AWS Glue-Datenkatalog](#) im AWS Glue-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateTable](#) in der AWS CLI Befehlsreferenz.

delete-job

Das folgende Codebeispiel zeigt die Verwendung `delete-job`.

AWS CLI

Einen Auftrag löschen

Das folgende Beispiel für `delete-job` löscht einen Auftrag, der nicht mehr benötigt wird.

```
aws glue delete-job \  
  --job-name my-testing-job
```

Ausgabe:

```
{  
  "JobName": "my-testing-job"  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Jobs auf der AWS Glue-Konsole](#) im AWS Glue-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteJob](#) in der AWS CLI Befehlsreferenz.

get-databases

Das folgende Codebeispiel zeigt die Verwendung `get-databases`.

AWS CLI

Um die Definitionen einiger oder aller Datenbanken im AWS Glue-Datenkatalog aufzulisten

Das folgende Beispiel für `get-databases` gibt Informationen über die Datenbanken im Datenkatalog zurück.

```
aws glue get-databases
```

Ausgabe:

```
{
  "DatabaseList": [
    {
      "Name": "default",
      "Description": "Default Hive database",
      "LocationUri": "file:/spark-warehouse",
      "CreateTime": 1602084052.0,
      "CreateTableDefaultPermissions": [
        {
          "Principal": {
            "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
          },
          "Permissions": [
            "ALL"
          ]
        }
      ],
      "CatalogId": "111122223333"
    },
    {
      "Name": "flights-db",
      "CreateTime": 1587072847.0,
      "CreateTableDefaultPermissions": [
        {
          "Principal": {
            "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
          },
          "Permissions": [
            "ALL"
          ]
        }
      ],
      "CatalogId": "111122223333"
    },
    {
      "Name": "legislators",
```



```

    "CreateTime": 1601415625.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ],
    "CatalogId": "111122223333"
  },
  {
    "Name": "tempdb",
    "CreateTime": 1601498566.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ],
    "CatalogId": "111122223333"
  }
]
}

```

Weitere Informationen finden Sie unter [Definieren einer Datenbank in Ihrem Datenkatalog](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [GetDatabases](#) in der AWS CLI Befehlsreferenz.

get-job-run

Das folgende Codebeispiel zeigt die Verwendung `get-job-run`.

AWS CLI

Informationen zu einer Auftragsausführung abrufen

Das folgende Beispiel für `get-job-run` ruft Informationen zu einer Auftragsausführung ab.

```
aws glue get-job-run \  
  --job-name "Combine legislators data" \  
  --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

Ausgabe:

```
{  
  "JobRun": {  
    "Id": "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",  
    "Attempt": 0,  
    "JobName": "Combine legislators data",  
    "StartedOn": 1602873931.255,  
    "LastModifiedOn": 1602874075.985,  
    "CompletedOn": 1602874075.985,  
    "JobRunState": "SUCCEEDED",  
    "Arguments": {  
      "--enable-continuous-cloudwatch-log": "true",  
      "--enable-metrics": "",  
      "--enable-spark-ui": "true",  
      "--job-bookmark-option": "job-bookmark-enable",  
      "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-east-1/  
sparkHistoryLogs/"  
    },  
    "PredecessorRuns": [],  
    "AllocatedCapacity": 10,  
    "ExecutionTime": 117,  
    "Timeout": 2880,  
    "MaxCapacity": 10.0,  
    "WorkerType": "G.1X",  
    "NumberOfWorkers": 10,  
    "LogGroupName": "/aws-glue/jobs",  
    "GlueVersion": "2.0"  
  }  
}
```

Weitere Informationen finden Sie unter [Auftragsausführungen](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [GetJobRun](#) in der AWS CLI Befehlsreferenz.

get-job-runs

Das folgende Codebeispiel zeigt die Verwendung `get-job-runs`.

AWS CLI

Informationen über alle Ausführungen eines Auftrags abrufen

Das folgende Beispiel für `get-job-runs` ruft Informationen zu allen Ausführungen eines Auftrags ab.

```
aws glue get-job-runs \  
  --job-name "my-testing-job"
```

Ausgabe:

```
{  
  "JobRuns": [  
    {  
      "Id":  
"jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",  
      "Attempt": 0,  
      "JobName": "my-testing-job",  
      "StartedOn": 1602873931.255,  
      "LastModifiedOn": 1602874075.985,  
      "CompletedOn": 1602874075.985,  
      "JobRunState": "SUCCEEDED",  
      "Arguments": {  
        "--enable-continuous-cloudwatch-log": "true",  
        "--enable-metrics": "",  
        "--enable-spark-ui": "true",  
        "--job-bookmark-option": "job-bookmark-enable",  
        "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-  
east-1/sparkHistoryLogs/"  
      },  
      "PredecessorRuns": [],  
      "AllocatedCapacity": 10,  
      "ExecutionTime": 117,  
      "Timeout": 2880,  
      "MaxCapacity": 10.0,  
      "WorkerType": "G.1X",  
      "NumberOfWorkers": 10,  
      "LogGroupName": "/aws-glue/jobs",
```

```

        "GlueVersion": "2.0"
    },
    {
        "Id":
"jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_2",
        "Attempt": 2,
        "PreviousRunId":
"jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
        "JobName": "my-testing-job",
        "StartedOn": 1602811168.496,
        "LastModifiedOn": 1602811282.39,
        "CompletedOn": 1602811282.39,
        "JobRunState": "FAILED",
        "ErrorMessage": "An error occurred while calling
o122.pyWriteDynamicFrame.
                Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
AccessDenied;
                Request ID: 021AAB703DB20A2D;
                S3 Extended Request ID: teZk24Y09TkXzBvMPG502L5VJBhe9DJuWA9/
TXtuG0qfByajkfl/Tlqt5JBGdEGpigAqzdMDM/U=)",
        "PredecessorRuns": [],
        "AllocatedCapacity": 10,
        "ExecutionTime": 110,
        "Timeout": 2880,
        "MaxCapacity": 10.0,
        "WorkerType": "G.1X",
        "NumberOfWorkers": 10,
        "LogGroupName": "/aws-glue/jobs",
        "GlueVersion": "2.0"
    },
    {
        "Id":
"jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
        "Attempt": 1,
        "PreviousRunId":
"jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f",
        "JobName": "my-testing-job",
        "StartedOn": 1602811020.518,
        "LastModifiedOn": 1602811138.364,
        "CompletedOn": 1602811138.364,
        "JobRunState": "FAILED",
        "ErrorMessage": "An error occurred while calling
o122.pyWriteDynamicFrame.

```

```

    Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
AccessDenied;
    Request ID: 2671D37856AE7ABB;
    S3 Extended Request ID: RLJCJw20brV
+PpC6Gp0RahyF2fp9f1B5SSb2bTGPhUSPVizLXR11PN3QZ1db+v1o9qRVktNYbW8=)",
    "PredecessorRuns": [],
    "AllocatedCapacity": 10,
    "ExecutionTime": 113,
    "Timeout": 2880,
    "MaxCapacity": 10.0,
    "WorkerType": "G.1X",
    "NumberOfWorkers": 10,
    "LogGroupName": "/aws-glue/jobs",
    "GlueVersion": "2.0"
  }
]
}

```

Weitere Informationen finden Sie unter [Auftragsausführungen](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [GetJobRuns](#) in der AWS CLI Befehlsreferenz.

get-job

Das folgende Codebeispiel zeigt die Verwendung `get-job`.

AWS CLI

Informationen zu einem Auftrag abrufen

Das folgende Beispiel für `get-job` ruft Informationen zu einem Auftrag ab.

```

aws glue get-job \
  --job-name my-testing-job

```

Ausgabe:

```

{
  "Job": {
    "Name": "my-testing-job",
    "Role": "Glue_DefaultRole",

```

```

    "CreatedOn": 1602805698.167,
    "LastModifiedOn": 1602805698.167,
    "ExecutionProperty": {
      "MaxConcurrentRuns": 1
    },
    "Command": {
      "Name": "gluestreaming",
      "ScriptLocation": "s3://janetst-bucket-01/Scripts/test_script.scala",
      "PythonVersion": "2"
    },
    "DefaultArguments": {
      "--class": "GlueApp",
      "--job-language": "scala"
    },
    "MaxRetries": 0,
    "AllocatedCapacity": 10,
    "MaxCapacity": 10.0,
    "GlueVersion": "1.0"
  }
}

```

Weitere Informationen finden Sie unter [Aufträge](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [GetJobin](#) der AWS CLI Befehlsreferenz.

get-plan

Das folgende Codebeispiel zeigt die Verwendung `get-plan`.

AWS CLI

Um den generierten Code für die Zuordnung von Daten aus Quelltabellen zu Zieltabellen abzurufen

Im Folgenden wird der generierte Code für die Zuordnung von Spalten aus der Datenquelle zum Datenziel `get-plan` abgerufen.

```

aws glue get-plan --mapping '[ \
  { \
    "SourcePath": "sensorid", \
    "SourceTable": "anything", \
    "SourceType": "int", \
    "TargetPath": "sensorid", \

```

```

    "TargetTable":"anything", \
    "TargetType":"int" \
  }, \
  { \
    "SourcePath":"currenttemperature", \
    "SourceTable":"anything", \
    "SourceType":"int", \
    "TargetPath":"currenttemperature", \
    "TargetTable":"anything", \
    "TargetType":"int" \
  }, \
  { \
    "SourcePath":"status", \
    "SourceTable":"anything", \
    "SourceType":"string", \
    "TargetPath":"status", \
    "TargetTable":"anything", \
    "TargetType":"string" \
  ]' \
--source '{ \
  "DatabaseName":"tempdb", \
  "TableName":"s3-source" \
}' \
--sinks '[ \
  { \
    "DatabaseName":"tempdb", \
    "TableName":"my-s3-sink" \
  }]' \
--language "scala"
--endpoint https://glue.us-east-1.amazonaws.com
--output "text"

```

Ausgabe:

```

import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext

```

```
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // @type: DataSource
    // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx =
"datasource0"]
    // @return: datasource0
    // @inputs: []
    val datasource0 = glueContext.getCatalogSource(database = "tempdb",
tableName = "s3-source", redshiftTmpDir = "", transformationContext =
"datasource0").getDynamicFrame()
    // @type: ApplyMapping
    // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
"status", "string")], transformation_ctx = "applymapping1"]
    // @return: applymapping1
    // @inputs: [frame = datasource0]
    val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
"int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
"int"), ("status", "string", "status", "string")), caseSensitive = false,
transformationContext = "applymapping1")
    // @type: SelectFields
    // @args: [paths = ["sensorid", "currenttemperature", "status"],
transformation_ctx = "selectfields2"]
    // @return: selectfields2
    // @inputs: [frame = applymapping1]
    val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
"currenttemperature", "status"), transformationContext = "selectfields2")
    // @type: ResolveChoice
    // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-s3-
sink", transformation_ctx = "resolvechoice3"]
    // @return: resolvechoice3
    // @inputs: [frame = selectfields2]
    val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
Some("my-s3-sink"), transformationContext = "resolvechoice3")
    // @type: DataSink
```



```
// @args: [database = "tempdb", table_name = "my-s3-sink", transformation_ctx =
"datasink4"]
// @return: datasink4
// @inputs: [frame = resolvechoice3]
val datasink4 = glueContext.getCatalogSink(database = "tempdb",
tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
"datasink4").writeDynamicFrame(resolvechoice3)
    Job.commit()
}
}
```

Weitere Informationen finden Sie unter [Editing Scripts in AWS Glue](#) im AWS Glue Developer Guide.

- API-Einheiten finden Sie [GetPlan](#) in der AWS CLI Befehlsreferenz.

get-tables

Das folgende Codebeispiel zeigt die Verwendung `get-tables`.

AWS CLI

Die Definitionen einiger oder aller Tabellen in der angegebenen Datenbank auflisten

Das folgende Beispiel für `get-tables` gibt Informationen zu den Tabellen in der angegebenen Datenbank zurück.

```
aws glue get-tables --database-name 'tempdb'
```

Ausgabe:

```
{
  "TableList": [
    {
      "Name": "my-s3-sink",
      "DatabaseName": "tempdb",
      "CreateTime": 1602730539.0,
      "UpdateTime": 1602730539.0,
      "Retention": 0,
      "StorageDescriptor": {
        "Columns": [
          {
            "Name": "sensorid",
```

```
        "Type": "int"
      },
      {
        "Name": "currenttemperature",
        "Type": "int"
      },
      {
        "Name": "status",
        "Type": "string"
      }
    ],
    "Location": "s3://janetst-bucket-01/test-s3-output/",
    "Compressed": false,
    "NumberOfBuckets": 0,
    "SerdeInfo": {
      "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
    },
    "SortColumns": [],
    "StoredAsSubDirectories": false
  },
  "Parameters": {
    "classification": "json"
  },
  "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
  "IsRegisteredWithLakeFormation": false,
  "CatalogId": "007436865787"
},
{
  "Name": "s3-source",
  "DatabaseName": "tempdb",
  "CreateTime": 1602730658.0,
  "UpdateTime": 1602730658.0,
  "Retention": 0,
  "StorageDescriptor": {
    "Columns": [
      {
        "Name": "sensorid",
        "Type": "int"
      },
      {
        "Name": "currenttemperature",
        "Type": "int"
      }
    ]
  }
}
```

```

        "Name": "status",
        "Type": "string"
    }
],
"Location": "s3://janetst-bucket-01/",
"Compressed": false,
"NumberOfBuckets": 0,
"SortColumns": [],
"StoredAsSubDirectories": false
},
"Parameters": {
    "classification": "json"
},
"CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
"IsRegisteredWithLakeFormation": false,
"CatalogId": "007436865787"
},
{
    "Name": "test-kinesis-input",
    "DatabaseName": "tempdb",
    "CreateTime": 1601507001.0,
    "UpdateTime": 1601507001.0,
    "Retention": 0,
    "StorageDescriptor": {
        "Columns": [
            {
                "Name": "sensorid",
                "Type": "int"
            },
            {
                "Name": "currenttemperature",
                "Type": "int"
            },
            {
                "Name": "status",
                "Type": "string"
            }
        ]
    },
    "Location": "my-testing-stream",
    "Compressed": false,
    "NumberOfBuckets": 0,
    "SerdeInfo": {
        "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
    }
},

```

```
        "SortColumns": [],
        "Parameters": {
            "kinesisUrl": "https://kinesis.us-east-1.amazonaws.com",
            "streamName": "my-testing-stream",
            "typeOfData": "kinesis"
        },
        "StoredAsSubDirectories": false
    },
    "Parameters": {
        "classification": "json"
    },
    "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
    "IsRegisteredWithLakeFormation": false,
    "CatalogId": "007436865787"
}
]
}
```

Weitere Informationen finden Sie unter [Definieren von Tabellen im AWS Glue-Datenkatalog](#) im AWS Glue-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetTables](#) in der AWS CLI Befehlsreferenz.

start-crawler

Das folgende Codebeispiel zeigt die Verwendung `start-crawler`.

AWS CLI

Einen Crawler starten

Das folgende Beispiel für `start-crawler` startet einen Crawler.

```
aws glue start-crawler --name my-crawler
```

Ausgabe:

```
None
```

Weitere Informationen finden Sie unter [Definieren von Crawlern](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [StartCrawler](#) in der AWS CLI Befehlsreferenz.

start-job-run

Das folgende Codebeispiel zeigt die Verwendung `start-job-run`.

AWS CLI

Die Auftragsausführung starten

Das folgende Beispiel für `start-job-run` startet die Ausführung eines Auftrags.

```
aws glue start-job-run \  
  --job-name my-job
```

Ausgabe:

```
{  
  "JobRunId":  
  "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"  
}
```

Weitere Informationen finden Sie unter [Autorisieren von Aufträgen](#) im Entwicklerhandbuch für AWS Glue.

- API-Einheiten finden Sie [StartJobRun](#) in der AWS CLI Befehlsreferenz.

GuardDuty Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren GuardDuty.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-invitation

Das folgende Codebeispiel zeigt die Verwendung `accept-invitation`.

AWS CLI

Um eine Einladung anzunehmen, ein GuardDuty Mitgliedskonto in der aktuellen Region zu werden

Das folgende `accept-invitation` Beispiel zeigt, wie Sie eine Einladung annehmen, ein GuardDuty Mitgliedskonto in der aktuellen Region zu werden.

```
aws guardduty accept-invitation \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --master-id 123456789111 \  
  --invitation-id d6b94fb03a66ff665f7db8764example
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch [unter GuardDuty Konten auf Einladung verwalten](#).

- API-Einheiten finden Sie [AcceptInvitation](#) in der AWS CLI Befehlsreferenz.

archive-findings

Das folgende Codebeispiel zeigt, wie man es benutzt `archive-findings`.

AWS CLI

Um Ergebnisse in der aktuellen Region zu archivieren

Dieses Beispiel zeigt, wie Ergebnisse in der aktuellen Region archiviert werden.

```
aws guardduty archive-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch [unter GuardDuty Konten auf Einladung verwalten](#).

- APIEinzelheiten finden Sie [ArchiveFindings](#) in der AWS CLI Befehlsreferenz.

create-detector

Das folgende Codebeispiel zeigt, wie man es benutzt `create-detector`.

AWS CLI

Um es GuardDuty in der aktuellen Region zu aktivieren

Dieses Beispiel zeigt, wie ein neuer Detektor, der aktiviert wird GuardDuty, in der aktuellen Region erstellt wird. :

```
aws guardduty create-detector \  
  --enable
```

Ausgabe:

```
{  
  "DetectorId": "b6b992d6d2f48e64bc59180bfexample"  
}
```

Weitere Informationen finden Sie GuardDuty im GuardDuty Benutzerhandbuch unter [Amazon aktivieren](#).

- APIEinzelheiten finden Sie [CreateDetector](#) in der AWS CLI Befehlsreferenz.

create-filter

Das folgende Codebeispiel zeigt, wie man es benutzt `create-filter`.

AWS CLI

Um einen neuen Filter für die aktuelle Region zu erstellen

In diesem Beispiel wird ein Filter erstellt, der allen Portscan-Ergebnissen entspricht, die beispielsweise aus einem bestimmten Bild erstellt wurden. :

```
aws guardduty create-filter \  
  --enable
```

```
--detector-id b6b992d6d2f48e64bc59180bfexample \  
--action ARCHIVE \  
--name myFilter \  
--finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/  
Portscan"]}, "resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

Ausgabe:

```
{  
  "Name": "myFilter"  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Filtern von Ergebnissen](#).

- API-Einheiten finden Sie [CreateFilter](#) in der AWS CLI Befehlsreferenz.

create-ip-set

Das folgende Codebeispiel zeigt, wie man es benutzt `create-ip-set`.

AWS CLI

Um einen vertrauenswürdigen IP-Satz zu erstellen

Im folgenden `create-ip-set` Beispiel wird ein vertrauenswürdiger IP-Satz in der aktuellen Region erstellt und aktiviert.

```
aws guardduty create-ip-set \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --name new-ip-set \  
  --format TXT \  
  --location s3://AWSDOC-EXAMPLE-BUCKET/customtrustlist.csv \  
  --activate
```

Ausgabe:

```
{  
  "IpSetId": "d4b94fc952d6912b8f3060768example"  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Arbeiten mit Listen vertrauenswürdiger IP-Adressen und Bedrohungslisten](#).

- API-Einheiten finden Sie [CreatepSet](#) unter AWS CLI Befehlsreferenz.

create-members

Das folgende Codebeispiel zeigt, wie man es benutzt `create-members`.

AWS CLI

Um ein neues Mitglied mit Ihrem GuardDuty Hauptkonto in der aktuellen Region zu verknüpfen.

Dieses Beispiel zeigt, wie Mitgliedskonten so verknüpft werden, dass sie vom Girokonto als GuardDuty Hauptkonto verwaltet werden.

```
aws guardduty create-members
  --detector-id b6b992d6d2f48e64bc59180bfexample \
  --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

Ausgabe:

```
{
  "UnprocessedAccounts": []
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Mehrere Konten verwalten](#).

- API-Einheiten finden Sie [CreateMembers](#) in der AWS CLI Befehlsreferenz.

create-publishing-destination

Das folgende Codebeispiel zeigt, wie man es benutzt `create-publishing-destination`.

AWS CLI

Um ein Veröffentlichungsziel zu erstellen, in das GuardDuty Ergebnisse in der aktuellen Region exportiert werden sollen.

Dieses Beispiel zeigt, wie ein Veröffentlichungsziel für GuardDuty Ergebnisse erstellt wird.

```
aws guardduty create-publishing-destination \
```

```
--detector-id b6b992d6d2f48e64bc59180bfexample \  
--destination-type S3 \  
--destination-  
properties DestinationArn=arn:aws:s3:::yourbucket,KmsKeyArn=arn:aws:kms:us-  
west-1:111122223333:key/84cee9c5-dea1-401a-ab6d-e1de7example
```

Ausgabe:

```
{  
  "DestinationId": "46b99823849e1bbc242dfbe3cexample"  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Ergebnisse exportieren](#).

- API-Einheiten finden Sie [CreatePublishingDestination](#) unter AWS CLI Befehlsreferenz.

create-sample-findings

Das folgende Codebeispiel zeigt, wie man es benutzt `create-sample-findings`.

AWS CLI

Um GuardDuty Beispielergebnisse in der aktuellen Region zu erstellen.

Dieses Beispiel zeigt, wie ein Stichprobenergebnis der angegebenen Typen erstellt wird.

```
aws guardduty create-sample-findings \  
--detector-id b6b992d6d2f48e64bc59180bfexample \  
--finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Beispielergebnisse](#) im GuardDuty Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSampleFindings](#) in der AWS CLI Befehlsreferenz.

create-threat-intel-set

Das folgende Codebeispiel zeigt, wie man es benutzt `create-threat-intel-set`.

AWS CLI

Um einen neuen Bedrohungsinformationssatz in der aktuellen Region zu erstellen.

Dieses Beispiel zeigt, wie ein Bedrohungsinformations-Set hochgeladen GuardDuty und sofort aktiviert wird.

```
aws guardduty create-threat-intel-set \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --name myThreatSet \  
  --format TXT \  
  --location s3://EXAMPLEBUCKET/threatlist.csv \  
  --activate
```

Ausgabe:

```
{  
  "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Vertrauenswürdige IP-Adressen und Bedrohungslisten](#).

- API-Einheiten finden Sie [CreateThreatIntelSet](#) in der AWS CLI Befehlsreferenz.

decline-invitations

Das folgende Codebeispiel zeigt, wie man es benutzt decline-invitations.

AWS CLI

Um eine Einladung abzulehnen, GuardDuty von einem anderen Konto in der aktuellen Region verwalten zu lassen.

Dieses Beispiel zeigt, wie Sie eine Einladung zur Mitgliedschaft ablehnen können.

```
aws guardduty decline-invitations \  
  --account-ids 111122223333
```

Ausgabe:

```
{  
  "UnprocessedAccounts": []  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch [unter GuardDuty Konten auf Einladung verwalten](#).

- API-Einheiten finden Sie [DeclineInvitations](#) in der AWS CLI Befehlsreferenz.

delete-detector

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-detector`.

AWS CLI

Um einen Detektor in der aktuellen Region zu löschen und zu deaktivieren GuardDuty.

Dieses Beispiel zeigt, wie ein Detektor gelöscht wird. Wenn dies erfolgreich ist, wird dies GuardDuty in der Region deaktiviert, die diesem Detektor zugeordnet ist.

```
aws guardduty delete-detector \  
  --detector-id b6b992d6d2f48e64bc59180bfexample
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie GuardDuty im GuardDuty Benutzerhandbuch unter [Sperrern oder Deaktivieren](#).

- API-Einheiten finden Sie [DeleteDetector](#) in der AWS CLI Befehlsreferenz.

delete-filter

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-filter`.

AWS CLI

Um einen vorhandenen Filter in der aktuellen Region zu löschen

Dieses Beispiel zeigt, wie ein Filter erstellt und gelöscht wird.

```
aws guardduty delete-filter \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --filter-name byebyeFilter
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Filtern von Ergebnissen](#).

- API-Einheiten finden Sie [DeleteFilter](#) in der AWS CLI Befehlsreferenz.

disable-organization-admin-account

Das folgende Codebeispiel zeigt, wie man es benutzt `disable-organization-admin-account`.

AWS CLI

Um ein Konto als delegierter Administrator für Ihre GuardDuty Organisation zu entfernen

Dieses Beispiel zeigt, wie Sie ein Konto als delegierter Administrator für entfernen. GuardDuty

```
aws guardduty disable-organization-admin-account \  
  --admin-account-id 111122223333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Konten bei AWS Organisationen verwalten](#).

- API-Einheiten finden Sie [DisableOrganizationAdminAccount](#) in der AWS CLI Befehlsreferenz.

disassociate-from-master-account

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-from-master-account`.

AWS CLI

Um die Verbindung zu Ihrem aktuellen Hauptkonto in der aktuellen Region zu trennen

Im folgenden `disassociate-from-master-account` Beispiel wird Ihr Konto vom aktuellen GuardDuty Hauptkonto in der aktuellen AWS Region getrennt.

```
aws guardduty disassociate-from-master-account \  
  --detector-id d4b040365221be2b54a6264dcexample
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [im GuardDuty Benutzerhandbuch unter Grundlegendes zur Beziehung zwischen GuardDuty Master- und Mitgliedskonten](#).

- API-Einheiten finden Sie [DisassociateFromMasterAccount](#) unter AWS CLI Befehlsreferenz.

get-detector

Das folgende Codebeispiel zeigt, wie man es benutzt `get-detector`.

AWS CLI

Um Details zu einem bestimmten Detektor abzurufen

Im folgenden `get-detector` Beispiel werden die Konfigurationsdetails des angegebenen Detektors angezeigt.

```
aws guardduty get-detector \
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Ausgabe:

```
{
  "Status": "ENABLED",
  "ServiceRole": "arn:aws:iam::111122223333:role/aws-service-role/guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDuty",
  "Tags": {},
  "FindingPublishingFrequency": "SIX_HOURS",
  "UpdatedAt": "2018-11-07T03:24:22.938Z",
  "CreatedAt": "2017-12-22T22:51:31.940Z"
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Konzepte und Terminologie](#).

- API-Einheiten finden Sie [GetDetector](#) in der AWS CLI Befehlsreferenz.

get-findings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-findings`.

AWS CLI

Beispiel 1: Um die Details eines bestimmten Ergebnisses abzurufen

Im folgenden `get-findings` Beispiel werden die vollständigen JSON Ergebnisdetails des angegebenen Ergebnisses abgerufen.

```
aws guardduty get-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-id 1ab92989eaf0e742df4a014d5example
```

Ausgabe:

```
{  
  "Findings": [  
    {  
      "Resource": {  
        "ResourceType": "AccessKey",  
        "AccessKeyDetails": {  
          "UserName": "testuser",  
          "UserType": "IAMUser",  
          "PrincipalId": "AIDACKCEVSQ6C2EXAMPLE",  
          "AccessKeyId": "ASIASZ4SI7REEEXAMPLE"  
        }  
      },  
      "Description": "APIs commonly used to discover the users, groups,  
policies and permissions in an account, was invoked by IAM principal testuser under  
unusual circumstances. Such activity is not typically seen from this principal.",  
      "Service": {  
        "Count": 5,  
        "Archived": false,  
        "ServiceName": "guardduty",  
        "EventFirstSeen": "2020-05-26T22:02:24Z",  
        "ResourceRole": "TARGET",  
        "EventLastSeen": "2020-05-26T22:33:55Z",  
        "DetectorId": "d4b040365221be2b54a6264dcexample",  
        "Action": {  
          "ActionType": "AWS_API_CALL",  
          "AwsApiCallAction": {  
            "RemoteIpDetails": {  
              "GeoLocation": {  
                "Lat": 51.5164,  
                "Lon": -0.093  
              },  
              "City": {  
                "CityName": "London"  
              }  
            }  
          }  
        }  
      }  
    ]  
  }
```

```

        "IpAddressV4": "52.94.36.7",
        "Organization": {
            "Org": "Amazon.com",
            "Isp": "Amazon.com",
            "Asn": "16509",
            "AsnOrg": "AMAZON-02"
        },
        "Country": {
            "CountryName": "United Kingdom"
        }
    },
    "Api": "ListPolicyVersions",
    "ServiceName": "iam.amazonaws.com",
    "CallerType": "Remote IP"
}
}
},
"Title": "Unusual user permission reconnaissance activity by testuser.",
"Type": "Recon:IAMUser/UserPermissions",
"Region": "us-east-1",
"Partition": "aws",
"Arn": "arn:aws:guardduty:us-east-1:111122223333:detector/
d4b040365221be2b54a6264dcexample/finding/1ab92989eaf0e742df4a014d5example",
"UpdatedAt": "2020-05-26T22:55:21.703Z",
"SchemaVersion": "2.0",
"Severity": 5,
"Id": "1ab92989eaf0e742df4a014d5example",
"CreatedAt": "2020-05-26T22:21:48.385Z",
"AccountId": "111122223333"
}
]
}

```

Weitere Informationen finden Sie unter [Ergebnisse](#) im GuardDuty Benutzerhandbuch.

- API-Einheiten finden Sie [GetFindings](#) in der AWS CLI Befehlsreferenz.

get-ip-set

Das folgende Codebeispiel zeigt, wie man es benutzt `get-ip-set`.

AWS CLI

Um die Liste aufzulisten, rufen Sie Details zu einem bestimmten vertrauenswürdigen IP-Satz ab

Das folgende `get-ip-set` Beispiel zeigt den Status und die Details des angegebenen vertrauenswürdigen IP-Sets.

```
aws guardduty get-ip-set \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --ip-set-id d4b94fc952d6912b8f3060768example
```

Ausgabe:

```
{  
  "Status": "ACTIVE",  
  "Location": "s3://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/  
customlist.csv",  
  "Tags": {},  
  "Format": "TXT",  
  "Name": "test-ip-set"  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Arbeiten mit Listen vertrauenswürdiger IP-Adressen und Bedrohungslisten](#).

- API-Einheiten finden Sie [GetIpSet](#) unter AWS CLI Befehlsreferenz.

get-master-account

Das folgende Codebeispiel zeigt, wie man es benutzt `get-master-account`.

AWS CLI

Um Details zu Ihrem Hauptkonto in der aktuellen Region abzurufen

Im folgenden `get-master-account` Beispiel werden der Status und die Details des Hauptkontos angezeigt, das Ihrem Melder in der aktuellen Region zugeordnet ist.

```
aws guardduty get-master-account \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Ausgabe:

```
{  
  "Master": {
```

```
    "InvitationId": "04b94d9704854a73f94e061e8example",
    "InvitedAt": "2020-06-09T22:23:04.970Z",
    "RelationshipStatus": "Enabled",
    "AccountId": "123456789111"
  }
}
```

Weitere Informationen finden Sie [im GuardDuty Benutzerhandbuch unter Grundlegendes zur Beziehung zwischen GuardDuty Master- und Mitgliedskonten](#).

- API-Einheiten finden Sie [GetMasterAccount](#) unter AWS CLI Befehlsreferenz.

list-detectors

Das folgende Codebeispiel zeigt, wie man es benutzt `list-detectors`.

AWS CLI

Um die verfügbaren Melder in der aktuellen Region aufzulisten

Das folgende `list-detectors` Beispiel listet die verfügbaren Melder in Ihrer aktuellen AWS Region auf.

```
aws guardduty list-detectors
```

Ausgabe:

```
{
  "DetectorIds": [
    "12abc34d567e8fa901bc2d34eexample"
  ]
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Konzepte und Terminologie](#).

- API-Einheiten finden Sie [ListDetectors](#) in der AWS CLI Befehlsreferenz.

list-findings

Das folgende Codebeispiel zeigt, wie man es benutzt `list-findings`.

AWS CLI

Beispiel 1: Um alle Ergebnisse für die aktuelle Region aufzulisten

Im folgenden `list-findings` Beispiel wird eine Liste aller Daten `findingIds` für die aktuelle Region angezeigt, sortiert nach Schweregrad vom höchsten zum niedrigsten.

```
aws guardduty list-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --sort-criteria '{"AttributeName": "severity", "OrderBy": "DESC"}'
```

Ausgabe:

```
{  
  "FindingIds": [  
    "04b8ab50fd29c64fc771b232dexample",  
    "5ab8ab50fd21373735c826d3aexample",  
    "90b93de7aba69107f05bbe60bexample",  
    ...  
  ]  
}
```

Weitere Informationen finden Sie unter [Ergebnisse](#) im GuardDuty Benutzerhandbuch.

Beispiel 2: Um Ergebnisse für die aktuelle Region aufzulisten, die bestimmten Suchkriterien entsprechen

Im folgenden `list-findings` Beispiel wird eine Liste aller Ergebnisse angezeigt `findingIds`, die einem bestimmten Ergebnistyp entsprechen.

```
aws guardduty list-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-criteria '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/  
SSHBruteForce"]}}}'
```

Ausgabe:

```
{  
  "FindingIds": [  
    "90b93de7aba69107f05bbe60bexample",  
    "6eb9430d7023d30774d6f05e3example",  
  ]  
}
```

```

    "2eb91a2d060ac9a21963a5848example",
    "44b8ab50fd2b0039a9e48f570example",
    "9eb8ab4cd2b7e5b66ba4f5e96example",
    "e0b8ab3a38e9b0312cc390ceeexample"
  ]
}

```

Weitere Informationen finden Sie unter [Ergebnisse](#) im GuardDuty Benutzerhandbuch.

Beispiel 3: Um Ergebnisse für die aktuelle Region aufzulisten, die einem bestimmten Satz von in einer JSON Datei definierten Suchkriterien entsprechen

Im folgenden `list-findings` Beispiel wird eine Liste aller Daten angezeigt `findingIds`, die nicht archiviert wurden und an denen der IAM Benutzer mit dem Namen „testuser“ beteiligt ist, wie in einer JSON Datei angegeben.

```

aws guardduty list-findings \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --finding-criteria file://myfile.json

```

Inhalt von `myfile.json`:

```

{"Criterion": {
  "resource.accessKeyDetails.userName":{
    "Eq":[
      "testuser"
    ]
  },
  "service.archived": {
    "Eq": [
      "false"
    ]
  }
}
}

```

Ausgabe:

```

{
  "FindingIds": [
    "1ab92989eaf0e742df4a014d5example"
  ]
}

```

```
}
```

Weitere Informationen finden Sie unter [Ergebnisse](#) im GuardDuty Benutzerhandbuch.

- API-Einheiten finden Sie [ListFindings](#) in der AWS CLI Befehlsreferenz.

list-invitations

Das folgende Codebeispiel zeigt, wie man es benutzt `list-invitations`.

AWS CLI

Um Details zu Ihren Einladungen aufzulisten, ein Mitgliedskonto in der aktuellen Region zu werden

Im folgenden `list-invitations` Beispiel werden Details und Status Ihrer Einladungen zu einem GuardDuty Mitgliedskonto in der aktuellen Region aufgeführt.

```
aws guardduty list-invitations
```

Ausgabe:

```
{
  "Invitations": [
    {
      "InvitationId": "d6b94fb03a66ff665f7db8764example",
      "InvitedAt": "2020-06-10T17:56:38.221Z",
      "RelationshipStatus": "Invited",
      "AccountId": "123456789111"
    }
  ]
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch [unter GuardDuty Konten auf Einladung verwalten](#).

- API-Einheiten finden Sie [ListInvitations](#) in der AWS CLI Befehlsreferenz.

list-ip-sets

Das folgende Codebeispiel zeigt, wie man es benutzt `list-ip-sets`.

AWS CLI

Um vertrauenswürdige IP-Sets in der aktuellen Region aufzulisten

Das folgende `list-ip-sets` Beispiel listet die vertrauenswürdigen IP-Sets in Ihrer aktuellen AWS Region auf.

```
aws guardduty list-ip-sets \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Ausgabe:

```
{  
  "IpSetIds": [  
    "d4b94fc952d6912b8f3060768example"  
  ]  
}
```

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Arbeiten mit Listen vertrauenswürdiger IP-Adressen und Bedrohungslisten](#).

- API-Einheiten finden Sie [ListIpSets](#) unter AWS CLI Befehlsreferenz.

list-members

Das folgende Codebeispiel zeigt, wie man es benutzt `list-members`.

AWS CLI

Um alle Mitglieder in der aktuellen Region aufzulisten

Das folgende `list-members` Beispiel listet alle Mitgliedskonten und ihre Details für die aktuelle Region auf.

```
aws guardduty list-members \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Ausgabe:

```
{
```

```
"Members": [  
  {  
    "RelationshipStatus": "Enabled",  
    "InvitedAt": "2020-06-09T22:49:00.910Z",  
    "MasterId": "123456789111",  
    "DetectorId": "7ab8b2f61b256c87f793f6a86example",  
    "UpdatedAt": "2020-06-09T23:08:22.512Z",  
    "Email": "your+member@example.com",  
    "AccountId": "123456789222"  
  }  
]  
}
```

Weitere Informationen finden Sie [im GuardDuty Benutzerhandbuch unter Grundlegendes zur Beziehung zwischen GuardDuty Master- und Mitgliedskonten](#).

- APIEinzelheiten finden Sie [ListMembers](#) unter AWS CLI Befehlsreferenz.

update-ip-set

Das folgende Codebeispiel zeigt, wie man es benutzt `update-ip-set`.

AWS CLI

Um einen vertrauenswürdigen IP-Satz zu aktualisieren

Das folgende `update-ip-set` Beispiel zeigt, wie die Details eines vertrauenswürdigen IP-Sets aktualisiert werden.

```
aws guardduty update-ip-set \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --ip-set-id d4b94fc952d6912b8f3060768example \  
  --location https://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/  
customtrustlist2.csv
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im GuardDuty Benutzerhandbuch unter [Arbeiten mit Listen vertrauenswürdiger IP-Adressen und Bedrohungslisten](#).

- APIEinzelheiten finden Sie [UpdateIpSet](#) unter AWS CLI Befehlsreferenz.

AWS Health Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Health.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-affected-entities

Das folgende Codebeispiel zeigt die Verwendung `describe-affected-entities`.

AWS CLI

Um die Entitäten aufzulisten, die von einem bestimmten AWS Gesundheitsereignis betroffen sind

Das folgende `describe-affected-entities` Beispiel listet die Entitäten auf, die von dem angegebenen AWS Integritätsereignis betroffen sind. Bei diesem Ereignis handelt es sich um eine Abrechnungsbenachrichtigung für das AWS Konto.

```
aws health describe-affected-entities \  
  --filter "eventArns=arn:aws:health:global::event/BILLING/  
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-  
EXAMPLE11145" \  
  --region us-east-1
```

Ausgabe:

```
{  
  "entities": [  
    {
```



```

    "entityArn": "arn:aws:health:global:123456789012:entity/
EXAMPLEimSMoULmWHpb",
    "eventArn": "arn:aws:health:global::event/BILLING/
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-
EXAMPLE11145",
    "entityValue": "AWS_ACCOUNT",
    "awsAccountId": "123456789012",
    "lastUpdatedTime": 1588356454.08
  }
]
}

```

Weitere Informationen finden Sie im AWS Health-Benutzerhandbuch unter [Ereignisprotokoll](#).

- API-Einheiten finden Sie [DescribeAffectedEntities](#) in der AWS CLI Befehlsreferenz.

describe-event-details

Das folgende Codebeispiel zeigt die Verwendung `describe-event-details`.

AWS CLI

Um Informationen über ein AWS Gesundheitsereignis aufzulisten

Das folgende `describe-event-details` Beispiel listet Informationen über das angegebene AWS Gesundheitsereignis auf.

```

aws health describe-event-details \
  --event-arns "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111" \
  --region us-east-1

```

Ausgabe:

```

{
  "successfulSet": [
    {
      "event": {
        "arn": "arn:aws:health:us-east-1::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
        "eventTypeCategory": "issue",

```

```

        "region": "us-east-1",
        "startTime": 1587462325.096,
        "endTime": 1587464204.774,
        "lastUpdatedTime": 1587464204.865,
        "statusCode": "closed"
    },
    "eventDescription": {
        "latestDescription": "[RESOLVED] Increased API Error Rates and Latencies\n\n[02:45 AM PDT] We are investigating increased API error rates and latencies in the US-EAST-1 Region.\n\n[03:16 AM PDT] Between 2:10 AM and 2:59 AM PDT we experienced increased API error rates and latencies in the US-EAST-1 Region. The issue has been resolved and the service is operating normally."
    }
}
],
"failedSet": []
}

```

Weitere Informationen finden Sie im AWS Health-Benutzerhandbuch im [Bereich mit den Ereignisdetails](#).

- API-Einheiten finden Sie [DescribeEventDetails](#) in der AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Beispiel 1: Um AWS Gesundheitsereignisse aufzulisten

Das folgende `describe-events` Beispiel listet aktuelle AWS Gesundheitsereignisse auf.

```
aws health describe-events \
  --region us-east-1
```

Ausgabe:

```
{
  "events": [
    {
      "arn": "arn:aws:health:us-west-1::event/ECS/AWS_ECS_OPERATIONAL_ISSUE/AWS_ECS_OPERATIONAL_ISSUE_KWQPY_EXAMPLE111",

```

```
    "service": "ECS",
    "eventTypeCode": "AWS_ECS_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "us-west-1",
    "startTime": 1589077890.53,
    "endTime": 1589086345.597,
    "lastUpdatedTime": 1589086345.905,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:global::event/BILLING/AWS_BILLING_NOTIFICATION/
AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-EXAMPLE1118b",
    "service": "BILLING",
    "eventTypeCode": "AWS_BILLING_NOTIFICATION",
    "eventTypeCategory": "accountNotification",
    "region": "global",
    "startTime": 1588356000.0,
    "lastUpdatedTime": 1588356524.358,
    "statusCode": "open",
    "eventScopeCode": "ACCOUNT_SPECIFIC"
  },
  {
    "arn": "arn:aws:health:us-west-2::event/
CLOUDFORMATION/AWS_CLOUDFORMATION_OPERATIONAL_ISSUE/
AWS_CLOUDFORMATION_OPERATIONAL_ISSUE_OHTWY_EXAMPLE111",
    "service": "CLOUDFORMATION",
    "eventTypeCode": "AWS_CLOUDFORMATION_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "us-west-2",
    "startTime": 1588279630.761,
    "endTime": 1588284650.0,
    "lastUpdatedTime": 1588284691.941,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:ap-northeast-1::event/LAMBDA/
AWS_LAMBDA_OPERATIONAL_ISSUE/AWS_LAMBDA_OPERATIONAL_ISSUE_JZDND_EXAMPLE111",
    "service": "LAMBDA",
    "eventTypeCode": "AWS_LAMBDA_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "ap-northeast-1",
    "startTime": 1587379534.08,
```

```
    "endTime": 1587391771.0,
    "lastUpdatedTime": 1587395689.316,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
    "service": "EC2",
    "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "us-east-1",
    "startTime": 1586473044.284,
    "endTime": 1586479706.091,
    "lastUpdatedTime": 1586479706.153,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:global::event/SECURITY/AWS_SECURITY_NOTIFICATION/
AWS_SECURITY_NOTIFICATION_42007387-8129-42da-8c88-EXAMPLE11139",
    "service": "SECURITY",
    "eventTypeCode": "AWS_SECURITY_NOTIFICATION",
    "eventTypeCategory": "accountNotification",
    "region": "global",
    "startTime": 1585674000.0,
    "lastUpdatedTime": 1585674004.132,
    "statusCode": "open",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:global::event/CLOUDFRONT/
AWS_CLOUDFRONT_OPERATIONAL_ISSUE/AWS_CLOUDFRONT_OPERATIONAL_ISSUE_FRQXG_EXAMPLE111",
    "service": "CLOUDFRONT",
    "eventTypeCode": "AWS_CLOUDFRONT_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "global",
    "startTime": 1585610898.589,
    "endTime": 1585617671.0,
    "lastUpdatedTime": 1585620638.869,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
```

```

        "arn": "arn:aws:health:us-east-1::event/SES/AWS_SES_OPERATIONAL_ISSUE/
AWS_SES_OPERATIONAL_ISSUE_URNDF_EXAMPLE111",
        "service": "SES",
        "eventTypeCode": "AWS_SES_OPERATIONAL_ISSUE",
        "eventTypeCategory": "issue",
        "region": "us-east-1",
        "startTime": 1585342008.46,
        "endTime": 1585344017.0,
        "lastUpdatedTime": 1585344355.989,
        "statusCode": "closed",
        "eventScopeCode": "PUBLIC"
    },
    {
        "arn": "arn:aws:health:global::event/IAM/
AWS_IAM_OPERATIONAL_NOTIFICATION/
AWS_IAM_OPERATIONAL_NOTIFICATION_b6771c34-6ecd-4aea-9d3e-EXAMPLE1117e",
        "service": "IAM",
        "eventTypeCode": "AWS_IAM_OPERATIONAL_NOTIFICATION",
        "eventTypeCategory": "accountNotification",
        "region": "global",
        "startTime": 1584978300.0,
        "lastUpdatedTime": 1584978553.572,
        "statusCode": "open",
        "eventScopeCode": "ACCOUNT_SPECIFIC"
    },
    {
        "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
        "eventTypeCategory": "issue",
        "region": "ap-southeast-2",
        "startTime": 1583881487.483,
        "endTime": 1583885056.785,
        "lastUpdatedTime": 1583885057.052,
        "statusCode": "closed",
        "eventScopeCode": "PUBLIC"
    }
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS Personal Health Dashboard](#) im AWS Health-Benutzerhandbuch.

Beispiel 2: So listen Sie AWS Gesundheitsereignisse nach Service und Ereignisstatuscode auf

Im folgenden `describe-events` Beispiel werden AWS Health-Ereignisse für Amazon Elastic Compute Cloud (AmazonEC2) aufgeführt, bei denen der Ereignisstatus geschlossen ist.

```
aws health describe-events \  
  --filter "services=EC2,eventStatusCodes=closed"
```

Ausgabe:

```
{  
  "events": [  
    {  
      "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/  
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",  
      "service": "EC2",  
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",  
      "eventTypeCategory": "issue",  
      "region": "us-east-1",  
      "startTime": 1587462325.096,  
      "endTime": 1587464204.774,  
      "lastUpdatedTime": 1587464204.865,  
      "statusCode": "closed",  
      "eventScopeCode": "PUBLIC"  
    },  
    {  
      "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/  
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",  
      "service": "EC2",  
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",  
      "eventTypeCategory": "issue",  
      "region": "us-east-1",  
      "startTime": 1586473044.284,  
      "endTime": 1586479706.091,  
      "lastUpdatedTime": 1586479706.153,  
      "statusCode": "closed",  
      "eventScopeCode": "PUBLIC"  
    },  
    {  
      "arn": "arn:aws:health:ap-southeast-2::event/EC2/  
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",  
      "service": "EC2",  
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
```

```
    "eventTypeCategory": "issue",
    "region": "ap-southeast-2",
    "startTime": 1583881487.483,
    "endTime": 1583885056.785,
    "lastUpdatedTime": 1583885057.052,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  }
]
```

Weitere Informationen finden Sie unter [Erste Schritte mit dem AWS Personal Health Dashboard](#) im AWS Health-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

HealthImaging Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren HealthImaging.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

copy-image-set

Das folgende Codebeispiel zeigt die Verwendung `copy-image-set`.

AWS CLI

Beispiel 1: Um einen Bilddatensatz ohne Ziel zu kopieren.

Im folgenden `copy-image-set` Beispiel wird eine doppelte Kopie eines Bilddatensatzes ohne Ziel erstellt.

```
aws medical-imaging copy-image-set \
  --datastore-id 12345678901234567890123456789012 \
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

Ausgabe:

```
{
  "destinationImageSetProperties": {
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "COPYING",
    "updatedAt": 1680042357.432,
    "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
    "imageSetState": "LOCKED",
    "createdAt": 1680042357.432
  },
  "sourceImageSetProperties": {
    "latestVersionId": "1",
    "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
    "updatedAt": 1680042357.432,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436
  },
  "datastoreId": "12345678901234567890123456789012"
}
```

Beispiel 2: Um einen Bilddatensatz mit einem Ziel zu kopieren.

Im folgenden `copy-image-set` Beispiel wird eine doppelte Kopie eines Bilddatensatzes mit einem Ziel erstellt.

```
aws medical-imaging copy-image-set \
  --datastore-id 12345678901234567890123456789012 \
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" },
  "destinationImageSet": { "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
  "latestVersionId": "1" } }'
```


Ausgabe:

```
{
  "destinationImageSetProperties": {
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "COPYING",
    "updatedAt": 1680042505.135,
    "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
    "imageSetState": "LOCKED",
    "createdAt": 1680042357.432
  },
  "sourceImageSetProperties": {
    "latestVersionId": "1",
    "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
    "updatedAt": 1680042505.135,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436
  },
  "datastoreId": "12345678901234567890123456789012"
}
```

Beispiel 3: Um eine Teilmenge von Instanzen aus einem Quell-Bildsatz in einen Ziel-Bildsatz zu kopieren.

Im folgenden `copy-image-set` Beispiel wird eine DICOM Instanz aus dem Quell-Bildsatz in den Ziel-Bildsatz kopiert. Der `Force`-Parameter dient dazu, Inkonsistenzen in den Attributen auf Patient-, Studien- und Serienebene zu überschreiben.

```
aws medical-imaging copy-image-set \
  --datastore-id 12345678901234567890123456789012 \
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId":
  "1", "DICOMCopies": {"copiableAttributes": {"SchemaVersion": "1.1", "Study":
  {"Series": {"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3666.0":
  {"Instances":
  {"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3669.0":
  {}}}}}}}}, "destinationImageSet": {"imageSetId":
  "b9eb50d8ee682eb9fcf4acbf92f62bb7", "latestVersionId": "1"}}' \
  --force
```

Ausgabe:

```
{
  "destinationImageSetProperties": {
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "COPYING",
    "updatedAt": 1680042505.135,
    "imageSetId": "b9eb50d8ee682eb9fcf4acbf92f62bb7",
    "imageSetState": "LOCKED",
    "createdAt": 1680042357.432
  },
  "sourceImageSetProperties": {
    "latestVersionId": "1",
    "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
    "updatedAt": 1680042505.135,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436
  },
  "datastoreId": "12345678901234567890123456789012"
}
```

Weitere Informationen finden Sie im [AWS HealthImaging Entwicklerhandbuch unter Kopieren eines Bilddatensatzes](#).

- API-Einheiten finden Sie [CopyImageSet](#) in der AWS CLI Befehlsreferenz.

create-datastore

Das folgende Codebeispiel zeigt die Verwendung `create-datastore`.

AWS CLI

Um einen Datenspeicher zu erstellen

Das folgende `create-datastore` Codebeispiel erstellt einen Datenspeicher mit dem Namen `my-datastore`.

```
aws medical-imaging create-datastore \
  --datastore-name "my-datastore"
```

Ausgabe:

```
{
```

```
"datastoreId": "12345678901234567890123456789012",  
"datastoreStatus": "CREATING"  
}
```

Weitere Informationen finden Sie unter [Erstellen eines AWS HealthImaging Datenspeichers](#) im Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateDatastore](#) in der AWS CLI Befehlsreferenz.

delete-datastore

Das folgende Codebeispiel zeigt die Verwendung `delete-datastore`.

AWS CLI

Um einen Datenspeicher zu löschen

Das folgende `delete-datastore` Codebeispiel löscht einen Datenspeicher.

```
aws medical-imaging delete-datastore \  
--datastore-id "12345678901234567890123456789012"
```

Ausgabe:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "datastoreStatus": "DELETING"  
}
```

Weitere Informationen finden Sie unter [Löschen eines AWS HealthImaging Datenspeichers](#) im Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteDatastore](#) in der AWS CLI Befehlsreferenz.

delete-image-set

Das folgende Codebeispiel zeigt die Verwendung `delete-image-set`.

AWS CLI

Um einen Bildsatz zu löschen

Das folgende `delete-image-set` Codebeispiel löscht einen Bildsatz.

```
aws medical-imaging delete-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Ausgabe:

```
{  
  "imageSetWorkflowStatus": "DELETING",  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "datastoreId": "12345678901234567890123456789012"  
}
```

Weitere Informationen finden Sie unter [Löschen eines Bildsatzes](#) im AWS HealthImaging Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteImageSet](#) in der AWS CLI Befehlsreferenz.

get-datastore

Das folgende Codebeispiel zeigt die Verwendung `get-datastore`.

AWS CLI

Um die Eigenschaften eines Datenspeichers abzurufen

Das folgende `get-datastore` Codebeispiel ruft die Eigenschaften eines Datenspeichers ab.

```
aws medical-imaging get-datastore \  
  --datastore-id 12345678901234567890123456789012
```

Ausgabe:

```
{  
  "datastoreProperties": {  
    "datastoreId": "12345678901234567890123456789012",  
    "datastoreName": "TestDatastore123",  
    "datastoreStatus": "ACTIVE",  
    "datastoreArn": "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012",
```

```

    "createdAt": "2022-11-15T23:33:09.643000+00:00",
    "updatedAt": "2022-11-15T23:33:09.643000+00:00"
  }
}

```

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Abrufen von Datenspeichereigenschaften](#).

- API-Einheiten finden Sie [GetDatastore](#) in der AWS CLI Befehlsreferenz.

get-dicom-import-job

Das folgende Codebeispiel zeigt die Verwendung `get-dicom-import-job`.

AWS CLI

Um die Eigenschaften eines DICOM-Importauftrags abzurufen

Im folgenden `get-dicom-import-job` Codebeispiel werden die Eigenschaften eines DICOM-Importauftrags abgerufen.

```

aws medical-imaging get-dicom-import-job \
  --datastore-id "12345678901234567890123456789012" \
  --job-id "09876543210987654321098765432109"

```

Ausgabe:

```

{
  "jobProperties": {
    "jobId": "09876543210987654321098765432109",
    "jobName": "my-job",
    "jobStatus": "COMPLETED",
    "datastoreId": "12345678901234567890123456789012",
    "dataAccessRoleArn": "arn:aws:iam::123456789012:role/
ImportJobDataAccessRole",
    "endedAt": "2022-08-12T11:29:42.285000+00:00",
    "submittedAt": "2022-08-12T11:28:11.152000+00:00",
    "inputS3Uri": "s3://medical-imaging-dicom-input/dicom_input/",
    "outputS3Uri": "s3://medical-imaging-output/
job_output/12345678901234567890123456789012-
DicomImport-09876543210987654321098765432109/"
  }
}

```

```
}
```

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Abrufen der Eigenschaften von Importaufträgen](#).

- APIEinzelheiten finden Sie unter [GetDICOMImport Job](#) in der AWS CLI Befehlsreferenz.

get-image-frame

Das folgende Codebeispiel zeigt, wie man `get-image-frame`.

AWS CLI

Um ein Bild zu erhalten, setzen Sie Pixeldaten ein

Das folgende `get-image-frame` Codebeispiel ruft einen Bildrahmen ab.

```
aws medical-imaging get-image-frame \
  --datastore-id "12345678901234567890123456789012" \
  --image-set-id "98765412345612345678907890789012" \
  --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \
  imageframe.jpg
```

Hinweis: Dieses Codebeispiel beinhaltet keine Ausgabe, da die `GetImageFrame` Aktion einen Stream von Pixeldaten an die Datei `imageframe.jpg` zurückgibt. Informationen zum Dekodieren und Anzeigen von Bildrahmen finden Sie unter Bibliotheken `dekodieren`. HTJ2K

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Abrufen von Pixeldaten von Bilddatensätzen](#).

- APIEinzelheiten finden Sie [GetImageFrame](#) in der AWS CLI Befehlsreferenz.

get-image-set-metadata

Das folgende Codebeispiel zeigt die Verwendung `get-image-set-metadata`.

AWS CLI

Beispiel 1: Um Metadaten eines Bildsatzes ohne Version abzurufen

Im folgenden `get-image-set-metadata` Codebeispiel werden Metadaten für einen Bildsatz abgerufen, ohne eine Version anzugeben.

Hinweis: `outfile` ist ein erforderlicher Parameter

```
aws medical-imaging get-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  studymetadata.json.gz
```

Die zurückgegebenen Metadaten werden mit gzip komprimiert und in der Datei `studymetadata.json.gz` gespeichert. Um den Inhalt des zurückgegebenen Objekts anzuzeigen, müssen Sie es zuerst dekomprimieren. JSON

Ausgabe:

```
{  
  "contentType": "application/json",  
  "contentEncoding": "gzip"  
}
```

Beispiel 2: Um Metadaten des Bildsatzes mit Version abzurufen

Im folgenden `get-image-set-metadata` Codebeispiel werden Metadaten für einen Bildsatz mit einer angegebenen Version abgerufen.

Hinweis: `outfile` ist ein erforderlicher Parameter

```
aws medical-imaging get-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --version-id 1 \  
  studymetadata.json.gz
```

Die zurückgegebenen Metadaten werden mit gzip komprimiert und in der Datei `studymetadata.json.gz` gespeichert. Um den Inhalt des zurückgegebenen Objekts anzuzeigen, müssen Sie es zuerst dekomprimieren. JSON

Ausgabe:

```
{  
  "contentType": "application/json",  
  "contentEncoding": "gzip"  
}
```

```
}
```

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Abrufen von Bildsatz-Metadaten](#).

- API-Einheiten finden Sie [GetImageSetMetadata](#) in der AWS CLI Befehlsreferenz.

get-image-set

Das folgende Codebeispiel zeigt die Verwendung `get-image-set`.

AWS CLI

Um die Eigenschaften von Bilddatensätzen abzurufen

Das folgende `get-image-set` Codebeispiel ruft die Eigenschaften für einen Bildsatz ab.

```
aws medical-imaging get-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id 18f88ac7870584f58d56256646b4d92b \  
  --version-id 1
```

Ausgabe:

```
{  
  "versionId": "1",  
  "imageSetWorkflowStatus": "COPIED",  
  "updatedAt": 1680027253.471,  
  "imageSetId": "18f88ac7870584f58d56256646b4d92b",  
  "imageSetState": "ACTIVE",  
  "createdAt": 1679592510.753,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Abrufen von Bilddatensatz-Eigenschaften](#).

- API-Einheiten finden Sie [GetImageSet](#) in der AWS CLI Befehlsreferenz.

list-datastores

Das folgende Codebeispiel zeigt die Verwendung `list-datastores`.

AWS CLI

Um Datenspeicher aufzulisten

Das folgende `list-datastores` Codebeispiel listet die verfügbaren Datenspeicher auf.

```
aws medical-imaging list-datastores
```

Ausgabe:

```
{
  "datastoreSummaries": [
    {
      "datastoreId": "12345678901234567890123456789012",
      "datastoreName": "TestDatastore123",
      "datastoreStatus": "ACTIVE",
      "datastoreArn": "arn:aws:medical-imaging:us-east-1:123456789012:datastore/12345678901234567890123456789012",
      "createdAt": "2022-11-15T23:33:09.643000+00:00",
      "updatedAt": "2022-11-15T23:33:09.643000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS HealthImaging Developer Guide unter [Auflisten von Datenspeichern](#).

- API-Einheiten finden Sie [ListDatastores](#) in der AWS CLI Befehlsreferenz.

`list-dicom-import-jobs`

Das folgende Codebeispiel zeigt die Verwendung `list-dicom-import-jobs`.

AWS CLI

Um DICOM-Importaufträge aufzulisten

Das folgende `list-dicom-import-jobs` Codebeispiel listet DICOM-Importaufträge auf.

```
aws medical-imaging list-dicom-import-jobs \
  --datastore-id "12345678901234567890123456789012"
```

Ausgabe:

```
{
  "jobSummaries": [
    {
      "jobId": "09876543210987654321098765432109",
      "jobName": "my-job",
      "jobStatus": "COMPLETED",
      "datastoreId": "12345678901234567890123456789012",
      "dataAccessRoleArn": "arn:aws:iam::123456789012:role/ImportJobDataAccessRole",
      "endedAt": "2022-08-12T11:21:56.504000+00:00",
      "submittedAt": "2022-08-12T11:20:21.734000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS HealthImaging Developer Guide unter [Auflisten von Importaufträgen](#).

- API-Einheiten finden Sie unter [istDICOMImportL-Jobs](#) in der AWS CLI Befehlsreferenz.

list-image-set-versions

Das folgende Codebeispiel zeigt, wie man `list-image-set-versions`.

AWS CLI

Um Versionen von Bildsätzen aufzulisten

Das folgende `list-image-set-versions` Codebeispiel listet den Versionsverlauf für einen Bildsatz auf.

```
aws medical-imaging list-image-set-versions \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Ausgabe:

```
{
  "imageSetPropertiesList": [
    {
```

```

    "ImageSetWorkflowStatus": "UPDATED",
    "versionId": "4",
    "updatedAt": 1680029436.304,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "ACTIVE",
    "createdAt": 1680027126.436
  },
  {
    "ImageSetWorkflowStatus": "UPDATED",
    "versionId": "3",
    "updatedAt": 1680029163.325,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "ACTIVE",
    "createdAt": 1680027126.436
  },
  {
    "ImageSetWorkflowStatus": "COPY_FAILED",
    "versionId": "2",
    "updatedAt": 1680027455.944,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "ACTIVE",
    "message": "INVALID_REQUEST: Series of SourceImageSet and
DestinationImageSet don't match.",
    "createdAt": 1680027126.436
  },
  {
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "ACTIVE",
    "versionId": "1",
    "ImageSetWorkflowStatus": "COPIED",
    "createdAt": 1680027126.436
  }
]
}

```

Weitere Informationen finden Sie im AWS HealthImaging Developer Guide unter [Auflisten von Imageset-Versionen](#).

- API-Einheiten finden Sie [ListImageSetVersions](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Beispiel 1: Um Ressourcen-Tags für einen Datenspeicher aufzulisten

Das folgende `list-tags-for-resource` Codebeispiel listet Tags für einen Datenspeicher auf.

```
aws medical-imaging list-tags-for-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012"
```

Ausgabe:

```
{  
  "tags":{  
    "Deployment":"Development"  
  }  
}
```

Beispiel 2: Um Ressourcen-Tags für einen Bildsatz aufzulisten

Das folgende `list-tags-for-resource` Codebeispiel listet Tags für einen Bildsatz auf.

```
aws medical-imaging list-tags-for-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b"
```

Ausgabe:

```
{  
  "tags":{  
    "Deployment":"Development"  
  }  
}
```

Weitere Informationen finden Sie unter [Ressourcen taggen mit AWS HealthImaging](#) im AWS HealthImaging Entwicklerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

search-image-sets

Das folgende Codebeispiel zeigt die Verwendung `search-image-sets`.

AWS CLI

Beispiel 1: Um Bilddatensätze mit einem EQUAL Operator zu durchsuchen

Im folgenden `search-image-sets` Codebeispiel wird der EQUAL Operator verwendet, um Bilddatensätze auf der Grundlage eines bestimmten Werts zu durchsuchen.

```
aws medical-imaging search-image-sets \  
  --datastore-id 12345678901234567890123456789012 \  
  --search-criteria file://search-criteria.json
```

Inhalt von `search-criteria.json`

```
{  
  "filters": [{  
    "values": [{"DICOMPatientId" : "SUBJECT08701"}],  
    "operator": "EQUAL"  
  }]  
}
```

Ausgabe:

```
{  
  "imageSetsMetadataSummaries": [{  
    "imageSetId": "09876543210987654321098765432109",  
    "createdAt": "2022-12-06T21:40:59.429000+00:00",  
    "version": 1,  
    "DICOMTags": {  
      "DICOMStudyId": "2011201407",  
      "DICOMStudyDate": "19991122",  
      "DICOMPatientSex": "F",  
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",  
      "DICOMPatientBirthDate": "19201120",  
      "DICOMStudyDescription": "UNKNOWN",  
      "DICOMPatientId": "SUBJECT08701",  
      "DICOMPatientName": "Melissa844 Huel628",  
      "DICOMNumberOfStudyRelatedInstances": 1,  
      "DICOMStudyTime": "140728",
```

```

        "DICOMNumberOfStudyRelatedSeries": 1
      },
      "updatedAt": "2022-12-06T21:40:59.429000+00:00"
    ]
  }

```

Beispiel 2: Um Bilddatensätze mit einem BETWEEN Operator zu durchsuchen, der DICOMStudyDate und verwendet DICOMStudyTime

Im folgenden search-image-sets Codebeispiel wird nach Bilddatensätzen mit DICOM Studien gesucht, die zwischen dem 1. Januar 1990 (12:00 Uhr) und dem 1. Januar 2023 (12:00 Uhr) generiert wurden.

Hinweis: DICOMStudyTime ist optional. Wenn es nicht vorhanden ist, ist 12:00 Uhr (Beginn des Tages) der Zeitwert für die Datumsangaben, die für die Filterung bereitgestellt werden.

```

aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json

```

Inhalt von search-criteria.json

```

{
  "filters": [{
    "values": [{
      "DICOMStudyDateAndTime": {
        "DICOMStudyDate": "19900101",
        "DICOMStudyTime": "000000"
      }
    },
    {
      "DICOMStudyDateAndTime": {
        "DICOMStudyDate": "20230101",
        "DICOMStudyTime": "000000"
      }
    }
  ]],
  "operator": "BETWEEN"
}

```

Ausgabe:

```
{
  "imageSetsMetadataSummaries": [{
    "imageSetId": "09876543210987654321098765432109",
    "createdAt": "2022-12-06T21:40:59.429000+00:00",
    "version": 1,
    "DICOMTags": {
      "DICOMStudyId": "2011201407",
      "DICOMStudyDate": "19991122",
      "DICOMPatientSex": "F",
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
      "DICOMPatientBirthDate": "19201120",
      "DICOMStudyDescription": "UNKNOWN",
      "DICOMPatientId": "SUBJECT08701",
      "DICOMPatientName": "Melissa844 Huel628",
      "DICOMNumberOfStudyRelatedInstances": 1,
      "DICOMStudyTime": "140728",
      "DICOMNumberOfStudyRelatedSeries": 1
    },
    "updatedAt": "2022-12-06T21:40:59.429000+00:00"
  ]
}
```

Beispiel 3: Um Bilddatensätze mit einem BETWEEN Operator zu suchen, der createdAt (Zeitstudien wurden zuvor persistent gespeichert) verwendet

Im folgenden search-image-sets Codebeispiel wird nach Bilddatensätzen gesucht, bei denen DICOM Studien HealthImaging zwischen den Zeitbereichen in UTC der Zeitzone gespeichert wurden.

Hinweis: Geben Sie das Beispielformat createdAt an („1985-04-12T 23:20:50.52 Z“).

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json
```

Inhalt von search-criteria.json

```
{
  "filters": [{
    "values": [{
      "createdAt": "1985-04-12T23:20:50.52Z"
    }],
  }],
}
```

```

    {
      "createdAt": "2022-04-12T23:20:50.52Z"
    }],
    "operator": "BETWEEN"
  ]
}

```

Ausgabe:

```

{
  "imageSetsMetadataSummaries": [{
    "imageSetId": "09876543210987654321098765432109",
    "createdAt": "2022-12-06T21:40:59.429000+00:00",
    "version": 1,
    "DICOMTags": {
      "DICOMStudyId": "2011201407",
      "DICOMStudyDate": "19991122",
      "DICOMPatientSex": "F",
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
      "DICOMPatientBirthDate": "19201120",
      "DICOMStudyDescription": "UNKNOWN",
      "DICOMPatientId": "SUBJECT08701",
      "DICOMPatientName": "Melissa844 Huel628",
      "DICOMNumberOfStudyRelatedInstances": 1,
      "DICOMStudyTime": "140728",
      "DICOMNumberOfStudyRelatedSeries": 1
    },
    "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
  ]
}

```

Beispiel 4: Um Bilddatensätze mit einem EQUAL Operator an und ein zu durchsuchen DICOMSeriesInstanceUID und die Antwort in der Reihenfolge nach dem Feld BETWEEN zu sortieren updatedAt ASC updatedAt

Im folgenden `search-image-sets` Codebeispiel wird nach Bilddatensätzen gesucht, bei denen der EQUAL Operator aktiviert DICOMSeriesInstanceUID und BETWEEN aktiviert ist, updatedAt und die Antwort wird in der ASC Reihenfolge nach dem updatedAt Feld sortiert.

Hinweis: Geben Sie das Beispielformat updatedAt an („1985-04-12T 23:20:50.52 Z“).

```
aws medical-imaging search-image-sets \
```



```
--datastore-id 12345678901234567890123456789012 \  
--search-criteria file://search-criteria.json
```

Inhalt von search-criteria.json

```
{  
  "filters": [{  
    "values": [{  
      "updatedAt": "2024-03-11T15:00:05.074000-07:00"  
    }, {  
      "updatedAt": "2024-03-11T16:00:05.074000-07:00"  
    }],  
    "operator": "BETWEEN"  
  }, {  
    "values": [{  
      "DICOMSeriesInstanceUID": "1.2.840.99999999.84710745.943275268089"  
    }],  
    "operator": "EQUAL"  
  }],  
  "sort": {  
    "sortField": "updatedAt",  
    "sortOrder": "ASC"  
  }  
}
```

Ausgabe:

```
{  
  "imageSetsMetadataSummaries": [{  
    "imageSetId": "09876543210987654321098765432109",  
    "createdAt": "2022-12-06T21:40:59.429000+00:00",  
    "version": 1,  
    "DICOMTags": {  
      "DICOMStudyId": "2011201407",  
      "DICOMStudyDate": "19991122",  
      "DICOMPatientSex": "F",  
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",  
      "DICOMPatientBirthDate": "19201120",  
      "DICOMStudyDescription": "UNKNOWN",  
      "DICOMPatientId": "SUBJECT08701",  
      "DICOMPatientName": "Melissa844 Huel628",  
      "DICOMNumberOfStudyRelatedInstances": 1,  
      "DICOMStudyTime": "140728",  
    }  
  }  
}
```

```

        "DICOMNumberOfStudyRelatedSeries": 1
      },
      "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
    }]
  }

```

Weitere Informationen finden Sie unter [Suchen](#) von Bilddatensätzen im Entwicklerhandbuch.AWS HealthImaging

- APIEinzelheiten finden Sie [SearchImageSets](#) in der AWS CLI Befehlsreferenz.

start-dicom-import-job

Das folgende Codebeispiel zeigt die Verwendung start-dicom-import-job.

AWS CLI

Um einen DICOM-Importjob zu starten

Das folgende start-dicom-import-job Codebeispiel startet einen DICOM-Importauftrag.

```

aws medical-imaging start-dicom-import-job \
  --job-name "my-job" \
  --datastore-id "12345678901234567890123456789012" \
  --input-s3-uri "s3://medical-imaging-dicom-input/dicom_input/" \
  --output-s3-uri "s3://medical-imaging-output/job_output/" \
  --data-access-role-arn "arn:aws:iam::123456789012:role/ImportJobDataAccessRole"

```

Ausgabe:

```

{
  "datastoreId": "12345678901234567890123456789012",
  "jobId": "09876543210987654321098765432109",
  "jobStatus": "SUBMITTED",
  "submittedAt": "2022-08-12T11:28:11.152000+00:00"
}

```

Weitere Informationen finden Sie unter [Starten eines Importauftrags](#) im AWS HealthImaging Entwicklerhandbuch.

- APIEinzelheiten finden Sie unter [StartDICOMImport Job](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Beispiel 1: Um einen Datenspeicher zu taggen

Die folgenden `tag-resource` Codebeispiele kennzeichnen einen Datenspeicher.

```
aws medical-imaging tag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tags '{"Deployment":"Development"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einen Bilddatensatz zu taggen

In den folgenden `tag-resource` Codebeispielen wird ein Bilddatensatz markiert.

```
aws medical-imaging tag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tags '{"Deployment":"Development"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ressourcen taggen mit AWS HealthImaging](#) im AWS HealthImaging Entwicklerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Beispiel 1: Um die Markierung eines Datenspeichers aufzuheben

Im folgenden `untag-resource` Codebeispiel wird die Markierung eines Datenspeichers aufgehoben.

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tag-keys ["Deployment"]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die Markierung eines Bilddatensatzes aufzuheben

Im folgenden `untag-resource` Codebeispiel wird die Markierung eines Bilddatensatzes aufgehoben.

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tag-keys ["Deployment"]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Ressourcen AWS HealthImaging taggen mit](#).

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-image-set-metadata

Das folgende Codebeispiel zeigt die Verwendung `update-image-set-metadata`.

AWS CLI

Beispiel 1: Um ein Attribut in Bildsatz-Metadaten einzufügen oder zu aktualisieren

Im folgenden `update-image-set-metadata` Beispiel wird ein Attribut in Bildsatz-Metadaten eingefügt oder aktualisiert.

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --tag-key Deployment
```

```
--latest-version-id 1 \
--cli-binary-format raw-in-base64-out \
--update-image-set-metadata-updates file://metadata-updates.json
```

Inhalt von metadata-updates.json

```
{
  "DICOMUpdates": {
    "updatableAttributes": "{\"SchemaVersion\":1.1,\"Patient\":{\"DICOM\":{\"PatientName\": \"MX^MX\"}}}"
  }
}
```

Ausgabe:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}
```

Beispiel 2: Um ein Attribut aus den Metadaten eines Bildsatzes zu entfernen

Im folgenden update-image-set-metadata Beispiel wird ein Attribut aus den Metadaten eines Bildsatzes entfernt.

```
aws medical-imaging update-image-set-metadata \
--datastore-id 12345678901234567890123456789012 \
--image-set-id ea92b0d8838c72a3f25d00d13616f87e \
--latest-version-id 1 \
--cli-binary-format raw-in-base64-out \
--update-image-set-metadata-updates file://metadata-updates.json
```

Inhalt von metadata-updates.json

```
{
  "DICOMUpdates": {
```

```

    "removableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"DICOM\":
{\"StudyDescription\": \"CHEST\"}}}"
  }
}

```

Ausgabe:

```

{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}

```

Beispiel 3: Um eine Instanz aus den Metadaten eines Bildsatzes zu entfernen

Im folgenden `update-image-set-metadata` Beispiel wird eine Instanz aus den Metadaten des Bildsatzes entfernt.

```

aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --latest-version-id 1 \
  --cli-binary-format raw-in-base64-out \
  --update-image-set-metadata-updates file://metadata-updates.json

```

Inhalt von `metadata-updates.json`

```

{
  "DICOMUpdates": {
    "removableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances\":
{\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {}}}}}}}"
  }
}

```

Ausgabe:

```

{

```

```

    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
  }

```

Beispiel 4: Um einen Bildsatz auf eine frühere Version zurückzusetzen

Das folgende `update-image-set-metadata` Beispiel zeigt, wie ein Bildsatz auf eine frühere Version zurückgesetzt wird. `CopyImageSet` und `UpdateImageSetMetadata` Aktionen erstellen neue Versionen von Bilddatensätzen.

```

aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
  --latest-version-id 3 \
  --cli-binary-format raw-in-base64-out \
  --update-image-set-metadata-updates '{"revertToVersionId": "1"}'

```

Ausgabe:

```

{
  "datastoreId": "12345678901234567890123456789012",
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "latestVersionId": "4",
  "imageSetState": "LOCKED",
  "imageSetWorkflowStatus": "UPDATING",
  "createdAt": 1680027126.436,
  "updatedAt": 1680042257.908
}

```

Beispiel 5: Um einer Instanz ein privates DICOM Datenelement hinzuzufügen

Das folgende `update-image-set-metadata` Beispiel zeigt, wie ein privates Element zu einer angegebenen Instanz innerhalb eines Bilddatensatzes hinzugefügt wird. Der DICOM Standard erlaubt private Datenelemente für die Übertragung von Informationen, die nicht in Standarddatenelementen enthalten sein können. Mit der `UpdateImageSetMetadata` Aktion können Sie private Datenelemente erstellen, aktualisieren und löschen.

```
aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
  --latest-version-id 1 \
  --cli-binary-format raw-in-base64-out \
  --force \
  --update-image-set-metadata-updates file://metadata-updates.json
```

Inhalt von metadata-updates.json

```
{
  "DICOMUpdates": {
    "updatableAttributes": "{\"SchemaVersion\": 1.1, \"Study\": {\"Series
\\\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\\\": {\"Instances
\\\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\\\": {\"DICOM\":
{\"001910F9\\\": \"97\\\"}, \"DICOMVRs\\\": {\"001910F9\\\": \"DS\\\"}}}}}}}"
  }
}
```

Ausgabe:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}
```

Beispiel 6: Um ein privates DICOM Datenelement auf eine Instanz zu aktualisieren

Das folgende update-image-set-metadata Beispiel zeigt, wie der Wert eines privaten Datenelements aktualisiert wird, das zu einer Instanz innerhalb eines Bilddatensatzes gehört.

```
aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
  --latest-version-id 1 \
  --cli-binary-format raw-in-base64-out \
```



```
--force \
--update-image-set-metadata-updates file://metadata-updates.json
```

Inhalt von `metadata-updates.json`

```
{
  "DICOMUpdates": {
    "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\\\": {\"1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\\\": {\"Instances
\\\": {\"1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\\\": {\"DICOM\":
{\\\"00091001\\\": \\\"GE_GENESIS_DD\\\"}}}}}}}"
  }
}
```

Ausgabe:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}
```

Beispiel 7: Um a SOPInstanceUID mit dem Force-Parameter zu aktualisieren

Das folgende `update-image-set-metadata` Beispiel zeigt, wie eine aktualisiert wird SOPInstanceUID, indem der Force-Parameter verwendet wird, um die DICOM Metadatenbeschränkungen zu überschreiben.

```
aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
  --latest-version-id 1 \
  --cli-binary-format raw-in-base64-out \
  --force \
  --update-image-set-metadata-updates file://metadata-updates.json
```

Inhalt von `metadata-updates.json`

```
{
  "DICOMUpdates": {
    "updatableAttributes": "{\\"SchemaVersion\\":1.1,\\"Study\\":{\\"Series\\":
{\\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3656.0\\":{\\"Instances
\\":{\\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.0\\":{\\"DICOM\\":
{\\"SOPInstanceUID\\":
\\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.9\\"}}}}}}}"
  }
}
```

Ausgabe:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}
```

Weitere Informationen finden Sie im AWS HealthImaging Entwicklerhandbuch unter [Aktualisieren von Bilddatensatz-Metadaten](#).

- API-Einheiten finden Sie [UpdateImageSetMetadata](#) in der AWS CLI Befehlsreferenz.

HealthLake Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren HealthLake.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-fhir-datastore

Das folgende Codebeispiel zeigt die Verwendung `create-fhir-datastore`.

AWS CLI

Um einen FHIR Datenspeicher zu erstellen.

Das folgende `create-fhir-datastore` Beispiel zeigt, wie Sie einen neuen Datenspeicher in Amazon erstellen HealthLake.

```
aws healthlake create-fhir-datastore \  
  --region us-east-1 \  
  --datastore-type-version R4 \  
  --datastore-type-version R4 \  
  --datastore-name "FhirTestDatastore"
```

Ausgabe:

```
{  
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Datastore ID)/r4/",  
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Datastore ID)",  
  "DatastoreStatus": "CREATING",  
  "DatastoreId": "(Datastore ID)"  
}
```

Weitere Informationen finden Sie unter [Erstellen und Überwachen eines FHIR Datenspeichers](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [CreateFhirDatastore](#) in der AWS CLI Befehlsreferenz.

delete-fhir-datastore

Das folgende Codebeispiel zeigt die Verwendung `delete-fhir-datastore`.

AWS CLI

Um einen FHIR Datenspeicher zu löschen

Das folgende `delete-fhir-datastore` Beispiel zeigt, wie Sie einen Datenspeicher und seinen gesamten Inhalt in Amazon löschen HealthLake.

```
aws healthlake delete-fhir-datastore \  
  --datastore-id (Data Store ID) \  
  --region us-east-1
```

Ausgabe:

```
{  
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Datastore ID)/r4/",  
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Datastore ID)",  
  "DatastoreStatus": "DELETING",  
  "DatastoreId": "(Datastore ID)"  
}
```

Weitere Informationen finden Sie unter Erstellen und Überwachen eines FHIR Datenspeichers <<https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-FHIR-healthlake.html>> im Amazon Developer Guide. HealthLake

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteFhirDatastore](#) AWS CLI

`describe-fhir-datastore`

Das folgende Codebeispiel zeigt die Verwendung `describe-fhir-datastore`.

AWS CLI

Um einen FHIR Datenspeicher zu beschreiben

Das folgende `describe-fhir-datastore` Beispiel zeigt, wie Sie die Eigenschaften eines Datenspeichers in Amazon finden HealthLake.

```
aws healthlake describe-fhir-datastore \  
  --datastore-id "1f2f459836ac6c513ce899f9e4f66a59" \  
  --region us-east-1
```

```
--region us-east-1
```

Ausgabe:

```
{
  "DatastoreProperties": {
    "PreloadDataConfig": {
      "PreloadDataType": "SYNTHEA"
    },
    "DatastoreName": "FhirTestDatastore",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/
(Datastore ID)",
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
(Datastore ID)/r4/",
    "DatastoreStatus": "CREATING",
    "DatastoreTypeVersion": "R4",
    "DatastoreId": "(Datastore ID)"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen und Überwachen von FHIR Datenspeichern](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [DescribeFhirDatastore](#) in der AWS CLI Befehlsreferenz.

describe-fhir-export-job

Das folgende Codebeispiel zeigt die Verwendung `describe-fhir-export-job`.

AWS CLI

Um einen FHIR Exportjob zu beschreiben

Das folgende `describe-fhir-export-job` Beispiel zeigt, wie Sie die Eigenschaften eines FHIR Exportauftrags in Amazon finden HealthLake.

```
aws healthlake describe-fhir-export-job \
  --datastore-id (Datastore ID) \
  --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

Ausgabe:

```
{
  "ExportJobProperties": {
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "IN_PROGRESS",
    "JobId": "9009813e9d69ba7cf79bcb3468780f16",
    "SubmitTime": 1609175692.715,
    "OutputDataConfig": {
      "S3Uri": "s3://(Bucket Name)/(Prefix
Name)/59593b2d0367ce252b5e66bf5fd6b574-
FHIR_EXPORT-9009813e9d69ba7cf79bcb3468780f16/"
    },
    "DatastoreId": "(Datastore ID)"
  }
}
```

Weitere Informationen finden Sie unter [Exportieren von Dateien aus einem FHIR Datenspeicher](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [DescribeFhirExportJob](#) in der AWS CLI Befehlsreferenz.

describe-fhir-import-job

Das folgende Codebeispiel zeigt die Verwendung `describe-fhir-import-job`.

AWS CLI

Um einen FHIR Importjob zu beschreiben

Das folgende `describe-fhir-import-job` Beispiel zeigt, wie Sie die Eigenschaften eines FHIR Importjobs mithilfe von Amazon lernen HealthLake.

```
aws healthlake describe-fhir-import-job \
  --datastore-id (Datastore ID) \
  --job-id c145fbb27b192af392f8ce6e7838e34f \
  --region us-east-1
```

Ausgabe:

```
{
  "ImportJobProperties": {
    "InputDataConfig": {
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
```

```

    { "arrayitem2": 2 }
  },
  "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
  "JobStatus": "COMPLETED",
  "JobId": "c145fbb27b192af392f8ce6e7838e34f",
  "SubmitTime": 1606272542.161,
  "EndTime": 1606272609.497,
  "DatastoreId": "(Datastore ID)"
}
}

```

Weitere Informationen finden Sie unter [Dateien in einen FHIR Datenspeicher importieren](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [DescribeFhirImportJob](#) in der AWS CLI Befehlsreferenz.

list-fhir-datastores

Das folgende Codebeispiel zeigt die Verwendung `list-fhir-datastores`.

AWS CLI

Um FHIR Datenspeicher aufzulisten

Das folgende `list-fhir-datastores` Beispiel zeigt, wie der Befehl verwendet wird und wie Benutzer Ergebnisse basierend auf dem Data Store-Status in Amazon filtern können HealthLake.

```

aws healthlake list-fhir-datastores \
  --region us-east-1 \
  --filter DatastoreId=ACTIVE

```

Ausgabe:

```

{
  "DatastoreIdList": [
    {
      "PreloadDataConfig": {
        "PreloadDataType": "SYNTHEA"
      },
      "DatastoreId": "FhirTestDatastoreId",
      "DatastoreIdArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastoreId/
<DatastoreId>",
    }
  ]
}

```

```

    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Datastore ID>/r4/",
    "DatastoreStatus": "ACTIVE",
    "DatastoreTypeVersion": "R4",
    "CreatedAt": 1605574003.209,
    "DatastoreId": "<Datastore ID>"
  },
  {
    "DatastoreName": "Demo",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Datastore ID>",
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Datastore ID>/r4/",
    "DatastoreStatus": "ACTIVE",
    "DatastoreTypeVersion": "R4",
    "CreatedAt": 1603761064.881,
    "DatastoreId": "<Datastore ID>"
  }
]
}

```

Weitere Informationen finden Sie unter [Erstellen und Überwachen eines FHIR Datenspeichers](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [ListFhirDatastores](#) in der AWS CLI Befehlsreferenz.

list-fhir-export-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-fhir-export-jobs`.

AWS CLI

Um alle FHIR Exportaufträge aufzulisten

Das folgende `list-fhir-export-jobs` Beispiel zeigt, wie Sie den Befehl verwenden, um eine Liste von Exportaufträgen anzuzeigen, die einem Konto zugeordnet sind.

```

aws healthlake list-fhir-export-jobs \
  --datastore-id (Datastore ID) \
  --submitted-before (DATE Like 2024-10-13T19:00:00Z) \
  --submitted-after (DATE Like 2020-10-13T19:00:00Z) \
  --job-name "FHIR-EXPORT" \
  --job-status SUBMITTED \

```


--max-results (*Integer between 1 and 500*)

Ausgabe:

```
{
  "ExportJobProperties": {
    "OutputDataConfig": {
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
      "S3Configuration": {
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
        "KmsKeyId" : "(KmsKey Id)"
      },
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "COMPLETED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",
    "JobName": "FHIR-EXPORT",
    "SubmitTime": 1606272542.161,
    "EndTime": 1606272609.497,
    "DatastoreId": "(Datastore ID)"
  }
}
"NextToken": String
```

Weitere Informationen finden Sie unter [Exportieren von Dateien aus einem FHIR Datenspeicher](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [ListFhirExportJobs](#) in der AWS CLI Befehlsreferenz.

list-fhir-import-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-fhir-import-jobs`.

AWS CLI

Um alle FHIR Importaufträge aufzulisten

Das folgende `list-fhir-import-jobs` Beispiel zeigt, wie Sie den Befehl verwenden, um eine Liste aller mit einem Konto verknüpften Importaufträge anzuzeigen.

```
aws healthlake list-fhir-import-jobs \
  --datastore-id (Datastore ID) \
```

```

--submitted-before (DATE Like 2024-10-13T19:00:00Z) \
--submitted-after (DATE Like 2020-10-13T19:00:00Z ) \
--job-name "FHIR-IMPORT" \
--job-status SUBMITTED \
-max-results (Integer between 1 and 500)

```

Ausgabe:

```

{
  "ImportJobProperties": {
    "OutputDataConfig": {
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
      "S3Configuration": {
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
        "KmsKeyId" : "(KmsKey Id)"
      },
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "COMPLETED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",
    "JobName": "FHIR-IMPORT",
    "SubmitTime": 1606272542.161,
    "EndTime": 1606272609.497,
    "DatastoreId": "(Datastore ID)"
  }
}
"NextToken": String

```

Weitere Informationen finden Sie unter [Dateien in den FHIR Datenspeicher importieren](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [ListFhirImportJobs](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für einen Datenspeicher aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags auf, die dem angegebenen Datenspeicher zugeordnet sind. :

```
aws healthlake list-tags-for-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/  
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \  
  --region us-east-1
```

Ausgabe:

```
{  
  "tags": {  
    "key": "value",  
    "key1": "value1"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging resources in Amazon HealthLake im Amazon HealthLake Developer Guide](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-fhir-export-job

Das folgende Codebeispiel zeigt die Verwendung `start-fhir-export-job`.

AWS CLI

Um einen FHIR Exportjob zu starten

Das folgende `start-fhir-export-job` Beispiel zeigt, wie Sie einen FHIR Exportjob mit Amazon HealthLake starten.

```
aws healthlake start-fhir-export-job \  
  --output-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \  
  --datastore-id (Datastore ID) \  
  --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

Ausgabe:

```
{  
  "DatastoreId": "(Datastore ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "9b9a51943afaedd0a8c0c26c49135a31"
```

```
}
```

Weitere Informationen finden Sie unter [Exportieren von Dateien aus einem FHIR Datenspeicher](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [StartFhirExportJob](#) in der AWS CLI Befehlsreferenz.

start-fhir-import-job

Das folgende Codebeispiel zeigt die Verwendung `start-fhir-import-job`.

AWS CLI

Um einen FHIR Importjob zu starten

Das folgende `start-fhir-import-job` Beispiel zeigt, wie Sie einen FHIR Importjob mit Amazon HealthLake starten.

```
aws healthlake start-fhir-import-job \
  --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \
  --datastore-id (Datastore ID) \
  --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)" \
  --region us-east-1
```

Ausgabe:

```
{
  "DatastoreId": "(Datastore ID)",
  "JobStatus": "SUBMITTED",
  "JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

Weitere Informationen finden Sie unter Dateien in einen FHIR Datenspeicher importieren 'https://docs.aws.amazon.com/healthlake/latest/devguide/import-datastore.html' im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [StartFhirImportJob](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Tag zum Datenspeicher hinzuzufügen

Das folgende `tag-resource` Beispiel zeigt, wie ein Tag zu einem Datenspeicher hinzugefügt wird.

```
aws healthlake tag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:691207106566:datastore/  
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \  
  --tags '[{"Key": "key1", "Value": "value1"}]' \  
  --region us-east-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter 'Hinzufügen eines Tags zu einem Datenspeicher < <https://docs.aws.amazon.com/healthlake/latest/devguide/add-a-tag.html>>'__ im Amazon Developer Guide. HealthLake .

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einem Datenspeicher zu entfernen.

Das folgende `untag-resource` Beispiel zeigt, wie Tags aus einem Datenspeicher entfernt werden.

```
aws healthlake untag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/fhir/  
b91723d65c6fdeb1d26543a49d2ed1fa" \  
  --tag-keys ["key1"] \  
  --region us-east-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Entfernen von Tags aus einem Datenspeicher](#) im Amazon HealthLake Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

HealthOmics Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren HealthOmics.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

abort-multipart-read-set-upload

Das folgende Codebeispiel zeigt die Verwendung `abort-multipart-read-set-upload`.

AWS CLI

Um den Upload eines mehrteiligen Lesesatzes zu beenden

Im folgenden `abort-multipart-read-set-upload` Beispiel wird der Upload eines mehrteiligen Lesesatzes in Ihren HealthOmics Sequenzspeicher gestoppt.

```
aws omics abort-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Direktes Hochladen in einen Sequenzspeicher](#).

- API-Einheiten finden Sie [AbortMultipartReadSetUpload](#) in der AWS CLI Befehlsreferenz.

accept-share

Das folgende Codebeispiel zeigt die Verwendung `accept-share`.

AWS CLI

Um einen Teil der Analytics-Daten zu akzeptieren, speichern Sie Daten

Im folgenden `accept-share` Beispiel wird ein Teil der HealthOmics Analytics-Store-Daten akzeptiert.

```
aws omics accept-share \  
  ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Ausgabe:

```
{  
  "status": "ACTIVATING"  
}
```

Weitere Informationen finden Sie unter [Kontoübergreifendes Teilen](#) im AWS HealthOmics Benutzerhandbuch.

- API-Einheiten finden Sie [AcceptShare](#) in der AWS CLI Befehlsreferenz.

batch-delete-read-set

Das folgende Codebeispiel zeigt die Verwendung `batch-delete-read-set`.

AWS CLI

Um mehrere Lesesätze zu löschen

Im folgenden `batch-delete-read-set` Beispiel werden zwei Lesesätze gelöscht.

```
aws omics batch-delete-read-set \  
  --sequence-store-id 1234567890 \  
  --ids 1234567890 0123456789
```

Wenn beim Löschen eines der angegebenen Lesesätze ein Fehler auftritt, gibt der Dienst eine Fehlerliste zurück.

```
{
  "errors": [
    {
      "code": "",
      "id": "0123456789",
      "message": "The specified readset does not exist."
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [BatchDeleteReadSet](#) in der AWS CLI Befehlsreferenz.

cancel-annotation-import-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-annotation-import-job`.

AWS CLI

Um einen Importauftrag für Anmerkungen abubrechen

Im folgenden `cancel-annotation-import-job` Beispiel wird ein Importauftrag für Anmerkungen mit ID `04f57618-xmpl-4fd0-9349-e5a85aefb997` abgebrochen.

```
aws omics cancel-annotation-import-job \
  --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CancelAnnotationImportJob](#) in der AWS CLI Befehlsreferenz.

cancel-run

Das folgende Codebeispiel zeigt die Verwendung `cancel-run`.

AWS CLI

Um einen Lauf abubrechen

Im folgenden `cancel-run` Beispiel wird ein Lauf mit ID `1234567` abgebrochen.


```
aws omics cancel-run \  
  --id 1234567
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CancelRun](#) in der AWS CLI Befehlsreferenz.

cancel-variant-import-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-variant-import-job`.

AWS CLI

Um einen Variantenimportjob abzubrechen

Im folgenden `cancel-variant-import-job` Beispiel wird ein Variantenimportjob mit ID `69cb65d6-xmpl-4a4a-9025-4565794b684e` storniert.

```
aws omics cancel-variant-import-job \  
  --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CancelVariantImportJob](#) in der AWS CLI Befehlsreferenz.

complete-multipart-read-set-upload

Das folgende Codebeispiel zeigt die Verwendung `complete-multipart-read-set-upload`.

AWS CLI

Um einen mehrteiligen Upload abzuschließen, nachdem Sie alle Komponenten hochgeladen haben.

Das folgende `complete-multipart-read-set-upload` Beispiel schließt einen mehrteiligen Upload in einen Sequenzspeicher ab, sobald alle Komponenten hochgeladen wurden.

```
aws omics complete-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  \
```

```
--parts '[{"checksum": "gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/Vobh5zBDb4W4=", "partNumber": 1, "partSource": "SOURCE1"}]'
```

Ausgabe:

```
{
  "readSetId": "0000000001"
  "readSetId": "0000000002"
  "readSetId": "0000000003"
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Direkter Upload in einen Sequenzspeicher](#).

- API-Einheiten finden Sie [CompleteMultipartReadSetUpload](#) in der AWS CLI Befehlsreferenz.

create-annotation-store-version

Das folgende Codebeispiel zeigt die Verwendung `create-annotation-store-version`.

AWS CLI

Um eine neue Version eines Annotationsspeichers zu erstellen

Im folgenden `create-annotation-store-version` Beispiel wird eine neue Version eines Annotationsspeichers erstellt.

```
aws omics create-annotation-store-version \
  --name my_annotation_store \
  --version-name my_version
```

Ausgabe:

```
{
  "creationTime": "2023-07-21T17:15:49.251040+00:00",
  "id": "3b93cdef69d2",
  "name": "my_annotation_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/6505293348/reference/5987565360"
  },
}
```

```
"status": "CREATING",
"versionName": "my_version"
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Neue Versionen von Annotationsspeichern erstellen](#).

- API-Einheiten finden Sie [CreateAnnotationStoreVersion](#) in der AWS CLI Befehlsreferenz.

create-annotation-store

Das folgende Codebeispiel zeigt die Verwendung `create-annotation-store`.

AWS CLI

Beispiel 1: Um einen VCF Annotationsspeicher zu erstellen

Im folgenden `create-annotation-store` Beispiel wird ein Speicher für Anmerkungen im VCF Format erstellt.

```
aws omics create-annotation-store \
  --name my_ann_store \
  --store-format VCF \
  --reference referenceArn=arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

Ausgabe:

```
{
  "creationTime": "2022-11-23T22:48:39.226492Z",
  "id": "0a91xmplc71f",
  "name": "my_ann_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
  "status": "CREATING",
  "storeFormat": "VCF"
}
```

Beispiel 2: Um einen TSV Annotationsspeicher zu erstellen

Im folgenden `create-annotation-store` Beispiel wird ein Speicher für Anmerkungen im TSV Format erstellt.

```
aws omics create-annotation-store \  
  --name tsv_ann_store \  
  --store-format TSV \  
  --reference referenceArn=arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890 \  
  --store-options file://tsv-store-options.json
```

`tsv-store-options.json` konfiguriert die Formatoptionen für Anmerkungen.

```
{  
  "tsvStoreOptions": {  
    "annotationType": "CHR_START_END_ZERO_BASE",  
    "formatToHeader": {  
      "CHR": "chromosome",  
      "START": "start",  
      "END": "end"  
    },  
    "schema": [  
      {  
        "chromosome": "STRING"  
      },  
      {  
        "start": "LONG"  
      },  
      {  
        "end": "LONG"  
      },  
      {  
        "name": "STRING"  
      }  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "creationTime": "2022-11-30T01:28:08.525586Z",  
  "id": "861cxmpl96b0",  
}
```

```

    "name": "tsv_ann_store",
    "reference": {
      "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "CREATING",
    "storeFormat": "TSV",
    "storeOptions": {
      "tsvStoreOptions": {
        "annotationType": "CHR_START_END_ZERO_BASE",
        "formatToHeader": {
          "CHR": "chromosome",
          "END": "end",
          "START": "start"
        },
        "schema": [
          {
            "chromosome": "STRING"
          },
          {
            "start": "LONG"
          },
          {
            "end": "LONG"
          },
          {
            "name": "STRING"
          }
        ]
      }
    }
  }
}

```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CreateAnnotationStore](#) in der AWS CLI Befehlsreferenz.

create-multipart-read-set-upload

Das folgende Codebeispiel zeigt die Verwendung `create-multipart-read-set-upload`.

AWS CLI

Um mit dem Hochladen eines mehrteiligen Lesesatzes zu beginnen.

Im folgenden `create-multipart-read-set-upload` Beispiel wird ein Upload eines mehrteiligen Lesesatzes initiiert.

```
aws omics create-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --name HG00146 \  
  --source-file-type FASTQ \  
  --subject-id mySubject\  
  --sample-id mySample\  
  --description "FASTQ for HG00146"\  
  --generated-from "1000 Genomes"
```

Ausgabe:

```
{  
  "creationTime": "2022-07-13T23:25:20Z",  
  "description": "FASTQ for HG00146",  
  "generatedFrom": "1000 Genomes",  
  "name": "HG00146",  
  "sampleId": "mySample",  
  "sequenceStoreId": "0123456789",  
  "sourceFileType": "FASTQ",  
  "subjectId": "mySubject",  
  "uploadId": "1122334455"  
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Direkter Upload in einen Sequenzspeicher](#).

- API-Einheiten finden Sie [CreateMultipartReadSetUpload](#) in der AWS CLI Befehlsreferenz.

create-reference-store

Das folgende Codebeispiel zeigt die Verwendung `create-reference-store`.

AWS CLI

Um einen Referenzspeicher zu erstellen

Im folgenden `create-reference-store` Beispiel wird ein Referenzspeicher erstellt `my-ref-store`.

```
aws omics create-reference-store \  
  --name my-ref-store
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",  
  "creationTime": "2022-11-22T22:13:25.947Z",  
  "id": "1234567890",  
  "name": "my-ref-store"  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CreateReferenceStore](#) in der AWS CLI Befehlsreferenz.

create-run-group

Das folgende Codebeispiel zeigt die Verwendung `create-run-group`.

AWS CLI

Um eine Ausführungsgruppe zu erstellen

Im folgenden `create-run-group` Beispiel wird eine Ausführungsgruppe mit dem Namen `cram-converter` erstellt.

```
aws omics create-run-group \  
  --name cram-converter \  
  --max-cpus 20 \  
  --max-duration 600
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",  
  "id": "1234567",  
  "tags": {}  
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CreateRunGroup](#) in der AWS CLI Befehlsreferenz.

create-sequence-store

Das folgende Codebeispiel zeigt die Verwendung `create-sequence-store`.

AWS CLI

Um einen Sequenzspeicher zu erstellen

Im folgenden `create-sequence-store` Beispiel wird ein Sequenzspeicher erstellt.

```
aws omics create-sequence-store \  
  --name my-seq-store
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",  
  "creationTime": "2022-11-23T01:24:33.629Z",  
  "id": "1234567890",  
  "name": "my-seq-store"  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CreateSequenceStore](#) in der AWS CLI Befehlsreferenz.

create-share

Das folgende Codebeispiel zeigt die Verwendung `create-share`.

AWS CLI

Um einen Share eines HealthOmics Analytics-Stores zu erstellen

Das folgende `create-share` Beispiel zeigt, wie Sie einen Share eines HealthOmics Analytics-Stores erstellen, der von einem Abonnenten außerhalb des Kontos akzeptiert werden kann.

```
aws omics create-share \  
  --name my-share
```



```
--resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/  
omics_dev_var_store" \  
--principal-subscriber "123456789012" \  
--name "my_Share-123"
```

Ausgabe:

```
{  
  "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",  
  "name": "my_Share-123",  
  "status": "PENDING"  
}
```

Weitere Informationen finden Sie unter [Accountübergreifendes Teilen](#) im AWS HealthOmics Benutzerhandbuch.

- API-Einheiten finden Sie [CreateShare](#) in AWS CLI der Befehlsreferenz.

create-variant-store

Das folgende Codebeispiel zeigt die Verwendung `create-variant-store`.

AWS CLI

Um einen Variantenspeicher zu erstellen

Im folgenden `create-variant-store` Beispiel wird ein Variantenspeicher mit dem Namen `my_var_store` erstellt.

```
aws omics create-variant-store \  
  --name my_var_store \  
  --reference referenceArn=arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

Ausgabe:

```
{  
  "creationTime": "2022-11-23T22:09:07.534499Z",  
  "id": "02dexplcfd",  
  "name": "my_var_store",  
  "reference": {
```

```
    "referenceArn": "arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890"  
  },  
  "status": "CREATING"  
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CreateVariantStore](#) in der AWS CLI Befehlsreferenz.

create-workflow

Das folgende Codebeispiel zeigt die Verwendung `create-workflow`.

AWS CLI

Um einen Workflow zu erstellen

Im folgenden `create-workflow` Beispiel wird ein WDL Workflow erstellt.

```
aws omics create-workflow \  
  --name cram-converter \  
  --engine WDL \  
  --definition-zip fileb://workflow-crambam.zip \  
  --parameter-template file://workflow-params.json
```

`workflow-crambam.zip` ist ein ZIP Archiv, das eine Workflow-Definition enthält. `workflow-params.json` definiert Laufzeitparameter für den Workflow.

```
{  
  "ref_fasta" : {  
    "description": "Reference genome fasta file",  
    "optional": false  
  },  
  "ref_fasta_index" : {  
    "description": "Index of the reference genome fasta file",  
    "optional": false  
  },  
  "ref_dict" : {  
    "description": "dictionary file for 'ref_fasta'",  
    "optional": false  
  },  
}
```

```
"input_cram" : {
  "description": "The Cram file to convert to BAM",
  "optional": false
},
"sample_name" : {
  "description": "The name of the input sample, used to name the output BAM",
  "optional": false
}
}
```

Ausgabe:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
  "id": "1234567",
  "status": "CREATING",
  "tags": {}
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [CreateWorkflow](#) in der AWS CLI Befehlsreferenz.

delete-annotation-store-versions

Das folgende Codebeispiel zeigt die Verwendung `delete-annotation-store-versions`.

AWS CLI

Um eine Annotation Store-Version zu löschen

Im folgenden `delete-annotation-store-versions` Beispiel wird eine Version des Annotationsspeichers gelöscht.

```
aws omics delete-annotation-store-versions \
  --name my_annotation_store \
  --versions my_version
```

Ausgabe:

```
{
  "errors": []
}
```

```
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Neue Versionen von Annotationsspeichern erstellen](#).

- APIEinzelheiten finden Sie [DeleteAnnotationStoreVersions](#) in der AWS CLI Befehlsreferenz.

delete-annotation-store

Das folgende Codebeispiel zeigt die Verwendung `delete-annotation-store`.

AWS CLI

Um einen Annotationsspeicher zu löschen

Im folgenden `delete-annotation-store` Beispiel wird ein Annotationsspeicher mit dem Namen `my_vcf_store` gelöscht.

```
aws omics delete-annotation-store \  
  --name my_vcf_store
```

Ausgabe:

```
{  
  "status": "DELETING"  
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- APIEinzelheiten finden Sie [DeleteAnnotationStore](#) in der AWS CLI Befehlsreferenz.

delete-reference-store

Das folgende Codebeispiel zeigt die Verwendung `delete-reference-store`.

AWS CLI

Um einen Referenzspeicher zu löschen

Im folgenden `delete-reference-store` Beispiel wird ein Referenzspeicher mit ID `1234567890` gelöscht.

```
aws omics delete-reference-store \  
  --id 1234567890
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [DeleteReferenceStore](#) in der AWS CLI Befehlsreferenz.

delete-reference

Das folgende Codebeispiel zeigt die Verwendung `delete-reference`.

AWS CLI

Um eine Referenz zu löschen

Im folgenden `delete-reference` Beispiel wird eine Referenz gelöscht.

```
aws omics delete-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [DeleteReference](#) in der AWS CLI Befehlsreferenz.

delete-run-group

Das folgende Codebeispiel zeigt die Verwendung `delete-run-group`.

AWS CLI

Um eine Ausführungsgruppe zu löschen

Im folgenden `delete-run-group` Beispiel wird eine Ausführungsgruppe mit ID 1234567 gelöscht.

```
aws omics delete-run-group \  
  --id 1234567
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- APIEinzelheiten finden Sie [DeleteRunGroup](#) in der AWS CLI Befehlsreferenz.

delete-run

Das folgende Codebeispiel zeigt die Verwendung `delete-run`.

AWS CLI

Um einen Workflow zu löschen, führen Sie ihn aus

Das folgende `delete-run` Beispiel löscht einen Lauf mit ID. 1234567

```
aws omics delete-run \  
  --id 1234567
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- APIEinzelheiten finden Sie [DeleteRun](#) in der AWS CLI Befehlsreferenz.

delete-sequence-store

Das folgende Codebeispiel zeigt die Verwendung `delete-sequence-store`.

AWS CLI

Um einen Sequenzspeicher zu löschen

Das folgende `delete-sequence-store` Beispiel löscht einen Sequenzspeicher mit ID. 1234567890

```
aws omics delete-sequence-store \  
  --id 1234567890
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- APIEinzelheiten finden Sie [DeleteSequenceStore](#) in der AWS CLI Befehlsreferenz.

delete-share

Das folgende Codebeispiel zeigt die Verwendung `delete-share`.

AWS CLI

Um einen Teil der HealthOmics Analysedaten zu löschen

Im folgenden `delete-share` Beispiel wird ein kontoübergreifender Anteil an Analysedaten gelöscht.

```
aws omics delete-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Ausgabe:

```
{  
  "status": "DELETING"  
}
```

Weitere Informationen finden Sie unter [Accountübergreifendes Teilen](#) im AWS HealthOmics Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteShare](#) in der AWS CLI Befehlsreferenz.

`delete-variant-store`

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-variant-store`.

AWS CLI

Um einen Variantenspeicher zu löschen

Im folgenden `delete-variant-store` Beispiel wird ein Variantenspeicher mit dem Namen `my_var_store` gelöscht.

```
aws omics delete-variant-store \  
  --name my_var_store
```

Ausgabe:

```
{  
  "status": "DELETING"
```

```
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [DeleteVariantStore](#) in der AWS CLI Befehlsreferenz.

delete-workflow

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-workflow`.

AWS CLI

Um einen Workflow zu löschen

Das folgende `delete-workflow` Beispiel löscht einen Workflow mit ID. 1234567

```
aws omics delete-workflow \  
  --id 1234567
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [DeleteWorkflow](#) in der AWS CLI Befehlsreferenz.

get-annotation-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `get-annotation-import-job`.

AWS CLI

Um einen Importjob für Anmerkungen anzuzeigen

Im folgenden `get-annotation-import-job` Beispiel werden Details zu einem Importauftrag für Anmerkungen abgerufen.

```
aws omics get-annotation-import-job \  
  --job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

Ausgabe:

```
{  
  "creationTime": "2022-11-30T01:40:11.017746Z",
```



```

    "destinationName": "tsv_ann_store",
    "id": "984162c7-xmpl-4d23-ab47-286f7950bfbf",
    "items": [
      {
        "jobStatus": "COMPLETED",
        "source": "s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz"
      }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2022-11-30T01:42:39.134009Z"
  }
}

```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetAnnotationImportJob](#) in der AWS CLI Befehlsreferenz.

get-annotation-store-version

Das folgende Codebeispiel zeigt, wie man es benutzt `get-annotation-store-version`.

AWS CLI

Um die Metadaten für eine Annotation Store-Version abzurufen

Im folgenden `get-annotation-store-version` Beispiel werden die Metadaten für die angeforderte Version des Annotationsspeichers abgerufen.

```

aws omics get-annotation-store-version \
  --name my_annotation_store \
  --version-name my_version

```

Ausgabe:

```

{
  "storeId": "4934045d1c6d",
  "id": "2a3f4a44aa7b",
  "status": "ACTIVE",
  "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version",

```

```
"name": "my_annotation_store",
"versionName": "my_version",
"creationTime": "2023-07-21T17:15:49.251040+00:00",
"updateTime": "2023-07-21T17:15:56.434223+00:00",
"statusMessage": "",
"versionSizeBytes": 0
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Neue Versionen von Annotationsspeichern erstellen](#).

- API-Einheiten finden Sie [GetAnnotationStoreVersion](#) in der AWS CLI Befehlsreferenz.

get-annotation-store

Das folgende Codebeispiel zeigt, wie man es benutzt `get-annotation-store`.

AWS CLI

Um einen Annotationsspeicher anzuzeigen

Im folgenden `get-annotation-store` Beispiel werden Details zu einem Annotationsspeicher mit dem Namen `my_ann_store` abgerufen.

```
aws omics get-annotation-store \
  --name my_ann_store
```

Ausgabe:

```
{
  "creationTime": "2022-11-23T22:48:39.226492Z",
  "id": "0a91xmplc71f",
  "name": "my_ann_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
  "status": "CREATING",
  "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/my_ann_store",
  "storeFormat": "VCF",
  "storeSizeBytes": 0,
  "tags": {}
}
```

```
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetAnnotationStore](#) in der AWS CLI Befehlsreferenz.

get-read-set-activation-job

Das folgende Codebeispiel zeigt, wie man es benutzt `get-read-set-activation-job`.

AWS CLI

Um einen Readset-Aktivierungsjob anzuzeigen

Im folgenden `get-read-set-activation-job` Beispiel werden Details zu einem Read-Set-Aktivierungsjob abgerufen.

```
aws omics get-read-set-activation-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Ausgabe:

```
{  
  "completionTime": "2022-12-06T22:33:42.828Z",  
  "creationTime": "2022-12-06T22:32:45.213Z",  
  "id": "1234567890",  
  "sequenceStoreId": "1234567890",  
  "sources": [  
    {  
      "readSetId": "1234567890",  
      "status": "FINISHED",  
      "statusMessage": "No activation needed as read set is already in  
ACTIVATING or ACTIVE state."  
    }  
  ],  
  "status": "COMPLETED",  
  "statusMessage": "The job completed successfully."  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReadSetActivationJob](#) in der AWS CLI Befehlsreferenz.

get-read-set-export-job

Das folgende Codebeispiel zeigt, wie man es benutzt `get-read-set-export-job`.

AWS CLI

Um einen Exportauftrag für Lesesätze anzuzeigen

Im folgenden `get-read-set-export-job` Beispiel werden Details zu einem Exportauftrag für Lesesätze abgerufen.

```
aws omics get-read-set-export-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Ausgabe:

```
{  
  "completionTime": "2022-12-06T22:39:14.491Z",  
  "creationTime": "2022-12-06T22:37:18.612Z",  
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",  
  "id": "1234567890",  
  "sequenceStoreId": "1234567890",  
  "status": "COMPLETED",  
  "statusMessage": "The job is submitted and will start soon."  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReadSetExportJob](#) in der AWS CLI Befehlsreferenz.

get-read-set-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `get-read-set-import-job`.

AWS CLI

Um einen Importjob für Lesesätze anzuzeigen

Im folgenden `get-read-set-import-job` Beispiel werden Details zu einem Importauftrag für Lesesätze abgerufen.

```
aws omics get-read-set-import-job \
  --sequence-store-id 1234567890 \
  --id 1234567890
```

Ausgabe:

```
{
  "creationTime": "2022-11-23T01:36:38.158Z",
  "id": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "sequenceStoreId": "1234567890",
  "sources": [
    {
      "name": "HG00100",
      "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
      "sampleId": "bam-sample",
      "sourceFileType": "BAM",
      "sourceFiles": {
        "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam",
        "source2": ""
      },
      "status": "IN_PROGRESS",
      "statusMessage": "The source job is currently in progress.",
      "subjectId": "bam-subject",
      "tags": {
        "aws:omics:sampleId": "bam-sample",
        "aws:omics:subjectId": "bam-subject"
      }
    },
    {
      "name": "HG00146",
      "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
      "sampleId": "fastq-sample",
      "sourceFileType": "FASTQ",
      "sourceFiles": {
        "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_1.filt.fastq.gz",
        "source2": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_2.filt.fastq.gz"
      }
    }
  ]
}
```

```

    },
    "status": "IN_PROGRESS",
    "statusMessage": "The source job is currently in progress.",
    "subjectId": "fastq-subject",
    "tags": {
      "aws:omics:sampleId": "fastq-sample",
      "aws:omics:subjectId": "fastq-subject"
    }
  },
  {
    "name": "HG00096",
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890",
    "sampleId": "cram-sample",
    "sourceFileType": "CRAM",
    "sourceFiles": {
      "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram",
      "source2": ""
    },
    "status": "IN_PROGRESS",
    "statusMessage": "The source job is currently in progress.",
    "subjectId": "cram-subject",
    "tags": {
      "aws:omics:sampleId": "cram-sample",
      "aws:omics:subjectId": "cram-subject"
    }
  }
],
"status": "IN_PROGRESS",
"statusMessage": "The job is currently in progress."
}

```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReadSetImportJob](#) in der AWS CLI Befehlsreferenz.

get-read-set-metadata

Das folgende Codebeispiel zeigt, wie man es benutzt `get-read-set-metadata`.

AWS CLI

Um einen Lesesatz anzusehen

Im folgenden `get-read-set-metadata` Beispiel werden Details zu den Dateien eines Lesesatzes abgerufen.

```
aws omics get-read-set-metadata \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/  
readSet/1234567890",  
  "creationTime": "2022-11-23T21:55:00.515Z",  
  "fileType": "FASTQ",  
  "files": {  
    "source1": {  
      "contentLength": 310054739,  
      "partSize": 104857600,  
      "totalParts": 3  
    },  
    "source2": {  
      "contentLength": 307846621,  
      "partSize": 104857600,  
      "totalParts": 3  
    }  
  },  
  "id": "1234567890",  
  "name": "HG00146",  
  "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/  
reference/1234567890",  
  "sampleId": "fastq-sample",  
  "sequenceInformation": {  
    "alignment": "UNALIGNED",  
    "totalBaseCount": 677717384,  
    "totalReadCount": 8917334  
  },  
  "sequenceStoreId": "1234567890",  
  "status": "ACTIVE",  
  "subjectId": "fastq-subject"  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReadSetMetadata](#) in der AWS CLI Befehlsreferenz.

get-read-set

Das folgende Codebeispiel zeigt, wie man es benutzt `get-read-set`.

AWS CLI

Um ein Leseset herunterzuladen

Im folgenden `get-read-set` Beispiel wird Teil 3 eines Lesesatzes als heruntergeladen `1234567890.3.bam`.

```
aws omics get-read-set \  
  --sequence-store-id 1234567890 \  
  --id 1234567890 \  
  --part-number 3 1234567890.3.bam
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReadSet](#) in der AWS CLI Befehlsreferenz.

get-reference-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `get-reference-import-job`.

AWS CLI

Um einen Referenzimportjob anzuzeigen

Im folgenden `get-reference-import-job` Beispiel werden Details zu einem Referenzimportauftrag abgerufen.

```
aws omics get-reference-import-job \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

Ausgabe:

```
{  
  "creationTime": "2022-11-22T22:25:41.124Z",
```



```

    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
    "sources": [
      {
        "name": "assembly-38",
        "sourceFile": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
        "status": "IN_PROGRESS",
        "statusMessage": "The source job is currently in progress."
      }
    ],
    "status": "IN_PROGRESS",
    "statusMessage": "The job is currently in progress."
  }

```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReferenceImportJob](#) in der AWS CLI Befehlsreferenz.

get-reference-metadata

Das folgende Codebeispiel zeigt, wie man es benutzt `get-reference-metadata`.

AWS CLI

Um eine Referenz anzuzeigen

Im folgenden `get-reference-metadata` Beispiel werden Details zu einer Referenz abgerufen.

```

aws omics get-reference-metadata \
  --reference-store-id 1234567890 \
  --id 1234567890

```

Ausgabe:

```

{
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
  "creationTime": "2022-11-22T22:27:09.033Z",
  "files": {
    "index": {

```

```
        "contentLength": 160928,
        "partSize": 104857600,
        "totalParts": 1
    },
    "source": {
        "contentLength": 3249912778,
        "partSize": 104857600,
        "totalParts": 31
    }
},
"id": "1234567890",
"md5": "7ff134953dcca8c8997453bbb80b6b5e",
"name": "assembly-38",
"referenceStoreId": "1234567890",
"status": "ACTIVE",
"updateTime": "2022-11-22T22:27:09.033Z"
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReferenceMetadata](#) in der AWS CLI Befehlsreferenz.

get-reference-store

Das folgende Codebeispiel zeigt, wie man es benutzt `get-reference-store`.

AWS CLI

Um ein Referenzgeschäft aufzurufen

Im folgenden `get-reference-store` Beispiel werden Details zu einem Referenzspeicher abgerufen.

```
aws omics get-reference-store \
  --id 1234567890
```

Ausgabe:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
  "creationTime": "2022-09-23T23:27:20.364Z",
  "id": "1234567890",
  "name": "my-rstore-0"
```

```
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReferenceStore](#) in der AWS CLI Befehlsreferenz.

get-reference

Das folgende Codebeispiel zeigt, wie man es benutzt `get-reference`.

AWS CLI

Um eine Genomreferenz herunterzuladen

Im folgenden `get-reference` Beispiel wird Teil 1 eines Genoms als heruntergeladen `hg38.1.fa`.

```
aws omics get-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890 \  
  --part-number 1 hg38.1.fa
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetReference](#) in der AWS CLI Befehlsreferenz.

get-run-group

Das folgende Codebeispiel zeigt, wie man es benutzt `get-run-group`.

AWS CLI

Um eine Ausführungsgruppe anzuzeigen

Im folgenden `get-run-group` Beispiel werden Details zu einer Ausführungsgruppe abgerufen.

```
aws omics get-run-group \  
  --id 1234567
```

Ausgabe:

```
{
```

```
"arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
"creationTime": "2022-12-01T00:58:42.915219Z",
"id": "1234567",
"maxCpus": 20,
"maxDuration": 600,
"name": "cram-convert",
"tags": {}
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetRunGroup](#) in der AWS CLI Befehlsreferenz.

get-run-task

Das folgende Codebeispiel zeigt, wie man es benutzt `get-run-task`.

AWS CLI

Um eine Aufgabe anzusehen

Im folgenden `get-run-task` Beispiel werden Details zu einer Workflow-Aufgabe abgerufen.

```
aws omics get-run-task \
  --id 1234567 \
  --task-id 1234567
```

Ausgabe:

```
{
  "cpus": 1,
  "creationTime": "2022-11-30T23:13:00.718651Z",
  "logStream": "arn:aws:logs:us-west-2:123456789012:log-group:/aws/omics/WorkflowLog:log-stream:run/1234567/task/1234567",
  "memory": 15,
  "name": "CramToBamTask",
  "startTime": "2022-11-30T23:17:47.016Z",
  "status": "COMPLETED",
  "stopTime": "2022-11-30T23:18:21.503Z",
  "taskId": "1234567"
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetRunTask](#) in der AWS CLI Befehlsreferenz.

get-run

Das folgende Codebeispiel zeigt, wie man es benutzt `get-run`.

AWS CLI

Um eine Workflow-Ausführung anzuzeigen

Im folgenden `get-run` Beispiel werden Details zu einer Workflow-Ausführung abgerufen.

```
aws omics get-run \  
--id 1234567
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
  "creationTime": "2022-11-30T22:58:22.615865Z",  
  "digest":  
    "sha256:c54bxmpl1742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",  
  "id": "1234567",  
  "name": "cram-to-bam",  
  "outputUri": "s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/",  
  "parameters": {  
    "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.dict",  
    "ref_fasta_index": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta.fai",  
    "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta",  
    "sample_name": "NA12878",  
    "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram"  
  },  
  "resourceDigests": {  
    "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta.fai":  
"etag:f76371b113734a56cde236bc0372de0a",  
    "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.dict":  
"etag:3884c62eb0e53fa92459ed9bff133ae6",  
    "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta":  
"etag:e307d81c605fb91b7720a08f00276842-388",
```

```
    "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram":  
    "etag:a9f52976381286c6143b5cc681671ec6"  
  },  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "startedBy": "arn:aws:iam::123456789012:user/laptop-2020",  
  "status": "STARTING",  
  "tags": {},  
  "workflowId": "1234567",  
  "workflowType": "PRIVATE"  
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetRun](#) in der AWS CLI Befehlsreferenz.

get-sequence-store

Das folgende Codebeispiel zeigt, wie man es benutzt `get-sequence-store`.

AWS CLI

Um einen Sequenzspeicher anzuzeigen

Im folgenden `get-sequence-store` Beispiel werden Details zu einem Sequenzspeicher mit ID abgerufen `1234567890`.

```
aws omics get-sequence-store \  
  --id 1234567890
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",  
  "creationTime": "2022-11-23T19:55:48.376Z",  
  "id": "1234567890",  
  "name": "my-seq-store"  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetSequenceStore](#) in der AWS CLI Befehlsreferenz.

get-share

Das folgende Codebeispiel zeigt, wie man es benutzt `get-share`.

AWS CLI

Um die Metadaten zu einem Teil von HealthOmics Analysedaten abzurufen

Im folgenden `get-share` Beispiel werden die Metadaten für einen kontoübergreifenden Anteil an Analysedaten abgerufen.

```
aws omics get-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Ausgabe:

```
{  
  "share": {  
    "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",  
    "name": "my_Share-123",  
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/omics_dev_var_store",  
    "principalSubscriber": "123456789012",  
    "ownerId": "555555555555",  
    "status": "PENDING"  
  }  
}
```

Weitere Informationen finden Sie unter [Kontoübergreifendes Teilen](#) im AWS HealthOmics Benutzerhandbuch.

- API-Einheiten finden Sie [GetShare](#) in der AWS CLI Befehlsreferenz.

get-variant-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `get-variant-import-job`.

AWS CLI

Um einen Variantenimport-Job anzuzeigen

Im folgenden `get-variant-import-job` Beispiel werden Details zu einem Variantenimportjob abgerufen.

```
aws omics get-variant-import-job \  
--job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

Ausgabe:

```
{  
  "creationTime": "2022-11-23T22:42:50.037812Z",  
  "destinationName": "my_var_store",  
  "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",  
  "items": [  
    {  
      "jobStatus": "IN_PROGRESS",  
      "source": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.known_indels.vcf.gz"  
    }  
  ],  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "runLeftNormalization": false,  
  "status": "IN_PROGRESS",  
  "updateTime": "2022-11-23T22:43:05.898309Z"  
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetVariantImportJob](#) in der AWS CLI Befehlsreferenz.

get-variant-store

Das folgende Codebeispiel zeigt, wie man es benutzt `get-variant-store`.

AWS CLI

Um einen Variantenspeicher anzuzeigen

Im folgenden `get-variant-store` Beispiel werden Details zu einem Variantenspeicher abgerufen.

```
aws omics get-variant-store \  

```



```
--name my_var_store
```

Ausgabe:

```
{
  "creationTime": "2022-11-23T22:09:07.534499Z",
  "id": "02dexplcfdd",
  "name": "my_var_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
  "status": "CREATING",
  "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/my_var_store",
  "storeSizeBytes": 0,
  "tags": {},
  "updateTime": "2022-11-23T22:09:24.931711Z"
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetVariantStore](#) in der AWS CLI Befehlsreferenz.

get-workflow

Das folgende Codebeispiel zeigt, wie man es benutzt `get-workflow`.

AWS CLI

Um einen Workflow anzuzeigen

Im folgenden `get-workflow` Beispiel werden Details zu einem Workflow mit ID `1234567` abgerufen.

```
aws omics get-workflow \
  --id 1234567
```

Ausgabe:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
  "creationTime": "2022-11-30T22:33:16.225368Z",
```

```

    "digest":
    "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
    "engine": "WDL",
    "id": "1234567",
    "main": "workflow-crambam.wdl",
    "name": "cram-converter",
    "parameterTemplate": {
      "ref_dict": {
        "description": "dictionary file for 'ref_fasta'"
      },
      "ref_fasta_index": {
        "description": "Index of the reference genome fasta file"
      },
      "ref_fasta": {
        "description": "Reference genome fasta file"
      },
      "input_cram": {
        "description": "The Cram file to convert to BAM"
      },
      "sample_name": {
        "description": "The name of the input sample, used to name the output
BAM"
      }
    },
    "status": "ACTIVE",
    "statusMessage": "workflow-crambam.wdl\n    workflow CramToBamFlow\n
call CramToBamTask\n        call ValidateSamFile\n    task CramToBamTask\n    task
ValidateSamFile\n",
    "tags": {},
    "type": "PRIVATE"
  }
}

```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [GetWorkflow](#) in der AWS CLI Befehlsreferenz.

list-annotation-import-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-annotation-import-jobs`.

AWS CLI

Um eine Liste von Importjobs für Anmerkungen zu erhalten

Im Folgenden finden `list-annotation-import-jobs` Sie eine Liste von Importaufträgen für Anmerkungen.

```
aws omics list-annotation-import-jobs
```

Ausgabe:

```
{
  "annotationImportJobs": [
    {
      "creationTime": "2022-11-30T01:39:41.478294Z",
      "destinationName": "gff_ann_store",
      "id": "18a9e792-xmpl-4869-a105-e5b602900444",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "runLeftNormalization": false,
      "status": "COMPLETED",
      "updateTime": "2022-11-30T01:47:09.145178Z"
    },
    {
      "creationTime": "2022-11-30T00:45:58.007838Z",
      "destinationName": "my_ann_store",
      "id": "4e9eafc8-xmpl-431e-a0b2-3bda27cb600a",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "runLeftNormalization": false,
      "status": "FAILED",
      "updateTime": "2022-11-30T00:47:01.706325Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListAnnotationImportJobs](#) in der AWS CLI Befehlsreferenz.

list-annotation-store-versions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-annotation-store-versions`.

AWS CLI

Um alle Versionen eines Annotationspeichers aufzulisten.

Das folgende `list-annotation-store-versions` Beispiel listet alle Versionen auf, die es in einem Annotationsspeicher gibt.

```
aws omics list-annotation-store-versions \  
  --name my_annotation_store
```

Ausgabe:

```
{  
  "annotationStoreVersions": [  
    {  
      "storeId": "4934045d1c6d",  
      "id": "2a3f4a44aa7b",  
      "status": "CREATING",  
      "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/  
my_annotation_store/version/my_version_2",  
      "name": "my_annotation_store",  
      "versionName": "my_version_2",  
      "creationTime": "2023-07-21T17:20:59.380043+00:00",  
      "versionSizeBytes": 0  
    },  
    {  
      "storeId": "4934045d1c6d",  
      "id": "4934045d1c6d",  
      "status": "ACTIVE",  
      "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/  
my_annotation_store/version/my_version_1",  
      "name": "my_annotation_store",  
      "versionName": "my_version_1",  
      "creationTime": "2023-07-21T17:15:49.251040+00:00",  
      "updateTime": "2023-07-21T17:15:56.434223+00:00",  
      "statusMessage": "",  
      "versionSizeBytes": 0  
    }  
  ]  
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Neue Versionen von Annotationsspeichern erstellen](#).

- API-Einheiten finden Sie [ListAnnotationStoreVersions](#) in der AWS CLI Befehlsreferenz.

list-annotation-stores

Das folgende Codebeispiel zeigt, wie man es benutzt `list-annotation-stores`.

AWS CLI

Um eine Liste von Annotationsspeichern abzurufen

Im folgenden `list-annotation-stores` Beispiel wird eine Liste von Annotationsspeichern abgerufen.

```
aws omics list-annotation-stores
```

Ausgabe:

```
{
  "annotationStores": [
    {
      "creationTime": "2022-11-23T22:48:39.226492Z",
      "id": "0a91xmplc71f",
      "name": "my_ann_store",
      "reference": {
        "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
      },
      "status": "ACTIVE",
      "statusMessage": "",
      "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/my_ann_store",
      "storeFormat": "VCF",
      "storeSizeBytes": 0,
      "updateTime": "2022-11-23T22:53:27.372840Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListAnnotationStores](#) in der AWS CLI Befehlsreferenz.

list-multipart-read-set-uploads

Das folgende Codebeispiel zeigt, wie man es benutzt `list-multipart-read-set-uploads`.

AWS CLI

Um alle mehrteiligen Read-Set-Uploads und deren Status aufzulisten.

Das folgende `list-multipart-read-set-uploads` Beispiel listet alle Uploads von mehrteiligen Lesesätzen und deren Status auf.

```
aws omics list-multipart-read-set-uploads \  
--sequence-store-id 0123456789
```

Ausgabe:

```
{  
  "uploads":  
    [  
      {  
        "sequenceStoreId": "0123456789",  
        "uploadId": "8749584421",  
        "sourceFileType": "FASTQ",  
        "subjectId": "mySubject",  
        "sampleId": "mySample",  
        "generatedFrom": "1000 Genomes",  
        "name": "HG00146",  
        "description": "FASTQ for HG00146",  
        "creationTime": "2023-11-29T19:22:51.349298+00:00"  
      },  
      {  
        "sequenceStoreId": "0123456789",  
        "uploadId": "5290538638",  
        "sourceFileType": "BAM",  
        "subjectId": "mySubject",  
        "sampleId": "mySample",  
        "generatedFrom": "1000 Genomes",  
        "referenceArn": "arn:aws:omics:us-west-2:845448930428:referenceStore/8168613728/reference/2190697383",  
        "name": "HG00146",  
        "description": "BAM for HG00146",  
        "creationTime": "2023-11-29T19:23:33.116516+00:00"  
      },  
      {  
        "sequenceStoreId": "0123456789",  
        "uploadId": "4174220862",  
        "sourceFileType": "BAM",
```

```
    "subjectId": "mySubject",
    "sampleId": "mySample",
    "generatedFrom": "1000 Genomes",
    "referenceArn": "arn:aws:omics:us-
west-2:845448930428:referenceStore/8168613728/reference/2190697383",
    "name": "HG00147",
    "description": "BAM for HG00147",
    "creationTime": "2023-11-29T19:23:47.007866+00:00"
  }
]
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Direkter Upload in einen Sequenzspeicher](#).AWS HealthOmics

- API-Einheiten finden Sie [ListMultipartReadSetUploads](#) in der AWS CLI Befehlsreferenz.

list-read-set-activation-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-read-set-activation-jobs`.

AWS CLI

Um eine Liste von Readset-Aktivierungsaufträgen abzurufen

Im folgenden `list-read-set-activation-jobs` Beispiel wird eine Liste von Aktivierungsaufträgen für einen Sequenzspeicher mit ID abgerufen `1234567890`.

```
aws omics list-read-set-activation-jobs \
  --sequence-store-id 1234567890
```

Ausgabe:

```
{
  "activationJobs": [
    {
      "completionTime": "2022-12-06T22:33:42.828Z",
      "creationTime": "2022-12-06T22:32:45.213Z",
      "id": "1234567890",
      "sequenceStoreId": "1234567890",
      "status": "COMPLETED"
    },
  ],
}
```

```
{
  "creationTime": "2022-12-06T22:35:10.100Z",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "IN_PROGRESS"
}
]
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReadSetActivationJobs](#) in der AWS CLI Befehlsreferenz.

list-read-set-export-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-read-set-export-jobs`.

AWS CLI

To ruft eine Liste von Readset-Exportaufträgen ab

Im folgenden `list-read-set-export-jobs` Beispiel wird eine Liste von Exportaufträgen für einen Sequenzspeicher mit der ID abgerufen `1234567890`.

```
aws omics list-read-set-export-jobs \
  --sequence-store-id 1234567890
```

Ausgabe:

```
{
  "exportJobs": [
    {
      "completionTime": "2022-12-06T22:39:14.491Z",
      "creationTime": "2022-12-06T22:37:18.612Z",
      "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
      "id": "1234567890",
      "sequenceStoreId": "1234567890",
      "status": "COMPLETED"
    },
    {
      "creationTime": "2022-12-06T22:38:04.871Z",
      "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
      "id": "1234567890",
```



```
        "sequenceStoreId": "1234567890",
        "status": "IN_PROGRESS"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReadSetExportJobs](#) in der AWS CLI Befehlsreferenz.

list-read-set-import-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-read-set-import-jobs`.

AWS CLI

Um eine Liste von Readset-Importaufträgen zu erhalten

Im folgenden `list-read-set-import-jobs` Beispiel wird eine Liste von Importaufträgen für einen Sequenzspeicher mit der ID abgerufen `1234567890`.

```
aws omics list-read-set-import-jobs \
  --sequence-store-id 1234567890
```

Ausgabe:

```
{
  "importJobs": [
    {
      "completionTime": "2022-11-29T18:17:49.244Z",
      "creationTime": "2022-11-29T17:32:47.700Z",
      "id": "1234567890",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "sequenceStoreId": "1234567890",
      "status": "COMPLETED"
    },
    {
      "completionTime": "2022-11-23T22:01:34.090Z",
      "creationTime": "2022-11-23T21:52:43.289Z",
      "id": "1234567890",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
```

```
        "sequenceStoreId": "1234567890",
        "status": "COMPLETED_WITH_FAILURES"
    }
]
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReadSetImportJobs](#) in der AWS CLI Befehlsreferenz.

list-read-set-upload-parts

Das folgende Codebeispiel zeigt, wie man es benutzt `list-read-set-upload-parts`.

AWS CLI

Um alle Teile in einem angeforderten mehrteiligen Upload für einen Sequenzspeicher aufzulisten.

Das folgende `list-read-set-upload-parts` Beispiel listet alle Teile in einem angeforderten mehrteiligen Upload für einen Sequenzspeicher auf.

```
aws omics list-read-set-upload-parts \
  --sequence-store-id 0123456789 \
  --upload-id 1122334455 \
  --part-source SOURCE1
```

Ausgabe:

```
{
  "parts": [
    {
      "partNumber": 1,
      "partSize": 94371840,
      "file": "SOURCE1",
      "checksum":
"984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
      "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
    }
    {
      "partNumber": 2,
      "partSize": 10471840,
      "file": "SOURCE1",

```

```

        "checksum":
          "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
          "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
        }
      ]
    }
  }
}

```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Direkter Upload in einen Sequenzspeicher](#).

- API-Einheiten finden Sie [ListReadSetUploadParts](#) in der AWS CLI Befehlsreferenz.

list-read-sets

Das folgende Codebeispiel zeigt, wie man es benutzt `list-read-sets`.

AWS CLI

Um eine Liste von Read-Sets zu erhalten

Im folgenden `list-read-sets` Beispiel wird eine Liste von Lesesätzen für einen Sequenzspeicher mit der ID abgerufen `1234567890`.

```

aws omics list-read-sets \
  --sequence-store-id 1234567890

```

Ausgabe:

```

{
  "readSets": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/readSet/1234567890",
      "creationTime": "2022-11-23T21:55:00.515Z",
      "fileType": "FASTQ",
      "id": "1234567890",
      "name": "HG00146",
      "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890",
      "sampleId": "fastq-sample",
      "sequenceStoreId": "1234567890",
    }
  ]
}

```

```

        "status": "ACTIVE",
        "subjectId": "fastq-subject"
    }
]
}

```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReadSets](#) in der AWS CLI Befehlsreferenz.

list-reference-import-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-reference-import-jobs`.

AWS CLI

Um eine Liste von Referenzimportaufträgen abzurufen

Im folgenden `list-reference-import-jobs` Beispiel wird eine Liste von Referenzimportaufträgen für einen Referenzspeicher mit der ID abgerufen `1234567890`.

```

aws omics list-reference-import-jobs \
  --reference-store-id 1234567890

```

Ausgabe:

```

{
  "importJobs": [
    {
      "completionTime": "2022-11-23T19:54:58.204Z",
      "creationTime": "2022-11-23T19:53:20.729Z",
      "id": "1234567890",
      "referenceStoreId": "1234567890",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "status": "COMPLETED"
    },
    {
      "creationTime": "2022-11-23T20:34:03.250Z",
      "id": "1234567890",
      "referenceStoreId": "1234567890",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",

```

```
        "status": "IN_PROGRESS"
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReferenceImportJobs](#) in der AWS CLI Befehlsreferenz.

list-reference-stores

Das folgende Codebeispiel zeigt, wie man es benutzt `list-reference-stores`.

AWS CLI

Um eine Liste von Referenzgeschäften zu erhalten

Im folgenden `list-reference-stores` Beispiel wird eine Liste von Referenzspeichern abgerufen.

```
aws omics list-reference-stores
```

Ausgabe:

```
{
  "referenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
      "creationTime": "2022-11-22T22:13:25.947Z",
      "id": "1234567890",
      "name": "my-ref-store"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReferenceStores](#) in der AWS CLI Befehlsreferenz.

list-references

Das folgende Codebeispiel zeigt, wie man es benutzt `list-references`.

AWS CLI

Um eine Referenzliste zu erhalten

Das folgende `list-references` Beispiel ruft eine Liste von Genomreferenzen für einen Referenzspeicher mit ID `ab1234567890`.

```
aws omics list-references \  
  --reference-store-id 1234567890
```

Ausgabe:

```
{  
  "references": [  
    {  
      "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/  
reference/1234567890",  
      "creationTime": "2022-11-22T22:27:09.033Z",  
      "id": "1234567890",  
      "md5": "7ff134953dcca8c8997453bbb80b6b5e",  
      "name": "assembly-38",  
      "referenceStoreId": "1234567890",  
      "status": "ACTIVE",  
      "updateTime": "2022-11-22T22:27:09.033Z"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListReferences](#) in der AWS CLI Befehlsreferenz.

list-run-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `list-run-groups`.

AWS CLI

Um eine Liste von Run-Gruppen zu erhalten

Im folgenden `list-run-groups` Beispiel wird eine Liste von Ausführungsgruppen abgerufen.

aws omics list-run-groups

Ausgabe:

```
{
  "items": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
      "creationTime": "2022-12-01T00:58:42.915219Z",
      "id": "1234567",
      "maxCpus": 20,
      "maxDuration": 600,
      "name": "cram-convert"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListRunGroups](#) in der AWS CLI Befehlsreferenz.

list-run-tasks

Das folgende Codebeispiel zeigt, wie man es benutzt `list-run-tasks`.

AWS CLI

Um eine Liste von Aufgaben zu erhalten

Im folgenden `list-run-tasks` Beispiel wird eine Liste von Aufgaben für eine Workflow-Ausführung abgerufen.

```
aws omics list-run-tasks \
  --id 1234567
```

Ausgabe:

```
{
  "items": [
    {
      "cpus": 1,
      "creationTime": "2022-11-30T23:13:00.718651Z",
```

```
    "memory": 15,
    "name": "CramToBamTask",
    "startTime": "2022-11-30T23:17:47.016Z",
    "status": "COMPLETED",
    "stopTime": "2022-11-30T23:18:21.503Z",
    "taskId": "1234567"
  },
  {
    "cpus": 1,
    "creationTime": "2022-11-30T23:18:32.315606Z",
    "memory": 4,
    "name": "ValidateSamFile",
    "startTime": "2022-11-30T23:23:40.165Z",
    "status": "COMPLETED",
    "stopTime": "2022-11-30T23:24:14.766Z",
    "taskId": "1234567"
  }
]
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListRunTasks](#) in der AWS CLI Befehlsreferenz.

list-runs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-runs`.

AWS CLI

Um eine Liste von Workflow-Läufen abzurufen

Im folgenden `list-runs` Beispiel wird eine Liste von Workflow-Ausführungen abgerufen.

```
aws omics list-runs
```

Ausgabe:

```
{
  "items": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
      "creationTime": "2022-12-02T23:20:01.202074Z",
```



```

    "id": "1234567",
    "name": "cram-to-bam",
    "priority": 1,
    "startTime": "2022-12-02T23:29:18.115Z",
    "status": "COMPLETED",
    "stopTime": "2022-12-02T23:57:54.428812Z",
    "storageCapacity": 10,
    "workflowId": "1234567"
  },
  {
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "creationTime": "2022-12-03T00:16:57.180066Z",
    "id": "1234567",
    "name": "cram-to-bam",
    "priority": 1,
    "startTime": "2022-12-03T00:26:50.233Z",
    "status": "FAILED",
    "stopTime": "2022-12-03T00:37:21.451340Z",
    "storageCapacity": 10,
    "workflowId": "1234567"
  },
  {
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "creationTime": "2022-12-05T17:57:08.444817Z",
    "id": "1234567",
    "name": "cram-to-bam",
    "status": "STARTING",
    "workflowId": "1234567"
  }
]
}

```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListRuns](#) in der AWS CLI Befehlsreferenz.

list-sequence-stores

Das folgende Codebeispiel zeigt, wie man es benutzt `list-sequence-stores`.

AWS CLI

Um eine Liste von Sequenzspeichern abzurufen

Im folgenden `list-sequence-stores` Beispiel wird eine Liste von Sequenzspeichern abgerufen.

```
aws omics list-sequence-stores
```

Ausgabe:

```
{
  "sequenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
      "creationTime": "2022-11-23T01:24:33.629Z",
      "id": "1234567890",
      "name": "my-seq-store"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListSequenceStores](#) in der AWS CLI Befehlsreferenz.

list-shares

Das folgende Codebeispiel zeigt, wie man es benutzt `list-shares`.

AWS CLI

Um die verfügbaren Anteile an HealthOmics Analysedaten aufzulisten

Das folgende `list-shares` Beispiel listet alle Shares auf, die für einen Ressourcenbesitzer erstellt wurden.

```
aws omics list-shares \
  --resource-owner SELF
```

Ausgabe:

```
{
  "shares": [
    {
      "shareId": "595c1cbd-a008-4eca-a887-954d30c91c6e",

```

```

    "name": "myShare",
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_1",
    "principalSubscriber": "123456789012",
    "ownerId": "555555555555",
    "status": "PENDING"
  }
  {
    "shareId": "39b65d0d-4368-4a19-9814-b0e31d73c10a",
    "name": "myShare3456",
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_2",
    "principalSubscriber": "123456789012",
    "ownerId": "555555555555",
    "status": "ACTIVE"
  },
  {
    "shareId": "203152f5-eef9-459d-a4e0-a691668d44ef",
    "name": "myShare4",
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_3",
    "principalSubscriber": "123456789012",
    "ownerId": "555555555555",
    "status": "ACTIVE"
  }
]
}

```

Weitere Informationen finden Sie unter [Kontoübergreifendes Teilen](#) im AWS HealthOmics Benutzerhandbuch.

- API-Einheiten finden Sie [ListShares](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um eine Liste von Tags zu erhalten

Im folgenden `list-tags-for-resource` Beispiel wird eine Liste von Tags für einen Workflow mit ID abgerufen `1234567`.

```
aws omics list-tags-for-resource \  
--resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567
```

Ausgabe:

```
{  
  "tags": {  
    "department": "analytics"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging resources in Amazon Omics im Amazon Omics Developer Guide](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in AWS CLI der Befehlsreferenz.

list-variant-import-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-variant-import-jobs`.

AWS CLI

Um eine Liste von Variantenimportaufträgen zu erhalten

Im folgenden `list-variant-import-jobs` Beispiel wird eine Liste von Variantenimportaufträgen abgerufen.

```
aws omics list-variant-import-jobs
```

Ausgabe:

```
{  
  "variantImportJobs": [  
    {  
      "creationTime": "2022-11-23T22:47:02.514002Z",  
      "destinationName": "my_var_store",  
      "id": "69cb65d6-xmpl-4a4a-9025-4565794b684e",  
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-  
serviceRole-W801XMPL7QZ",  
      "runLeftNormalization": false,  
    }  
  ]  
}
```

```

        "status": "COMPLETED",
        "updateTime": "2022-11-23T22:49:17.976597Z"
    },
    {
        "creationTime": "2022-11-23T22:42:50.037812Z",
        "destinationName": "my_var_store",
        "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",
        "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
        "runLeftNormalization": false,
        "status": "COMPLETED",
        "updateTime": "2022-11-23T22:45:26.009880Z"
    }
]
}

```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListVariantImportJobs](#) in der AWS CLI Befehlsreferenz.

list-variant-stores

Das folgende Codebeispiel zeigt, wie man es benutzt `list-variant-stores`.

AWS CLI

Um eine Liste von Variantengeschäften zu erhalten

Im folgenden `list-variant-stores` Beispiel wird eine Liste von Variantenspeichern abgerufen.

```
aws omics list-variant-stores
```

Ausgabe:

```

{
  "variantStores": [
    {
      "creationTime": "2022-11-23T22:09:07.534499Z",
      "id": "02dexmplcfdd",
      "name": "my_var_store",
      "reference": {

```

```
        "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
      },
      "status": "CREATING",
      "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/my_var_store",
      "storeSizeBytes": 0,
      "updateTime": "2022-11-23T22:09:24.931711Z"
    },
    {
      "creationTime": "2022-09-23T23:00:09.140265Z",
      "id": "8777xmpl1a24",
      "name": "myvstore0",
      "status": "ACTIVE",
      "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/myvstore0",
      "storeSizeBytes": 0,
      "updateTime": "2022-09-23T23:03:26.013220Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListVariantStores](#) in der AWS CLI Befehlsreferenz.

list-workflows

Das folgende Codebeispiel zeigt, wie man es benutzt `list-workflows`.

AWS CLI

Um eine Liste von Workflows abzurufen

Im folgenden `list-workflows` Beispiel wird eine Liste von Workflows abgerufen.

```
aws omics list-workflows
```

Ausgabe:

```
{
  "items": [
```

```

    {
      "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
      "creationTime": "2022-09-23T23:08:22.041227Z",
      "digest": "nSCNo/qMWFxmplXpUdokXJnwgne0axyyc2Y0xVxrJTE=",
      "id": "1234567",
      "name": "my-wkflow-0",
      "status": "ACTIVE",
      "type": "PRIVATE"
    },
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
      "creationTime": "2022-11-30T22:33:16.225368Z",
      "digest":
"sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
      "id": "1234567",
      "name": "cram-converter",
      "status": "ACTIVE",
      "type": "PRIVATE"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [ListWorkflows](#) in der AWS CLI Befehlsreferenz.

start-annotation-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `start-annotation-import-job`.

AWS CLI

Um Anmerkungen zu importieren

Das folgende `start-annotation-import-job` Beispiel importiert Anmerkungen aus Amazon S3.

```

aws omics start-annotation-import-job \
  --destination-name tsv_ann_store \
  --no-run-left-normalization \
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ \
  --items source=s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz

```

Ausgabe:

```
{
  "jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartAnnotationImportJob](#) in der AWS CLI Befehlsreferenz.

start-read-set-activation-job

Das folgende Codebeispiel zeigt, wie man es benutzt `start-read-set-activation-job`.

AWS CLI

Um einen archivierten Lesesatz zu aktivieren

Im folgenden `start-read-set-activation-job` Beispiel werden zwei Lesesätze aktiviert.

```
aws omics start-read-set-activation-job \
  --sequence-store-id 1234567890 \
  --sources readSetId=1234567890 readSetId=1234567890
```

Ausgabe:

```
{
  "creationTime": "2022-12-06T22:35:10.100Z",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartReadSetActivationJob](#) in der AWS CLI Befehlsreferenz.

start-read-set-export-job

Das folgende Codebeispiel zeigt, wie man es benutzt `start-read-set-export-job`.

AWS CLI

Um einen Lesesatz zu exportieren

Das folgende `start-read-set-export-job` Beispiel exportiert zwei Lesesätze nach Amazon S3.

```
aws omics start-read-set-export-job \  
  --sequence-store-id 1234567890 \  
  --sources readSetId=1234567890 readSetId=1234567890 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ  
\  
  --destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

Ausgabe:

```
{  
  "creationTime": "2022-12-06T22:37:18.612Z",  
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",  
  "id": "1234567890",  
  "sequenceStoreId": "1234567890",  
  "status": "SUBMITTED"  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartReadSetExportJob](#) in der AWS CLI Befehlsreferenz.

start-read-set-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `start-read-set-import-job`.

AWS CLI

Um einen Lesesatz zu importieren

Im folgenden `start-read-set-import-job` Beispiel wird ein Lesesatz importiert.

```
aws omics start-read-set-import-job \  
  --sequence-store-id 1234567890 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ \  
  --destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-import/
```

```
--sources file://readset-sources.json
```

readset-sources.json ist ein JSON Dokument mit dem folgenden Inhalt.

```
[
  {
    "sourceFiles":
    {
      "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"
    },
    "sourceFileType": "BAM",
    "subjectId": "bam-subject",
    "sampleId": "bam-sample",
    "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
    "name": "HG00100"
  }
]
```

Ausgabe:

```
{
  "creationTime": "2022-11-23T01:36:38.158Z",
  "id": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartReadSetImportJob](#) in der AWS CLI Befehlsreferenz.

start-reference-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `start-reference-import-job`.

AWS CLI

Um ein Referenzgenom zu importieren

Das folgende `start-reference-import-job` Beispiel importiert ein Referenzgenom aus Amazon S3.

```
aws omics start-reference-import-job \  
  --reference-store-id 1234567890 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ \  
  --sources sourceFile=s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta,name=assembly-38
```

Ausgabe:

```
{  
  "creationTime": "2022-11-22T22:25:41.124Z",  
  "id": "1234567890",  
  "referenceStoreId": "1234567890",  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "status": "SUBMITTED"  
}
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartReferenceImportJob](#) in der AWS CLI Befehlsreferenz.

start-run

Das folgende Codebeispiel zeigt, wie man es benutzt `start-run`.

AWS CLI

Um einen Workflow auszuführen

Im folgenden `start-run` Beispiel wird ein Workflow mit ID ausgeführt 1234567.

```
aws omics start-run \  
  --workflow-id 1234567 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ \  
  --name 'cram-to-bam' \  
  --output-uri s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/ \  
  --run-group-id 1234567 \  
  --
```

```
--priority 1 \  
--storage-capacity 10 \  
--log-level ALL \  
--parameters file://workflow-inputs.json
```

workflow-inputs.json ist ein JSON Dokument mit dem folgenden Inhalt.

```
{  
  "sample_name": "NA12878",  
  "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram",  
  "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.dict",  
  "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta",  
  "ref_fasta_index": "omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta.fai"  
}
```

Ausgabe:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
  "id": "1234567",  
  "status": "PENDING",  
  "tags": {}  
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

So laden Sie Quelldateien von Amazon Omics

Sie können Quelldateien auch aus dem Amazon Omics-Speicher laden, indem Sie URIs dienstspezifisch verwenden. Die folgende Beispieldatei workflow-inputs.json verwendet Amazon Omics URIs für Lesesatz- und Referenzgenomquellen.

```
{  
  "sample_name": "NA12878",  
  "input_cram": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/  
readSet/1234567890/source1",  
  "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.dict",
```

```
"ref_fasta": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/reference/1234567890",
"ref_fasta_index": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/reference/1234567890/index"
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartRun](#) in der AWS CLI Befehlsreferenz.

start-variant-import-job

Das folgende Codebeispiel zeigt, wie man es benutzt `start-variant-import-job`.

AWS CLI

Um eine Variantendatei zu importieren

Im folgenden `start-variant-import-job` Beispiel wird eine VCF Formatvariantendatei importiert.

```
aws omics start-variant-import-job \
  --destination-name my_var_store \
  --no-run-left-normalization \
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-WS01XMPL7QZ \
  --items source=s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.known_indels.vcf.gz
```

Ausgabe:

```
{
  "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [StartVariantImportJob](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Das folgende `tag-resource` Beispiel fügt einem Workflow ein `department` Tag mit der ID `hinzu1234567`.

```
aws omics tag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tags department=analytics
```

Weitere Informationen finden Sie unter [Tagging resources in Amazon Omics im Amazon Omics Developer Guide](#).

- API-Einheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das `department` Tag aus einem Workflow entfernt.

```
aws omics untag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tag-keys department
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-annotation-store

Das folgende Codebeispiel zeigt, wie man es benutzt `update-annotation-store`.

AWS CLI

Um einen Annotationsspeicher zu aktualisieren

Im folgenden `update-annotation-store` Beispiel wird die Beschreibung eines Annotationsspeichers mit dem Namen aktualisiert `my_vcf_store`.

```
aws omics update-annotation-store \  
  --name my_vcf_store \  
  --description "VCF annotation store"
```

Ausgabe:

```
{  
  "creationTime": "2022-12-05T18:00:56.101860Z",  
  "description": "VCF annotation store",  
  "id": "bd6axmpl2444",  
  "name": "my_vcf_store",  
  "reference": {  
    "referenceArn": "arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890"  
  },  
  "status": "ACTIVE",  
  "storeFormat": "VCF",  
  "updateTime": "2022-12-05T18:13:16.100051Z"  
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [UpdateAnnotationStore](#) in der AWS CLI Befehlsreferenz.

update-run-group

Das folgende Codebeispiel zeigt, wie man es benutzt `update-run-group`.

AWS CLI

Um eine Ausführungsgruppe zu aktualisieren

Im folgenden `update-run-group` Beispiel werden die Einstellungen einer Ausführungsgruppe mit der ID aktualisiert `1234567`.

```
aws omics update-run-group \  
  --id 1234567 \  
  --max-cpus 10
```

Ausgabe:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
  "creationTime": "2022-12-01T00:58:42.915219Z",
  "id": "1234567",
  "maxCpus": 10,
  "maxDuration": 600,
  "name": "cram-convert",
  "tags": {}
}
```

Weitere Informationen finden Sie unter [Omics Workflows](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [UpdateRunGroup](#) in der AWS CLI Befehlsreferenz.

update-variant-store

Das folgende Codebeispiel zeigt, wie man es benutzt `update-variant-store`.

AWS CLI

Um einen Variantenspeicher zu aktualisieren

Im folgenden `update-variant-store` Beispiel wird die Beschreibung eines Variantenspeichers mit dem Namen `aktualisiertmy_var_store`.

```
aws omics update-variant-store \
  --name my_var_store \
  --description "variant store"
```

Ausgabe:

```
{
  "creationTime": "2022-11-23T22:09:07.534499Z",
  "description": "variant store",
  "id": "02dexmplcfdd",
  "name": "my_var_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
}
```



```
"status": "ACTIVE",  
"updateTime": "2022-12-05T18:23:37.686402Z"  
}
```

Weitere Informationen finden Sie unter [Omics Analytics](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [UpdateVariantStore](#) in der AWS CLI Befehlsreferenz.

update-workflow

Das folgende Codebeispiel zeigt, wie man es benutzt `update-workflow`.

AWS CLI

Um einen Workflow zu aktualisieren

Im folgenden `update-workflow` Beispiel wird die Beschreibung eines Workflows mit der ID `aktualisiert1234567`.

```
aws omics update-workflow \  
  --id 1234567 \  
  --description "copy workflow"
```

Weitere Informationen finden Sie unter [Omics Storage](#) im Amazon Omics Developer Guide.

- API-Einheiten finden Sie [UpdateWorkflow](#) in der AWS CLI Befehlsreferenz.

upload-read-set-part

Das folgende Codebeispiel zeigt, wie man es benutzt `upload-read-set-part`.

AWS CLI

Um einen Teil des Lesesatzes hochzuladen.

Im folgenden `upload-read-set-part` Beispiel wird ein bestimmter Teil eines Lesesatzes hochgeladen.

```
aws omics upload-read-set-part \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --part-source SOURCE1 \  
  --part-number 1
```

```
--part-number 1 \  
--payload /path/to/file/read_1_part_1.fastq.gz
```

Ausgabe:

```
{  
  "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"  
}
```

Weitere Informationen finden Sie im AWS HealthOmics Benutzerhandbuch unter [Direkter Upload in einen Sequenzspeicher](#).

- API-Einheiten finden Sie [UploadReadSetPart](#) in der AWS CLI Befehlsreferenz.

IAM-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren IAM.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-client-id-to-open-id-connect-provider

Das folgende Codebeispiel zeigt die Verwendung `add-client-id-to-open-id-connect-provider`.

AWS CLI

Um eine Client-ID (Audience) zu einem Open-ID Connect (OIDC) -Anbieter hinzuzufügen

Der folgende `add-client-id-to-open-id-connect-provider` Befehl fügt dem genannten OIDC `server.example.com` Anbieter `my-application-ID` die Client-ID hinzu.

```
aws iam add-client-id-to-open-id-connect-provider \  
  --client-id my-application-ID \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `create-open-id-connect-provider` Befehl, um einen OIDC Anbieter zu erstellen.

Weitere Informationen finden Sie unter [Erstellen von OpenID Connect \(OIDC\) -Identitätsanbietern](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie unter [AddClientIdToOpenIdConnectProvider AWS CLI](#) Befehlsreferenz.

add-role-to-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `add-role-to-instance-profile`.

AWS CLI

Um einem Instanzprofil eine Rolle hinzuzufügen

Der folgende `add-role-to-instance-profile` Befehl fügt die benannte Rolle `S3Access` dem genannten Instanzprofil hinzu `Webserver`.

```
aws iam add-role-to-instance-profile \  
  --role-name S3Access \  
  --instance-profile-name Webserver
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `create-instance-profile` Befehl, um ein Instanzprofil zu erstellen.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden](#).

- API-Einheiten finden Sie [AddRoleToInstanceProfile](#) in der AWS CLI Befehlsreferenz.

add-user-to-group

Das folgende Codebeispiel zeigt die Verwendung `add-user-to-group`.

AWS CLI

Um einen Benutzer zu einer IAM Gruppe hinzuzufügen

Mit dem folgenden `add-user-to-group` Befehl wird der IAM Gruppe mit dem Namen Bob ein IAM Benutzer mit dem Namen hinzugefügt Admins.

```
aws iam add-user-to-group \  
  --user-name Bob \  
  --group-name Admins
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Hinzufügen und Entfernen von Benutzern in einer IAM AWS IAM Benutzergruppe](#).

- API-Einheiten finden Sie [AddUserToGroup](#) unter AWS CLI Befehlsreferenz.

attach-group-policy

Das folgende Codebeispiel zeigt die Verwendung `attach-group-policy`.

AWS CLI

Um eine verwaltete Richtlinie an eine IAM Gruppe anzuhängen

Mit dem folgenden `attach-group-policy` Befehl wird die benannte AWS verwaltete Richtlinie `ReadOnlyAccess` an die angegebene IAM Gruppe angehängt. Finance

```
aws iam attach-group-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --group-name Finance
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltete Richtlinien und Inline-Richtlinien](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [AttachGroupPolicy](#) in der AWS CLI Befehlsreferenz.

attach-role-policy

Das folgende Codebeispiel zeigt die Verwendung `attach-role-policy`.

AWS CLI

Um einer IAM Rolle eine verwaltete Richtlinie zuzuordnen

Mit dem folgenden `attach-role-policy` Befehl wird die benannte AWS verwaltete Richtlinie `ReadOnlyAccess` an die angegebene IAM Rolle angehängt. `ReadOnlyRole`

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --role-name ReadOnlyRole
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltete Richtlinien und Inline-Richtlinien](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [AttachRolePolicy](#) in der AWS CLI Befehlsreferenz.

attach-user-policy

Das folgende Codebeispiel zeigt die Verwendung `attach-user-policy`.

AWS CLI

Um einem IAM Benutzer eine verwaltete Richtlinie anzuhängen

Mit dem folgenden `attach-user-policy` Befehl wird die benannte AWS verwaltete Richtlinie `AdministratorAccess` an den genannten IAM Alice Benutzer angehängt.

```
aws iam attach-user-policy \  
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \  
  --user-name Alice
```

```
--user-name Alice
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltete Richtlinien und Inline-Richtlinien](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [AttachUserPolicy](#) in der AWS CLI Befehlsreferenz.

change-password

Das folgende Codebeispiel zeigt die Verwendung `change-password`.

AWS CLI

Um das Passwort für Ihren IAM Benutzer zu ändern

Um das Passwort für Ihren IAM Benutzer zu ändern, empfehlen wir, den `--cli-input-json` Parameter zu verwenden, um eine JSON Datei zu übergeben, die Ihre alten und neuen Passwörter enthält. Mit dieser Methode können Sie sichere Passwörter mit nicht alphanumerischen Zeichen verwenden. Es kann schwierig sein, Passwörter mit nicht alphanumerischen Zeichen zu verwenden, wenn Sie sie als Befehlszeilenparameter übergeben. Um den `--cli-input-json` Parameter zu verwenden, verwenden Sie zunächst den `change-password` Befehl mit dem `--generate-cli-skeleton` Parameter, wie im folgenden Beispiel.

```
aws iam change-password \  
  --generate-cli-skeleton > change-password.json
```

Mit dem vorherigen Befehl wird eine JSON Datei mit dem Namen `change-password.json` erstellt, mit der Sie Ihre alten und neuen Passwörter eingeben können. Die Datei könnte beispielsweise wie folgt aussehen.

```
{  
  "OldPassword": "3s0K_;xh4~8XXI",  
  "NewPassword": "]35d/{pB9Fo9wJ"  
}
```

Verwenden Sie als Nächstes den `change-password` Befehl erneut, um Ihr Passwort zu ändern. Übergeben Sie diesmal den `--cli-input-json` Parameter zur Angabe Ihrer JSON Datei. Der

folgende `change-password` Befehl verwendet den `--cli-input-json` Parameter mit einer JSON Datei namens `change-password.json`.

```
aws iam change-password \  
  --cli-input-json file://change-password.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Dieser Befehl kann nur von Benutzern aufgerufen werden. IAM Wenn dieser Befehl mit AWS Kontoanmeldeinformationen (Root) aufgerufen wird, gibt der Befehl einen `InvalidUserType` Fehler zurück.

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter So ändert ein AWS IAMBenutzer sein eigenes Passwort](#).

- APIEinzelheiten finden Sie [ChangePassword](#) in der AWS CLI Befehlsreferenz.

create-access-key

Das folgende Codebeispiel zeigt die Verwendung `create-access-key`.

AWS CLI

Um einen Zugriffsschlüssel für einen IAM Benutzer zu erstellen

Der folgende `create-access-key` Befehl erstellt einen Zugriffsschlüssel (Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel) für den genannten IAM Benutzer `Bob`.

```
aws iam create-access-key \  
  --user-name Bob
```

Ausgabe:

```
{  
  "AccessKey": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CreateDate": "2015-03-09T18:39:23.411Z",  
    "SecretAccessKey": "wJa1rXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"  
  }  
}
```

```
}
```

Speichern Sie den geheimen Zugriffsschlüssel an einem sicheren Ort. Bei Verlust kann er nicht wiederhergestellt werden und Sie müssen einen neuen Zugriffsschlüssel erstellen.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verwaltung von Zugriffsschlüsseln für IAM AWS IAM Benutzer](#).

- API-Einheiten finden Sie [CreateAccessKey](#) in der AWS CLI Befehlsreferenz.

create-account-alias

Das folgende Codebeispiel zeigt die Verwendung `create-account-alias`.

AWS CLI

So erstellen Sie einen Konto-Alias

Der folgende `create-account-alias` Befehl erstellt den Alias `examplecorp` für Ihr AWS Konto.

```
aws iam create-account-alias \  
  --account-alias examplecorp
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ihre AWS Konto-ID und deren Alias](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [CreateAccountAlias](#) in der AWS CLI Befehlsreferenz.

create-group

Das folgende Codebeispiel zeigt die Verwendung `create-group`.

AWS CLI

Um eine IAM Gruppe zu erstellen

Der folgende `create-group` Befehl erstellt eine IAM Gruppe mit dem Namen `Admins`.

```
aws iam create-group \  
  --group-name Admins
```



```
--group-name Admins
```

Ausgabe:

```
{
  "Group": {
    "Path": "/",
    "CreateDate": "2015-03-09T20:30:24.940Z",
    "GroupId": "AIDGPM59R04H3FEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "GroupName": "Admins"
  }
}
```

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Benutzergruppen erstellen](#).AWS IAM

- APIEinzelheiten finden Sie [CreateGroup](#)unter AWS CLI Befehlsreferenz.

create-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `create-instance-profile`.

AWS CLI

So erstellen Sie ein Instance-Profil

Der folgende `create-instance-profile`-Befehl erstellt ein Instance-Profil mit dem Namen `Webserver`.

```
aws iam create-instance-profile \  
  --instance-profile-name Webserver
```

Ausgabe:

```
{
  "InstanceProfile": {
    "InstanceProfileId": "AIPA5MBYC7DLSPEXAMPLE",
    "Roles": [],
    "CreateDate": "2015-03-09T20:33:19.626Z",
    "InstanceProfileName": "Webserver",
    "Path": "/",
  }
}
```

```
    "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"
  }
}
```

Verwenden Sie den `add-role-to-instance-profile`-Befehl, um einem Instance-Profil eine Rolle hinzuzufügen.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden](#).

- API-Einheiten finden Sie [CreateInstanceProfile](#) in der AWS CLI Befehlsreferenz.

create-login-profile

Das folgende Codebeispiel zeigt die Verwendung `create-login-profile`.

AWS CLI

Um ein Passwort für einen IAM Benutzer zu erstellen

Um ein Passwort für einen IAM Benutzer zu erstellen, empfehlen wir, den `--cli-input-json` Parameter zu verwenden, um eine JSON-Datei zu übergeben, die das Passwort enthält. Mit dieser Methode können Sie ein sicheres Passwort mit nicht-alphanumerischen Zeichen erstellen. Es kann schwierig sein, ein Passwort mit nicht-alphanumerischen Zeichen zu erstellen, wenn Sie es als Befehlszeilenparameter übergeben.

Um den `--cli-input-json` Parameter zu verwenden, verwenden Sie zunächst den `create-login-profile` Befehl mit dem `--generate-cli-skeleton` Parameter, wie im folgenden Beispiel.

```
aws iam create-login-profile \
  --generate-cli-skeleton > create-login-profile.json
```

Mit dem vorherigen Befehl wird eine JSON-Datei namens `create-login-profile.json` erstellt, mit der Sie die Informationen für einen nachfolgenden `create-login-profile` Befehl eingeben können. Beispielsweise:

```
{
  "UserName": "Bob",
```

```
"Password": "&1-3a6u:RA0djs",
"PasswordResetRequired": true
}
```

Verwenden Sie als Nächstes den `create-login-profile` Befehl erneut, um ein Passwort für einen IAM Benutzer zu erstellen. Übergeben Sie diesmal den `--cli-input-json` Parameter zur Angabe Ihrer JSON Datei. Der folgende `create-login-profile` Befehl verwendet den `--cli-input-json` Parameter mit einer JSON Datei namens `create-login-profile.json`.

```
aws iam create-login-profile \
  --cli-input-json file://create-login-profile.json
```

Ausgabe:

```
{
  "LoginProfile": {
    "UserName": "Bob",
    "CreateDate": "2015-03-10T20:55:40.274Z",
    "PasswordResetRequired": true
  }
}
```

Wenn das neue Passwort gegen die Kontopasswortrichtlinie verstößt, gibt der Befehl einen `PasswordPolicyViolation` Fehler zurück.

Um das Passwort für einen Benutzer zu ändern, der bereits eines hat, verwenden Sie `update-login-profile`. Verwenden Sie den `update-account-password-policy` Befehl, um eine Kennwortrichtlinie für das Konto festzulegen.

Wenn die Kontokennwortrichtlinie dies zulässt, können IAM Benutzer ihre eigenen Passwörter mithilfe des `change-password` Befehls ändern.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Passwörter für IAM AWS IAM Benutzer verwalten](#).

- API-Einheiten finden Sie [CreateLoginProfile](#) in der AWS CLI Befehlsreferenz.

create-open-id-connect-provider

Das folgende Codebeispiel zeigt die Verwendung `create-open-id-connect-provider`.

AWS CLI

Um einen OpenID Connect (OIDC) -Anbieter zu erstellen

Um einen OpenID Connect (OIDC) -Anbieter zu erstellen, empfehlen wir, den `--cli-input-json` Parameter zu verwenden, um eine JSON Datei zu übergeben, die die erforderlichen Parameter enthält. Wenn Sie einen OIDC Anbieter erstellen, müssen Sie den Namen URL des Anbieters übergeben, und der URL muss mit `https://` beginnen. Es kann schwierig sein, den URL als Befehlszeilenparameter zu übergeben, da der Doppelpunkt (`:`) und der Schrägstrich (`/`) in manchen Befehlszeilenumgebungen eine besondere Bedeutung haben. Durch die Verwendung des `--cli-input-json` Parameters wird diese Einschränkung umgangen.

Um den `--cli-input-json` Parameter zu verwenden, verwenden Sie zunächst den `create-open-id-connect-provider` Befehl mit dem `--generate-cli-skeleton` Parameter, wie im folgenden Beispiel.

```
aws iam create-open-id-connect-provider \  
  --generate-cli-skeleton > create-open-id-connect-provider.json
```

Mit dem vorherigen Befehl wird eine JSON Datei mit dem Namen `create-open-id-connect-provider.json` erstellt, mit der Sie die Informationen für einen nachfolgenden Befehl eingeben können. `create-open-id-connect-provider` Beispielsweise:

```
{  
  "Url": "https://server.example.com",  
  "ClientIDList": [  
    "example-application-ID"  
  ],  
  "ThumbprintList": [  
    "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"  
  ]  
}
```

Verwenden Sie als Nächstes den `create-open-id-connect-provider` Befehl erneut, um den OpenID Connect (OIDC) -Anbieter zu erstellen. Übergeben Sie diesmal den `--cli-input-json` Parameter zur Angabe Ihrer JSON Datei. Der folgende `create-open-id-connect-provider` Befehl verwendet den `--cli-input-json` Parameter mit einer JSON Datei namens `create-open-id-connect-provider.json`.

```
aws iam create-open-id-connect-provider \  
  --cli-input-json create-open-id-connect-provider.json
```

```
--cli-input-json file://create-open-id-connect-provider.json
```

Ausgabe:

```
{
  "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

Weitere Informationen zu OIDC Anbietern finden Sie unter [Erstellen von OpenID Connect \(OIDC\) -Identitätsanbietern](#) im AWS IAMBenutzerhandbuch.

Weitere Informationen zum Abrufen von Fingerabdrücken für einen OIDC Anbieter finden Sie unter [Abrufen des Fingerabdrucks für einen OpenID Connect-Identitätsanbieter](#) im Benutzerhandbuch.AWS IAM

- APIEinzelheiten finden Sie in der Befehlsreferenz. [CreateOpenIdConnectProvider](#)AWS CLI

create-policy-version

Das folgende Codebeispiel zeigt die Verwendungcreate-policy-version.

AWS CLI

So erstellen Sie eine neue Version einer verwalteten Richtlinie

In diesem Beispiel wird eine neue v2 Version der IAM Richtlinie erstellt, deren Standardversion es ARN ist, `arn:aws:iam::123456789012:policy/MyPolicy` und macht sie zur Standardversion.

```
aws iam create-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --policy-document file://NewPolicyVersion.json \  
  --set-as-default
```

Ausgabe:

```
{
  "PolicyVersion": {
    "CreateDate": "2015-06-16T18:56:03.721Z",
```

```
    "VersionId": "v2",
    "IsDefaultVersion": true
  }
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMVersionierungsrichtlinien](#).

- APIEinzelheiten finden Sie [CreatePolicyVersion](#) in der AWS CLI Befehlsreferenz.

create-policy

Das folgende Codebeispiel zeigt die Verwendung `create-policy`.

AWS CLI

Beispiel 1: So erstellen Sie eine vom Kunden verwaltete Richtlinie

Mit dem folgenden Befehl wird eine vom Kunden verwaltete Richtlinie mit dem Namen `my-policy` erstellt.

```
aws iam create-policy \
  --policy-name my-policy \
  --policy-document file://policy
```

Die Datei `policy` ist ein JSON Dokument im aktuellen Ordner, das nur Lesezugriff auf den `shared` Ordner in einem Amazon S3 S3-Bucket mit dem Namen `gewährmy-bucket`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket/shared/*"
      ]
    }
  ]
}
```

```
}

```

Ausgabe:

```
{
  "Policy": {
    "PolicyName": "my-policy",
    "CreateDate": "2015-06-01T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::0123456789012:policy/my-policy",
    "UpdateDate": "2015-06-01T19:31:18.620Z"
  }
}
```

Weitere Informationen zur Verwendung von Dateien als Eingabe für Zeichenkettenparameter finden [Sie unter Angeben von Parameterwerten für AWS CLI im AWS CLIBenutzerhandbuch](#).

Beispiel 2: So erstellen Sie eine vom Kunden verwaltete Richtlinie mit einer Beschreibung

Mit dem folgenden Befehl wird eine vom Kunden verwaltete Richtlinie mit dem Namen `my-policy` und einer unveränderlichen Beschreibung erstellt:

```
aws iam create-policy \
  --policy-name my-policy \
  --policy-document file://policy.json \
  --description "This policy grants access to all Put, Get, and List actions for my-bucket"
```

Die Datei `policy.json` ist ein JSON Dokument im aktuellen Ordner, das Zugriff auf alle Put-, List- und Get-Aktionen für einen Amazon S3 S3-Bucket mit dem Namen `my-bucket`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```

        "s3:ListBucket*",
        "s3:PutBucket*",
        "s3:GetBucket*"
    ],
    "Resource": [
        "arn:aws:s3:::my-bucket"
    ]
}
]
}

```

Ausgabe:

```

{
  "Policy": {
    "PolicyName": "my-policy",
    "PolicyId": "ANPAWGSUGIDPEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:policy/my-policy",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2023-05-24T22:38:47+00:00",
    "UpdateDate": "2023-05-24T22:38:47+00:00"
  }
}

```

Weitere Informationen zu identitätsbasierten Richtlinien finden Sie unter [Identitätsbasierte Richtlinien und ressourcenbasierte Richtlinien](#) im Benutzerhandbuch.AWS IAM

Beispiel 3: So erstellen Sie eine vom Kunden verwaltete Richtlinie mit Tags

Mit dem folgenden Befehl wird eine vom Kunden verwaltete Richtlinie mit dem Namen `my-policy` mit Tags erstellt. In diesem Beispiel wird der `--tags` Parameter flag mit den folgenden Tags im `-Format` verwendet: `JSON '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'` Alternativ kann das `--tags`-Flag mit Tags im Kurzformat `'Key=Department,Value=Accounting Key=Location,Value=Seattle'` verwendet werden.

```
aws iam create-policy \
```



```
--policy-name my-policy \  
--policy-document file://policy.json \  
--tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
"Value": "Seattle"}'
```

Die Datei `policy.json` ist ein JSON Dokument im aktuellen Ordner, das Zugriff auf alle Put-, List- und Get-Aktionen für einen Amazon S3 S3-Bucket mit dem Namen `gewährtny-bucket`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:ListBucket*",  
        "s3:PutBucket*",  
        "s3:GetBucket*"  
      ],  
      "Resource": [  
        "arn:aws:s3:::my-bucket"  
      ]  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "Policy": {  
    "PolicyName": "my-policy",  
    "PolicyId": "ANPAWGSUGIDPEXAMPLE",  
    "Arn": "arn:aws:iam::12345678012:policy/my-policy",  
    "Path": "/",  
    "DefaultVersionId": "v1",  
    "AttachmentCount": 0,  
    "PermissionsBoundaryUsageCount": 0,  
    "IsAttachable": true,  
    "CreateDate": "2023-05-24T23:16:39+00:00",  
    "UpdateDate": "2023-05-24T23:16:39+00:00",  
    "Tags": [  
      {  
        "Key": "Department",  
        "Value": "Accounting"  
      }  
    ]  
  }  
}
```

```
    },
    "Key": "Location",
    "Value": "Seattle"
  }
]
}
```

Weitere Informationen zu Tagging-Richtlinien finden Sie unter [Tagging von kundenverwalteten Richtlinien](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreatePolicy AWS CLI Befehlsreferenz](#).

create-role

Das folgende Codebeispiel zeigt die Verwendung `create-role`.

AWS CLI

Beispiel 1: Um eine IAM Rolle zu erstellen

Mit dem folgenden `create-role`-Befehl wird eine Rolle mit dem Namen `Test-Role` erstellt und ihr eine Vertrauensrichtlinie angefügt.

```
aws iam create-role \
  --role-name Test-Role \
  --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

Ausgabe:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2013-06-07T20:43:32.821Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
  }
}
```

Die Vertrauensrichtlinie ist als JSON Dokument in der Datei Test-Role-Trust-Policy.json definiert. (Der Dateiname und die Erweiterung sind nicht von Bedeutung.) Die Vertrauensrichtlinie muss einen Prinzipal angeben.

Verwenden Sie den `put-role-policy`-Befehl, um die Berechtigungsrichtlinie der Rolle anzufügen.

[Weitere Informationen finden Sie im Benutzerhandbuch unter Rollen erstellen. IAM AWS IAM](#)

Beispiel 2: Um eine IAM Rolle mit einer bestimmten maximalen Sitzungsdauer zu erstellen

Mit dem folgenden `create-role`-Befehl wird eine Rolle mit dem Namen `Test-Role` erstellt und eine maximale Sitzungsdauer von 7 200 Sekunden (2 Stunden) festgelegt.

```
aws iam create-role \  
  --role-name Test-Role \  
  --assume-role-policy-document file://Test-Role-Trust-Policy.json \  
  --max-session-duration 7200
```

Ausgabe:

```
{  
  "Role": {  
    "Path": "/",  
    "RoleName": "Test-Role",  
    "RoleId": "AKIAIOSFODNN7EXAMPLE",  
    "Arn": "arn:aws:iam::12345678012:role/Test-Role",  
    "CreateDate": "2023-05-24T23:50:25+00:00",  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Sid": "Statement1",  
          "Effect": "Allow",  
          "Principal": {  
            "AWS": "arn:aws:iam::12345678012:root"  
          },  
          "Action": "sts:AssumeRole"  
        }  
      ]  
    }  
  }  
}
```

```
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ändern der maximalen Sitzungsdauer einer Rolle \(AWS API\)](#).

Beispiel 3: So erstellen Sie eine IAM Rolle mit Tags

Der folgende Befehl erstellt eine IAM Rolle Test-Rolle mit Tags. In diesem Beispiel wird der `--tags` Parameter flag mit den folgenden Tags im JSON -Format verwendet:

'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}' Alternativ kann das `--tags`-Flag mit Tags im Kurzformat 'Key=Department,Value=Accounting Key=Location,Value=Seattle' verwendet werden.

```
aws iam create-role \  
  --role-name Test-Role \  
  --assume-role-policy-document file://Test-Role-Trust-Policy.json \  
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
  "Value": "Seattle"}'
```

Ausgabe:

```
{  
  "Role": {  
    "Path": "/",  
    "RoleName": "Test-Role",  
    "RoleId": "AKIAIOSFODNN7EXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:role/Test-Role",  
    "CreateDate": "2023-05-25T23:29:41+00:00",  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Sid": "Statement1",  
          "Effect": "Allow",  
          "Principal": {  
            "AWS": "arn:aws:iam::123456789012:root"  
          },  
          "Action": "sts:AssumeRole"  
        }  
      ],  
    },  
  },  
},
```

```
    "Tags": [
      {
        "Key": "Department",
        "Value": "Accounting"
      },
      {
        "Key": "Location",
        "Value": "Seattle"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [IAM Rollen taggen](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [CreateRole](#) in der AWS CLI Befehlsreferenz.

create-saml-provider

Das folgende Codebeispiel zeigt die Verwendung `create-saml-provider`.

AWS CLI

Um einen SAML-Anbieter zu erstellen

In diesem Beispiel wird ein neuer SAML-Anbieter in IAM mit dem Namen `erstelltMySAMLProvider` erstellt. Es wird durch das SAML-Metadaten-Dokument beschrieben, das sich in der Datei `SAMLMetaData.xml` befindet.

```
aws iam create-saml-provider \
  --saml-metadata-document file://SAMLMetaData.xml \
  --name MySAMLProvider
```

Ausgabe:

```
{
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Creating IAM SAML Identity Providers](#).

- API-Einheiten finden Sie unter [CreateSAMLProvider](#) in der AWS CLI Befehlsreferenz.

create-service-linked-role

Das folgende Codebeispiel zeigt, wie man `create-service-linked-role`.

AWS CLI

So erstellen Sie eine serviceverknüpfte Rolle

Im folgenden `create-service-linked-role` Beispiel wird eine dienstbezogene Rolle für den angegebenen AWS Dienst erstellt und die angegebene Beschreibung angehängt.

```
aws iam create-service-linked-role \  
  --aws-service-name lex.amazonaws.com \  
  --description "My service-linked role to support Lex"
```

Ausgabe:

```
{  
  "Role": {  
    "Path": "/aws-service-role/lex.amazonaws.com/",  
    "RoleName": "AWSServiceRoleForLexBots",  
    "RoleId": "AROAI234567890EXAMPLE",  
    "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/  
AWSServiceRoleForLexBots",  
    "CreateDate": "2019-04-17T20:34:14+00:00",  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": [  
            "sts:AssumeRole"  
          ],  
          "Effect": "Allow",  
          "Principal": {  
            "Service": [  
              "lex.amazonaws.com"  
            ]  
          }  
        }  
      ]  
    }  
  }  
}
```

```

    }
  }
}

```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verwenden von dienstbezogenen Rollen](#).AWS IAM

- APIEinzelheiten finden Sie [CreateServiceLinkedRole](#) in der AWS CLI Befehlsreferenz.

create-service-specific-credential

Das folgende Codebeispiel zeigt die Verwendung `create-service-specific-credential`.

AWS CLI

Erstellen Sie einen Satz dienstspezifischer Anmeldeinformationen für einen Benutzer

Im folgenden `create-service-specific-credential` Beispiel werden ein Benutzername und ein Passwort erstellt, die nur für den Zugriff auf den konfigurierten Dienst verwendet werden können.

```

aws iam create-service-specific-credential \
  --user-name sofia \
  --service-name codecommit.amazonaws.com

```

Ausgabe:

```

{
  "ServiceSpecificCredential": {
    "CreateDate": "2019-04-18T20:45:36+00:00",
    "ServiceName": "codecommit.amazonaws.com",
    "ServiceUserName": "sofia-at-123456789012",
    "ServicePassword": "k1zPZM6uVxMQ3oxqgoY1NuJPyRTZ1vREs76zTQE3eJk=",
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
    "UserName": "sofia",
    "Status": "Active"
  }
}

```

Weitere Informationen finden Sie CodeCommit im AWS CodeCommit Benutzerhandbuch unter [Erstellen von Git-Anmeldeinformationen für HTTPS Verbindungen zu](#).

- API-Einheiten finden Sie [CreateServiceSpecificCredential](#) in der AWS CLI Befehlsreferenz.

create-user

Das folgende Codebeispiel zeigt die Verwendung `create-user`.

AWS CLI

Beispiel 1: Um einen IAM Benutzer zu erstellen

Der folgende `create-user` Befehl erstellt einen IAM Benutzer mit Bob dem Namen des aktuellen Kontos.

```
aws iam create-user \  
  --user-name Bob
```

Ausgabe:

```
{  
  "User": {  
    "UserName": "Bob",  
    "Path": "/",  
    "CreateDate": "2023-06-08T03:20:41.270Z",  
    "UserId": "AIDAIOSFODNN7EXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:user/Bob"  
  }  
}
```

Weitere Informationen finden Sie [im IAM Benutzerhandbuch unter Einen Benutzer in Ihrem AWS Konto erstellen](#).AWS IAM

Beispiel 2: Um einen IAM Benutzer in einem bestimmten Pfad zu erstellen

Der folgende `create-user` Befehl erstellt einen IAM Benutzer mit dem Namen Bob im angegebenen Pfad.

```
aws iam create-user \  
  --user-name Bob \  
  --path /division_abc/subdivision_xyz/
```

Ausgabe:


```
{
  "User": {
    "Path": "/division_abc/subdivision_xyz/",
    "UserName": "Bob",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::12345678012:user/division_abc/subdivision_xyz/Bob",
    "CreateDate": "2023-05-24T18:20:17+00:00"
  }
}
```

Weitere Informationen finden Sie unter [IAMIdentifikatoren](#) im AWS IAMBenutzerhandbuch.

Beispiel 3: So erstellen Sie einen IAM Benutzer mit Tags

Der folgende `create-user` Befehl erstellt einen IAM Benutzer mit einem Namen Bob mit Tags. In diesem Beispiel wird das `--tags` Parameter-Flag mit den folgenden Tags im JSON - Format verwendet: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'` Alternativ kann das `--tags`-Flag mit Tags im Kurzformat `'Key=Department,Value=Accounting Key=Location,Value=Seattle'` verwendet werden.

```
aws iam create-user \
  --user-name Bob \
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
"Value": "Seattle"}'
```

Ausgabe:

```
{
  "User": {
    "Path": "/",
    "UserName": "Bob",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::12345678012:user/Bob",
    "CreateDate": "2023-05-25T17:14:21+00:00",
    "Tags": [
      {
        "Key": "Department",
        "Value": "Accounting"
      },
      {
```

```

        "Key": "Location",
        "Value": "Seattle"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [IAMBenutzer taggen](#) im AWS IAM Benutzerhandbuch.

Beispiel 3: So erstellen Sie einen IAM Benutzer mit einer festgelegten Berechtigungsgrenze

Mit dem folgenden `create-user` Befehl wird ein IAM Benutzer Bob mit der Berechtigungsgrenze von AmazonS3 erstellt. FullAccess

```

aws iam create-user \
  --user-name Bob \
  --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess

```

Ausgabe:

```

{
  "User": {
    "Path": "/",
    "UserName": "Bob",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::12345678012:user/Bob",
    "CreateDate": "2023-05-24T17:50:53+00:00",
    "PermissionsBoundary": {
      "PermissionsBoundaryType": "Policy",
      "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
    }
  }
}

```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).

- APIEinzelheiten finden Sie [CreateUser](#) in der AWS CLI Befehlsreferenz.

create-virtual-mfa-device

Das folgende Codebeispiel zeigt die Verwendung `create-virtual-mfa-device`.

AWS CLI

Um ein virtuelles MFA Gerät zu erstellen

In diesem Beispiel wird ein neues virtuelles MFA Gerät mit dem Namen `BobsMFADevice` erstellt. Es erstellt eine Datei, die Bootstrap-Informationen enthält, `QRCode.png` und platziert sie im `C:/` Verzeichnis. Die in diesem Beispiel verwendete Bootstrap-Methode ist `QRCodePNG`.

```
aws iam create-virtual-mfa-device \
  --virtual-mfa-device-name BobsMFADevice \
  --outfile C:/QRCode.png \
  --bootstrap-method QRCodePNG
```

Ausgabe:

```
{
  "VirtualMFADevice": {
    "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
  }
}
```

Weitere Informationen finden Sie [im AWS IAM Benutzerhandbuch unter Verwenden der Multi-Faktor-Authentifizierung \(MFA\)](#).

- API-Einheiten finden Sie [CreateVirtualMfaDevice](#) in der AWS CLI Befehlsreferenz.

`deactivate-mfa-device`

Das folgende Codebeispiel zeigt die Verwendung `deactivate-mfa-device`.

AWS CLI

Um ein MFA Gerät zu deaktivieren

Dieser Befehl deaktiviert das virtuelle MFA Gerät mit dem ARN `arn:aws:iam::210987654321:mfa/BobsMFADevice`, das dem Benutzer Bob zugeordnet ist.

```
aws iam deactivate-mfa-device \
  --user-name Bob \
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [im AWS IAM Benutzerhandbuch unter Verwenden der Multi-Faktor-Authentifizierung \(MFA\)](#).

- API-Einheiten finden Sie [DeactivateMfaDevice](#) in der AWS CLI Befehlsreferenz.

decode-authorization-message

Das folgende Codebeispiel zeigt die Verwendung `decode-authorization-message`.

AWS CLI

Um eine Meldung über einen Autorisierungsfehler zu dekodieren

Im folgenden `decode-authorization-message` Beispiel wird die Meldung dekodiert, die von der EC2 Konsole zurückgegeben wird, wenn versucht wird, eine Instance ohne die erforderlichen Berechtigungen zu starten.

```
aws sts decode-authorization-message \
  --encoded-message LxzA8VEjEvu-s0TTt3PgYCXik9Yak0qsrfJGRZR98xNcyWAxwRq14xIvd-
  npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-
  mWR9hVYdvX-0zKgVOWF9pwjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpu0IjxDjzf22PF3S0E3XvIQ-
  _PE00aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8LsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX
  Xx9AYAAIr6bhcis7C__bZh4dLAAWooHFGKgfoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

Die Ausgabe ist als einzeilige Textzeichenfolge formatiert, die Sie mit einem beliebigen JSON Textverarbeitungsprogramm analysieren können. JSON

```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements\
  \":{\\"items\\":[]},\\"failures\\":{\\"items\\":[]},\\"context\\":{\\"principal\
  \":{\\"id\\":\\"AIDAV3ZUEFP6J7GY706L0\\",\\"name\\":\\"chain-user\\",\\"arn\\":\
  \\"arn:aws:iam::403299380220:user/chain-user\\"},\\"action\\":\\"ec2:RunInstances\\",\
  \\"resource\\":\\"arn:aws:ec2:us-east-2:403299380220:instance/*\\"},\\"conditions\\":\
  {\\"items\\":[{\\"key\\":\\"ec2:InstanceMarketType\\",\\"values\\":{\\"items\\":[{\\"value\
  \":\\"on-demand\\"}]}},{\\"key\\":\\"aws:Resource\\",\\"values\\":{\\"items\\":[{\\"value\
  \":\\"instance/*\\"}]}},{\\"key\\":\\"aws:Account\\",\\"values\\":{\\"items\\":[{\\"value\
  \":\\"403299380220\\"}]}},{\\"key\\":\\"ec2:AvailabilityZone\\",\\"values\\":{\\"items\\":\
  [{\\"value\\":\\"us-east-2b\\"}]}},{\\"key\\":\\"ec2:ebsoptimized\\",\\"values\\":{\\"items\
  \":[{\\"value\\":\\"false\\"}]}},{\\"key\\":\\"ec2:IsLaunchTemplateResource\\",\\"values\
  \":{\\"items\\":[{\\"value\\":\\"false\\"}]}},{\\"key\\":\\"ec2:InstanceType\\",\\"values\
  \":{\\"items\\":[{\\"value\\":\\"t2.micro\\"}]}},{\\"key\\":\\"ec2:RootDeviceType\\",\
  \\"values\\":{\\"items\\":[{\\"value\\":\\"ebs\\"}]}},{\\"key\\":\\"aws:Region\\",\\"values\
  \":{\\"items\\":[{\\"value\\":\\"us-east-2\\"}]}},{\\"key\\":\\"aws:Service\\",\\"values
```

```

\":{\\"items\":[{\\"value\":"ec2\\"}]}},{\\"key\":"ec2:InstanceID\\",\\"values\":
{\\"items\":[{\\"value\":"*\\"}]}},{\\"key\":"aws:Type\\",\\"values\":{\\"items\":
[{\\"value\":"instance\\"}]}},{\\"key\":"ec2:Tenancy\\",\\"values\":{\\"items\":
[{\\"value\":"default\\"}]}},{\\"key\":"ec2:Region\\",\\"values\":{\\"items\":[{\\"value
\":"us-east-2\\"}]}},{\\"key\":"aws:ARN\\",\\"values\":{\\"items\":[{\\"value\":
\\\"arn:aws:ec2:us-east-2:403299380220:instance/*\\"}]}]}]}"}
}

```

Weitere Informationen finden Sie unter [Wie kann ich eine Meldung mit einem Autorisierungsfehler dekodieren, nachdem ich beim Start einer Instanz einen Fehler "UnauthorizedOperation" erhalten habe? EC2](#) in AWS re:POST.

- API-Einheiten finden Sie [DecodeAuthorizationMessage](#) in der AWS CLI Befehlsreferenz.

delete-access-key

Das folgende Codebeispiel zeigt die Verwendung `delete-access-key`.

AWS CLI

Um einen Zugriffsschlüssel für einen IAM Benutzer zu löschen

Der folgende `delete-access-key` Befehl löscht den angegebenen Zugriffsschlüssel (Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel) für den genannten IAM Bob Benutzer.

```

aws iam delete-access-key \
  --access-key-id AKIDPMS9R04H3FEXAMPLE \
  --user-name Bob

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `list-access-keys` Befehl, um die für einen IAM Benutzer definierten Zugriffsschlüssel aufzulisten.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verwaltung von Zugriffsschlüsseln für IAM AWS IAM Benutzer](#).

- API-Einheiten finden Sie [DeleteAccessKey](#) in der AWS CLI Befehlsreferenz.

delete-account-alias

Das folgende Codebeispiel zeigt die Verwendung `delete-account-alias`.

AWS CLI

So löschen Sie einen Konto-Alias

Mit dem folgenden `delete-account-alias`-Befehl wird der Alias `mycompany` für das aktuelle Konto entfernt.

```
aws iam delete-account-alias \  
  --account-alias mycompany
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ihre AWS Konto-ID und ihr Alias](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteAccountAlias](#) in der AWS CLI Befehlsreferenz.

`delete-account-password-policy`

Das folgende Codebeispiel zeigt die Verwendung `delete-account-password-policy`.

AWS CLI

Um die Passwortrichtlinie für das aktuelle Konto zu löschen

Mit dem folgenden `delete-account-password-policy` Befehl wird die Kennwortrichtlinie für das aktuelle Konto entfernt.

```
aws iam delete-account-password-policy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Einrichten einer Kontokennwortrichtlinie für IAM AWS IAM Benutzer](#).

- APIEinzelheiten finden Sie [DeleteAccountPasswordPolicy](#) in der AWS CLI Befehlsreferenz.

`delete-group-policy`

Das folgende Codebeispiel zeigt die Verwendung `delete-group-policy`.

AWS CLI

Um eine Richtlinie aus einer IAM Gruppe zu löschen

Mit dem folgenden `delete-group-policy`-Befehl wird die Richtlinie mit dem Namen `ExamplePolicy` aus der Gruppe mit dem Namen `Admins` gelöscht.

```
aws iam delete-group-policy \  
  --group-name Admins \  
  --policy-name ExamplePolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `list-group-policies`-Befehl, um die einer Gruppe zugeordneten Richtlinien anzuzeigen.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRichtlinien verwalten](#).

- APIEinzelheiten finden Sie [DeleteGroupPolicy](#) in der AWS CLI Befehlsreferenz.

`delete-group`

Das folgende Codebeispiel zeigt die Verwendung `delete-group`.

AWS CLI

Um eine IAM Gruppe zu löschen

Der folgende `delete-group` Befehl löscht eine IAM Gruppe mit dem Namen `MyTestGroup`.

```
aws iam delete-group \  
  --group-name MyTestGroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Löschen einer AWS IAMBenutzergruppe](#).

- APIEinzelheiten finden Sie [DeleteGroup](#) in der AWS CLI Befehlsreferenz.

`delete-instance-profile`

Das folgende Codebeispiel zeigt die Verwendung `delete-instance-profile`.

AWS CLI

So löschen Sie ein Instance-Profil

Mit dem folgenden `delete-instance-profile`-Befehl wird das Instance-Profil mit dem Namen `ExampleInstanceProfile` gelöscht.

```
aws iam delete-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch [unter Verwenden von Instanzprofilen](#).

- API-Einheiten finden Sie [DeleteInstanceProfile](#) in der AWS CLI Befehlsreferenz.

`delete-login-profile`

Das folgende Codebeispiel zeigt die Verwendung `delete-login-profile`.

AWS CLI

Um ein Passwort für einen IAM Benutzer zu löschen

Der folgende `delete-login-profile` Befehl löscht das Passwort für den genannten IAM Bob Benutzer.

```
aws iam delete-login-profile \  
  --user-name Bob
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Passwörter für IAM AWS IAM Benutzer verwalten](#).

- API-Einheiten finden Sie [DeleteLoginProfile](#) in der AWS CLI Befehlsreferenz.

`delete-open-id-connect-provider`

Das folgende Codebeispiel zeigt die Verwendung `delete-open-id-connect-provider`.

AWS CLI

Um einen IAM OpenID Connect-Identitätsanbieter zu löschen

In diesem Beispiel wird der IAM OIDC Anbieter gelöscht, der eine Verbindung zum Anbieter herstellt. `example.oidcprovider.com`

```
aws iam delete-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
example.oidcprovider.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen von OpenID Connect \(OIDC\) -Identitätsanbietern](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie unter [DeleteOpenIdConnectProvider AWS CLI](#) Befehlsreferenz.

delete-policy-version

Das folgende Codebeispiel zeigt die Verwendung `delete-policy-version`.

AWS CLI

Um eine Version einer verwalteten Richtlinie zu löschen

In diesem Beispiel wird die als identifizierte Version `v2` aus der Richtlinie gelöscht, deren ARN ist `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [DeletePolicyVersion](#) in der AWS CLI Befehlsreferenz.

delete-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-policy`.

AWS CLI

Um eine IAM Richtlinie zu löschen

In diesem Beispiel wird die Richtlinie gelöscht, deren ARN ist `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Richtlinien und Berechtigungen](#).

- API-Einheiten finden Sie [DeletePolicy](#) in der AWS CLI Befehlsreferenz.

delete-role-permissions-boundary

Das folgende Codebeispiel zeigt die Verwendung `delete-role-permissions-boundary`.

AWS CLI

Um eine Berechtigungsgrenze aus einer IAM Rolle zu löschen

Im folgenden `delete-role-permissions-boundary` Beispiel wird die Berechtigungsgrenze für die angegebene IAM Rolle gelöscht. Verwenden Sie den `put-role-permissions-boundary` Befehl, um einer Rolle eine Berechtigungsgrenze zuzuweisen.

```
aws iam delete-role-permissions-boundary \  
  --role-name lambda-application-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Richtlinien und Berechtigungen](#).

- API-Einheiten finden Sie [DeleteRolePermissionsBoundary](#) in der AWS CLI Befehlsreferenz.

delete-role-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-role-policy`.

AWS CLI

Um eine Richtlinie aus einer IAM Rolle zu entfernen

Mit dem folgenden `delete-role-policy`-Befehl wird die Richtlinie mit dem Namen `ExamplePolicy` aus der Rolle mit dem Namen `Test-Role` entfernt.

```
aws iam delete-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ändern einer Rolle](#).

- APIEinzelheiten finden Sie [DeleteRolePolicy](#) in der AWS CLI Befehlsreferenz.

`delete-role`

Das folgende Codebeispiel zeigt die Verwendung `delete-role`.

AWS CLI

Um eine IAM Rolle zu löschen

Mit dem folgenden `delete-role`-Befehl wird die Rolle mit dem Namen `Test-Role` entfernt.

```
aws iam delete-role \  
  --role-name Test-Role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Bevor Sie eine Rolle löschen können, müssen Sie die Rolle aus allen Instance-Profilen entfernen (`remove-role-from-instance-profile`), alle verwalteten Richtlinien entfernen (`detach-role-policy`) und alle eingebundenen Richtlinien, die der Rolle angefügt sind (`delete-role-policy`), löschen.

Weitere Informationen finden Sie unter [IAMRollen erstellen](#) und [Instanzprofile verwenden](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteRole](#) in der AWS CLI Befehlsreferenz.

delete-saml-provider

Das folgende Codebeispiel zeigt die Verwendung `delete-saml-provider`.

AWS CLI

Um einen SAML Anbieter zu löschen

In diesem Beispiel wird der IAM SAML 2.0-Anbieter gelöscht, dessen ARN `arn:aws:iam::123456789012:saml-provider/SAMLADFSPProvider` ist.

```
aws iam delete-saml-provider \  
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSPProvider
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Creating IAM SAML Identity Providers](#).

- API Einzelheiten finden Sie unter [Delete SAML Provider](#) in der AWS CLI Befehlsreferenz.

delete-server-certificate

Das folgende Codebeispiel zeigt, wie man `delete-server-certificate` verwendet.

AWS CLI

Um ein Serverzertifikat aus Ihrem AWS Konto zu löschen

Mit dem folgenden `delete-server-certificate` Befehl wird das angegebene Serverzertifikat aus Ihrem AWS Konto entfernt.

```
aws iam delete-server-certificate \  
--server-certificate-name myUpdatedServerCertificate
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `list-server-certificates` Befehl, um die in Ihrem AWS Konto verfügbaren Serverzertifikate aufzulisten.

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Verwalten von Serverzertifikaten](#).

- API-Einheiten finden Sie [DeleteServerCertificate](#) in der AWS CLI Befehlsreferenz.

delete-service-linked-role

Das folgende Codebeispiel zeigt die Verwendung `delete-service-linked-role`.

AWS CLI

So löschen Sie eine serviceverknüpfte Rolle

Im folgenden `delete-service-linked-role`-Beispiel wird die angegebene serviceverknüpfte Rolle, die Sie nicht mehr benötigen, gelöscht. Der Löschvorgang erfolgt asynchron. Sie können den Status des Löschvorgangs mithilfe des `get-service-linked-role-deletion-status`-Befehls überprüfen und bestätigen, wann der Vorgang abgeschlossen ist.

```
aws iam delete-service-linked-role \  
  --role-name AWSServiceRoleForLexBots
```

Ausgabe:

```
{  
  "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/  
  AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"  
}
```

Weitere Informationen finden Sie im AWS IAM-Benutzerhandbuch unter [Verwenden von serviceverknüpften Rollen](#).

- API-Einheiten finden Sie [DeleteServiceLinkedRole](#) in der AWS CLI Befehlsreferenz.

delete-service-specific-credential

Das folgende Codebeispiel zeigt die Verwendung `delete-service-specific-credential`.

AWS CLI

Beispiel 1: Löschen Sie dienstspezifische Anmeldeinformationen für den anfragenden Benutzer

Im folgenden `delete-service-specific-credential` Beispiel werden die angegebenen dienstspezifischen Anmeldeinformationen für den Benutzer gelöscht, der die Anfrage stellt. Das `service-specific-credential-id` wird bereitgestellt, wenn Sie die Anmeldeinformationen

erstellen, und Sie können sie mithilfe des Befehls abrufen. `list-service-specific-credentials`

```
aws iam delete-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Löschen Sie dienstspezifische Anmeldeinformationen für einen bestimmten Benutzer

Im folgenden `delete-service-specific-credential` Beispiel werden die angegebenen dienstspezifischen Anmeldeinformationen für den angegebenen Benutzer gelöscht. Das `service-specific-credential-id` wird bereitgestellt, wenn Sie die Anmeldeinformationen erstellen, und Sie können sie mithilfe des Befehls abrufen. `list-service-specific-credentials`

```
aws iam delete-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie CodeCommit im AWS CodeCommit Benutzerhandbuch unter [Erstellen von Git-Anmeldeinformationen für HTTPS Verbindungen zu](#).

- API-Einheiten finden Sie [DeleteServiceSpecificCredential](#) in der AWS CLI Befehlsreferenz.

delete-signing-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-signing-certificate`.

AWS CLI

Um ein Signaturzertifikat für einen IAM Benutzer zu löschen

Der folgende `delete-signing-certificate` Befehl löscht das angegebene Signaturzertifikat für den genannten IAM Bob Benutzer.

```
aws iam delete-signing-certificate \  
  --user-name Bob \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `list-signing-certificates` Befehl, um die ID für ein Signaturzertifikat abzurufen.

Weitere Informationen finden Sie unter [Signaturzertifikate verwalten](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteSigningCertificate](#) in der AWS CLI Befehlsreferenz.

delete-ssh-public-key

Das folgende Codebeispiel zeigt die Verwendung `delete-ssh-public-key`.

AWS CLI

Um einen SSH öffentlichen Schlüssel zu löschen, der einem IAM Benutzer zugewiesen ist

Der folgende `delete-ssh-public-key` Befehl löscht den angegebenen SSH öffentlichen Schlüssel, der dem IAM Benutzer `sofia` zugewiesen ist.

```
aws iam delete-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von SSH Schlüsseln und SSH mit CodeCommit](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteSshPublicKey](#) in der AWS CLI Befehlsreferenz.

delete-user-permissions-boundary

Das folgende Codebeispiel zeigt die Verwendung `delete-user-permissions-boundary`.

AWS CLI

Um eine Berechtigungsgrenze für einen IAM Benutzer zu löschen

Im folgenden `delete-user-permissions-boundary` Beispiel wird die dem genannten IAM intern Benutzer zugeordnete Berechtigungsgrenze gelöscht. Verwenden Sie den `put-user-permissions-boundary` Befehl, um einem Benutzer eine Berechtigungsgrenze zuzuweisen.

```
aws iam delete-user-permissions-boundary \  
  --user-name intern
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [DeleteUserPermissionsBoundary](#)in der AWS CLI Befehlsreferenz.

delete-user-policy

Das folgende Codebeispiel zeigt die Verwendungdelete-user-policy.

AWS CLI

Um eine Richtlinie von einem IAM Benutzer zu entfernen

Mit dem folgenden delete-user-policy Befehl wird die angegebene Richtlinie von dem genannten IAM Benutzer entferntBob.

```
aws iam delete-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den list-user-policies Befehl, um eine Liste der Richtlinien für einen IAM Benutzer abzurufen.

Weitere Informationen finden Sie [im IAM Benutzerhandbuch unter Einen Benutzer in Ihrem AWS Konto erstellen](#).AWS IAM

- APIEinzelheiten finden Sie [DeleteUserPolicy](#)in der AWS CLI Befehlsreferenz.

delete-user

Das folgende Codebeispiel zeigt die Verwendungdelete-user.

AWS CLI

Um einen IAM Benutzer zu löschen

Mit dem folgenden `delete-user` Befehl wird der IAM angegebene Benutzer Bob aus dem aktuellen Konto entfernt.

```
aws iam delete-user \  
  --user-name Bob
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Löschen eines AWS IAMBenutzers](#).

- APIEinzelheiten finden Sie [DeleteUser](#) in der AWS CLI Befehlsreferenz.

delete-virtual-mfa-device

Das folgende Codebeispiel zeigt die Verwendung `delete-virtual-mfa-device`.

AWS CLI

Um ein virtuelles MFA Gerät zu entfernen

Mit dem folgenden `delete-virtual-mfa-device` Befehl wird das angegebene MFA Gerät aus dem aktuellen Konto entfernt.

```
aws iam delete-virtual-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Deaktivierung von MFA Geräten](#).

- APIEinzelheiten finden Sie [DeleteVirtualMfaDevice](#) in der AWS CLI Befehlsreferenz.

detach-group-policy

Das folgende Codebeispiel zeigt die Verwendung `detach-group-policy`.

AWS CLI

Um eine Richtlinie von einer Gruppe zu trennen

In diesem Beispiel wird die verwaltete Richtlinie mit dem Namen ARN `arn:aws:iam::123456789012:policy/TesterAccessPolicy` Testers aus der Gruppe entfernt.

```
aws iam detach-group-policy \  
  --group-name Testers \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Benutzergruppen verwalten](#).AWS IAM

- APIEinzelheiten finden Sie [DetachGroupPolicy](#) in der AWS CLI Befehlsreferenz.

detach-role-policy

Das folgende Codebeispiel zeigt die Verwendung `detach-role-policy`.

AWS CLI

So trennen Sie eine Richtlinie von einer Rolle

In diesem Beispiel wird die verwaltete Richtlinie mit dem Namen ARN `arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy` aus der Rolle entfernt `FedTesterRole`.

```
aws iam detach-role-policy \  
  --role-name FedTesterRole \  
  --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ändern einer Rolle](#).

- APIEinzelheiten finden Sie [DetachRolePolicy](#) in der AWS CLI Befehlsreferenz.

detach-user-policy

Das folgende Codebeispiel zeigt die Verwendung `detach-user-policy`.

AWS CLI

So trennen Sie eine Richtlinie von einem Benutzer

In diesem Beispiel wird die verwaltete Richtlinie mit dem ARN `arn:aws:iam::123456789012:policy/TesterPolicy` vom Benutzer entfernt Bob.

```
aws iam detach-user-policy \  
  --user-name Bob \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [IAM Benutzerhandbuch unter Ändern der Berechtigungen für einen AWS IAM Benutzer](#).

- API Einzelheiten finden Sie [DetachUserPolicy](#) in der AWS CLI Befehlsreferenz.

enable-mfa-device

Das folgende Codebeispiel zeigt die Verwendung `enable-mfa-device`.

AWS CLI

Um ein MFA Gerät zu aktivieren

Nachdem Sie mit dem `create-virtual-mfa-device` Befehl ein neues virtuelles MFA Gerät erstellt haben, können Sie das MFA Gerät einem Benutzer zuweisen. Im folgenden `enable-mfa-device` Beispiel wird dem Benutzer MFA Bob das Gerät mit der Seriennummer `arn:aws:iam::210987654321:mfa/BobsMFADevice` zugewiesen. Der Befehl synchronisiert auch das Gerät mit AWS indem die ersten beiden Codes nacheinander aus dem virtuellen MFA Gerät eingefügt werden.

```
aws iam enable-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 789012
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Aktivieren eines Geräts mit virtueller Multifaktor-Authentifizierung \(MFA\)](#).

- APIEinzelheiten finden Sie [EnableMfaDevice](#) in der AWS CLI Befehlsreferenz.

generate-credential-report

Das folgende Codebeispiel zeigt die Verwendung `generate-credential-report`.

AWS CLI

So erstellen Sie einen Bericht zu Anmeldeinformationen

Im folgenden Beispiel wird versucht, einen Anmeldeinformationsbericht für das AWS Konto zu generieren.

```
aws iam generate-credential-report
```

Ausgabe:

```
{
  "State": "STARTED",
  "Description": "No report exists. Starting a new report generation task"
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Berichte über Anmeldeinformationen für Ihr AWS Konto](#) abrufen.

- APIEinzelheiten finden Sie unter [GenerateCredentialReport AWS CLI](#) Befehlsreferenz.

generate-organizations-access-report

Das folgende Codebeispiel zeigt die Verwendung `generate-organizations-access-report`.

AWS CLI

Beispiel 1: Um einen Zugriffsbericht für einen Stamm in einer Organisation zu generieren

Im folgenden `generate-organizations-access-report` Beispiel wird ein Hintergrundjob gestartet, um einen Zugriffsbericht für den angegebenen Stamm in einer Organisation zu erstellen. Sie können den Bericht nach seiner Erstellung anzeigen, indem Sie den `get-organizations-access-report` Befehl ausführen.

```
aws iam generate-organizations-access-report \  
  --entity-path o-4fxmplt198/r-c3xb
```

Ausgabe:

```
{  
  "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"  
}
```

Beispiel 2: Um einen Zugriffsbericht für ein Konto in einer Organisation zu generieren

Im folgenden `generate-organizations-access-report` Beispiel wird ein Hintergrundjob gestartet, um einen Zugriffsbericht für die Konto-ID 123456789012 in der Organisation zu erstellen `o-4fxmplt198`. Sie können den Bericht nach seiner Erstellung anzeigen, indem Sie den `get-organizations-access-report` Befehl ausführen.

```
aws iam generate-organizations-access-report \  
  --entity-path o-4fxmplt198/r-c3xb/123456789012
```

Ausgabe:

```
{  
  "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"  
}
```

Beispiel 3: Um einen Zugriffsbericht für ein Konto in einer Organisationseinheit in einer Organisation zu generieren

Im folgenden `generate-organizations-access-report` Beispiel wird ein Hintergrundjob gestartet, um einen Zugriffsbericht für die Konto-ID 234567890123 in der Organisationseinheit `ou-c3xb-lmu7j2yg` der Organisation zu erstellen `o-4fxmplt198`. Sie können den Bericht nach seiner Erstellung anzeigen, indem Sie den `get-organizations-access-report` Befehl ausführen.

```
aws iam generate-organizations-access-report \  
  --entity-path o-4fxmplt198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

Ausgabe:

```
{
```

```
"JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

Verwenden Sie die `organizations list-organizational-units-for-parent` Befehle und, um Details zu Stammverzeichnissen `organizations list-roots` und Organisationseinheiten in Ihrer Organisation abzurufen.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Verfeinerung von Berechtigungen bei der AWS Verwendung von Informationen, auf die zuletzt zugegriffen wurde](#).

- API Einheiten finden Sie [GenerateOrganizationsAccessReport](#) in der AWS CLI Befehlsreferenz.

generate-service-last-accessed-details

Das folgende Codebeispiel zeigt die Verwendung `generate-service-last-accessed-details`.

AWS CLI

Beispiel 1: Um einen Servicezugriffsbericht für eine benutzerdefinierte Richtlinie zu generieren

Im folgenden `generate-service-last-accessed-details` Beispiel wird ein Hintergrundjob gestartet, um einen Bericht zu generieren, der die Dienste auflistet, auf die IAM Benutzer und andere Entitäten zugreifen, mit einer benutzerdefinierten Richtlinie `namensintern-boundary`. Sie können den Bericht nach seiner Erstellung anzeigen, indem Sie den `get-service-last-accessed-details` Befehl ausführen.

```
aws iam generate-service-last-accessed-details \
  --arn arn:aws:iam::123456789012:policy/intern-boundary
```

Ausgabe:

```
{
  "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
}
```

Beispiel 2: Um einen Servicezugriffsbericht für die AWS verwaltete `AdministratorAccess` Richtlinie zu generieren

Im folgenden `generate-service-last-accessed-details` Beispiel wird ein Hintergrundjob gestartet, um einen Bericht zu generieren, der die Dienste auflistet, auf die IAM Benutzer und

andere Entitäten mit der AWS verwalteten AdministratorAccess Richtlinie zugreifen. Sie können den Bericht nach seiner Erstellung anzeigen, indem Sie den `get-service-last-accessed-details` Befehl ausführen.

```
aws iam generate-service-last-accessed-details \  
  --arn arn:aws:iam::aws:policy/AdministratorAccess
```

Ausgabe:

```
{  
  "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"  
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Verfeinerung von Berechtigungen bei der AWS Verwendung von Informationen, auf die zuletzt zugegriffen wurde](#).

- APIEinzelheiten finden Sie [GenerateServiceLastAccessedDetails](#) in der AWS CLI Befehlsreferenz.

get-access-key-last-used

Das folgende Codebeispiel zeigt die Verwendung `get-access-key-last-used`.

AWS CLI

So rufen Sie Informationen darüber ab, wann der angegebene Zugriffsschlüssel zuletzt verwendet wurde

Im folgenden Beispiel werden Informationen darüber abgerufen, wann der Zugriffsschlüssel ABCDEXAMPLE zuletzt verwendet wurde.

```
aws iam get-access-key-last-used \  
  --access-key-id ABCDEXAMPLE
```

Ausgabe:

```
{  
  "UserName": "Bob",  
  "AccessKeyLastUsed": {  
    "Region": "us-east-1",  
    "ServiceName": "iam",
```

```
    "LastUsedDate": "2015-06-16T22:45:00Z"
  }
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verwaltung von Zugriffsschlüsseln für IAM AWS IAM Benutzer](#).

- API-Einheiten finden Sie [GetAccessKeyLastUsed](#) in der AWS CLI Befehlsreferenz.

get-account-authorization-details

Das folgende Codebeispiel zeigt die Verwendung `get-account-authorization-details`.

AWS CLI

Um IAM Benutzer, Gruppen, Rollen und Richtlinien eines AWS Kontos aufzulisten

Der folgende `get-account-authorization-details` Befehl gibt Informationen zu allen IAM Benutzern, Gruppen, Rollen und Richtlinien im AWS Konto zurück.

```
aws iam get-account-authorization-details
```

Ausgabe:

```
{
  "RoleDetailList": [
    {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "RoleId": "AROA1234567890EXAMPLE",
      "CreateDate": "2014-07-30T17:09:20Z",
      "InstanceProfileList": [
```



```
{
  "InstanceProfileId": "AIPA1234567890EXAMPLE",
  "Roles": [
    {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "RoleId": "AROA1234567890EXAMPLE",
      "CreateDate": "2014-07-30T17:09:20Z",
      "RoleName": "EC2role",
      "Path": "/",
      "Arn": "arn:aws:iam::123456789012:role/EC2role"
    }
  ],
  "CreateDate": "2014-07-30T17:09:20Z",
  "InstanceProfileName": "EC2role",
  "Path": "/",
  "Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
},
"RoleName": "EC2role",
"Path": "/",
"AttachedManagedPolicies": [
  {
    "PolicyName": "AmazonS3FullAccess",
    "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
  },
  {
    "PolicyName": "AmazonDynamoDBFullAccess",
    "PolicyArn": "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess"
  }
],
"RoleLastUsed": {
  "Region": "us-west-2",
```

```
        "LastUsedDate": "2019-11-13T17:30:00Z"
      },
      "RolePolicyList": [],
      "Arn": "arn:aws:iam::123456789012:role/EC2role"
    }
  ],
  "GroupDetailList": [
    {
      "GroupId": "AIDA1234567890EXAMPLE",
      "AttachedManagedPolicies": {
        "PolicyName": "AdministratorAccess",
        "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
      },
      "GroupName": "Admins",
      "Path": "/",
      "Arn": "arn:aws:iam::123456789012:group/Admins",
      "CreateDate": "2013-10-14T18:32:24Z",
      "GroupPolicyList": []
    },
    {
      "GroupId": "AIDA1234567890EXAMPLE",
      "AttachedManagedPolicies": {
        "PolicyName": "PowerUserAccess",
        "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
      },
      "GroupName": "Dev",
      "Path": "/",
      "Arn": "arn:aws:iam::123456789012:group/Dev",
      "CreateDate": "2013-10-14T18:33:55Z",
      "GroupPolicyList": []
    },
    {
      "GroupId": "AIDA1234567890EXAMPLE",
      "AttachedManagedPolicies": [],
      "GroupName": "Finance",
      "Path": "/",
      "Arn": "arn:aws:iam::123456789012:group/Finance",
      "CreateDate": "2013-10-14T18:57:48Z",
      "GroupPolicyList": [
        {
          "PolicyName": "policygen-201310141157",
          "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
```

```

        {
            "Action": "aws-portal:*",
            "Sid": "Stmt1381777017000",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
],
"UserDetailList": [
    {
        "UserName": "Alice",
        "GroupList": [
            "Admins"
        ],
        "CreateDate": "2013-10-14T18:32:24Z",
        "UserId": "AIDA1234567890EXAMPLE",
        "UserPolicyList": [],
        "Path": "/",
        "AttachedManagedPolicies": [],
        "Arn": "arn:aws:iam::123456789012:user/Alice"
    },
    {
        "UserName": "Bob",
        "GroupList": [
            "Admins"
        ],
        "CreateDate": "2013-10-14T18:32:25Z",
        "UserId": "AIDA1234567890EXAMPLE",
        "UserPolicyList": [
            {
                "PolicyName": "DenyBillingAndIAMPolicy",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": {
                        "Effect": "Deny",
                        "Action": [
                            "aws-portal:*",
                            "iam:*"
                        ],
                    },
                },
                "Resource": "*"
            }
        ]
    }
]

```

```

        }
      }
    }
  ],
  "Path": "/",
  "AttachedManagedPolicies": [],
  "Arn": "arn:aws:iam::123456789012:user/Bob"
},
{
  "UserName": "Charlie",
  "GroupList": [
    "Dev"
  ],
  "CreateDate": "2013-10-14T18:33:56Z",
  "UserId": "AIDA1234567890EXAMPLE",
  "UserPolicyList": [],
  "Path": "/",
  "AttachedManagedPolicies": [],
  "Arn": "arn:aws:iam::123456789012:user/Charlie"
}
],
"Policies": [
  {
    "PolicyName": "create-update-delete-set-managed-policies",
    "CreateDate": "2015-02-06T19:58:34Z",
    "AttachmentCount": 1,
    "IsAttachable": true,
    "PolicyId": "ANPA1234567890EXAMPLE",
    "DefaultVersionId": "v1",
    "PolicyVersionList": [
      {
        "CreateDate": "2015-02-06T19:58:34Z",
        "VersionId": "v1",
        "Document": {
          "Version": "2012-10-17",
          "Statement": {
            "Effect": "Allow",
            "Action": [
              "iam:CreatePolicy",
              "iam:CreatePolicyVersion",
              "iam>DeletePolicy",
              "iam>DeletePolicyVersion",
              "iam:GetPolicy",
              "iam:GetPolicyVersion",
            ]
          }
        }
      }
    ]
  }
]

```

```

        "iam:ListPolicies",
        "iam:ListPolicyVersions",
        "iam:SetDefaultPolicyVersion"
    ],
    "Resource": "*"
}
},
"IsDefaultVersion": true
}
],
"Path": "/",
"Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
"UpdateDate": "2015-02-06T19:58:34Z"
},
{
    "PolicyName": "S3-read-only-specific-bucket",
    "CreateDate": "2015-01-21T21:39:41Z",
    "AttachmentCount": 1,
    "IsAttachable": true,
    "PolicyId": "ANPA1234567890EXAMPLE",
    "DefaultVersionId": "v1",
    "PolicyVersionList": [
        {
            "CreateDate": "2015-01-21T21:39:41Z",
            "VersionId": "v1",
            "Document": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "s3:Get*",
                            "s3:List*"
                        ],
                        "Resource": [
                            "arn:aws:s3:::example-bucket",
                            "arn:aws:s3:::example-bucket/*"
                        ]
                    }
                ]
            }
        }
    ],
    "IsDefaultVersion": true
}
}

```

```
    ],
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-bucket",
    "UpdateDate": "2015-01-21T23:39:41Z"
  },
  {
    "PolicyName": "AmazonEC2FullAccess",
    "CreateDate": "2015-02-06T18:40:15Z",
    "AttachmentCount": 1,
    "IsAttachable": true,
    "PolicyId": "ANPA1234567890EXAMPLE",
    "DefaultVersionId": "v1",
    "PolicyVersionList": [
      {
        "CreateDate": "2014-10-30T20:59:46Z",
        "VersionId": "v1",
        "Document": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "ec2:*",
              "Effect": "Allow",
              "Resource": "*"
            },
            {
              "Effect": "Allow",
              "Action": "elasticloadbalancing:*",
              "Resource": "*"
            },
            {
              "Effect": "Allow",
              "Action": "cloudwatch:*",
              "Resource": "*"
            },
            {
              "Effect": "Allow",
              "Action": "autoscaling:*",
              "Resource": "*"
            }
          ]
        },
        "IsDefaultVersion": true
      }
    ]
  }
],
```

```
        "Path": "/",
        "Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
        "UpdateDate": "2015-02-06T18:40:15Z"
    }
],
"Marker": "EXAMPLEkakov9BCuUNFDtxWSyfetYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbp125FDWEXAMPLE",
"IsTruncated": true
}
```

Weitere Informationen finden Sie in den [Richtlinien für AWS Sicherheitsaudits](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [GetAccountAuthorizationDetails](#) in der AWS CLI Befehlsreferenz.

get-account-password-policy

Das folgende Codebeispiel zeigt die Verwendung `get-account-password-policy`.

AWS CLI

So zeigen Sie die Passwortrichtlinie für das aktuelle Konto an

Mit dem folgenden `get-account-password-policy`-Befehl werden Details zur Passwortrichtlinie für das aktuelle Konto angezeigt.

```
aws iam get-account-password-policy
```

Ausgabe:

```
{
  "PasswordPolicy": {
    "AllowUsersToChangePassword": false,
    "RequireLowercaseCharacters": false,
    "RequireUppercaseCharacters": false,
    "MinimumPasswordLength": 8,
    "RequireNumbers": true,
    "RequireSymbols": true
  }
}
```

Wenn keine Passwortrichtlinie für das Konto definiert ist, gibt der Befehl einen `NoSuchEntity`-Fehler zurück.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Einrichten einer Kontokennwortrichtlinie für IAM AWS IAM Benutzer](#).

- API-Einheiten finden Sie [GetAccountPasswordPolicy](#) in der AWS CLI Befehlsreferenz.

get-account-summary

Das folgende Codebeispiel zeigt die Verwendung `get-account-summary`.

AWS CLI

Um Informationen über die Nutzung von IAM Entitäten und die IAM Kontingente im aktuellen Konto zu erhalten

Der folgende `get-account-summary` Befehl gibt Informationen zur aktuellen IAM Entitätsnutzung und zu den aktuellen IAM Entitätskontingenten im Konto zurück.

```
aws iam get-account-summary
```

Ausgabe:

```
{
  "SummaryMap": {
    "UsersQuota": 5000,
    "GroupsQuota": 100,
    "InstanceProfiles": 6,
    "SigningCertificatesPerUserQuota": 2,
    "AccountAccessKeysPresent": 0,
    "RolesQuota": 250,
    "RolePolicySizeQuota": 10240,
    "AccountSigningCertificatesPresent": 0,
    "Users": 27,
    "ServerCertificatesQuota": 20,
    "ServerCertificates": 0,
    "AssumeRolePolicySizeQuota": 2048,
    "Groups": 7,
    "MFADevicesInUse": 1,
    "Roles": 3,
    "AccountMFAEnabled": 1,
    "MFADevices": 3,
```



```

    "GroupsPerUserQuota": 10,
    "GroupPolicySizeQuota": 5120,
    "InstanceProfilesQuota": 100,
    "AccessKeysPerUserQuota": 2,
    "Providers": 0,
    "UserPolicySizeQuota": 2048
  }
}

```

Weitere Informationen zu Entitätsbeschränkungen finden Sie unter [IAM und unter AWS STS Kontingente](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [GetAccountSummary](#) in der AWS CLI Befehlsreferenz.

get-context-keys-for-custom-policy

Das folgende Codebeispiel zeigt die Verwendung `get-context-keys-for-custom-policy`.

AWS CLI

Beispiel 1: Um die Kontextschlüssel aufzulisten, auf die von einer oder mehreren benutzerdefinierten JSON Richtlinien verwiesen wird, die als Parameter in der Befehlszeile bereitgestellt werden

Der folgende `get-context-keys-for-custom-policy` Befehl analysiert jede angegebene Richtlinie und listet die von diesen Richtlinien verwendeten Kontextschlüssel auf. Verwenden Sie diesen Befehl, um zu ermitteln, welche Kontextschlüsselwerte Sie angeben müssen, um die Richtlinien-Simulatorbefehle `simulate-custom-policy` und `simulate-custom-policy` erfolgreich verwenden zu können. Mit dem `get-context-keys-for-custom-policy` Befehl können Sie auch die Liste der Kontextschlüssel abrufen, die von allen Richtlinien verwendet werden, die einem IAM Benutzer oder einer Rolle zugeordnet sind. Parameterwerte, die mit `file://` beginnen, weisen den Befehl an, die Datei zu lesen und den Inhalt anstelle des Dateinamens selbst als Wert für den Parameter zu verwenden.

```

aws iam get-context-keys-for-custom-policy \
  --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/${aws:username}","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}}'

```

Ausgabe:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

Beispiel 2: Um die Kontextschlüssel aufzulisten, auf die in einer oder mehreren benutzerdefinierten JSON Richtlinien verwiesen wird, die als Dateieingabe bereitgestellt werden

Der folgende `get-context-keys-for-custom-policy` Befehl entspricht dem vorherigen Beispiel, außer dass die Richtlinien in einer Datei und nicht als Parameter bereitgestellt werden. Da der Befehl eine JSON Liste von Zeichenfolgen und keine Liste von JSON Strukturen erwartet, muss die Datei wie folgt strukturiert sein, obwohl Sie sie zu einer einzigen zusammenfassen können.

```
[
  "Policy1",
  "Policy2"
]
```

Eine Datei, die die Richtlinie aus dem vorherigen Beispiel enthält, muss also wie folgt aussehen. Sie müssen jedem eingebetteten doppelten Anführungszeichen in der Richtlinienzeichenfolge einen umgekehrten Schrägstrich voranstellen.

```
[ "{\"Version\": \"2012-10-17\", \"Statement\": {\"Effect\": \"Allow\", \"Action\": \"dynamodb:*\", \"Resource\": \"arn:aws:dynamodb:us-west-2:128716708097:table/${aws:username}\", \"Condition\": {\"DateGreaterThan\": {\"aws:CurrentTime\": \"2015-08-16T12:00:00Z\"}}}}" ]
```

Diese Datei kann dann an den folgenden Befehl gesendet werden.

```
aws iam get-context-keys-for-custom-policy \
  --policy-input-list file://policyfile.json
```

Ausgabe:

```
{
  "ContextKeyNames": [
    "aws:username",
```

```
    "aws:CurrentTime"  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwenden des IAM Richtlinien-simulators \(AWS CLI und AWS API\)](#) im AWS IAM Benutzerhandbuch.

- API Einzelheiten finden Sie [GetContextKeysForCustomPolicy](#) in der AWS CLI Befehlsreferenz.

get-context-keys-for-principal-policy

Das folgende Codebeispiel zeigt die Verwendung `get-context-keys-for-principal-policy`.

AWS CLI

Um die Kontextschlüssel aufzulisten, auf die von allen Richtlinien verwiesen wird, die einem IAM Prinzipal zugeordnet sind

Mit dem folgenden `get-context-keys-for-principal-policy` Befehl werden alle Richtlinien abgerufen, die dem Benutzer `saanvi` und allen Gruppen, denen er angehört, zugeordnet sind. Anschließend analysiert er die einzelnen Richtlinien und listet die von diesen Richtlinien verwendeten Kontextschlüssel auf. Verwenden Sie diesen Befehl, um zu ermitteln, welche Kontextschlüsselwerte Sie angeben müssen, um die `simulate-principal-policy` Befehle `simulate-custom-policy` und erfolgreich verwenden zu können. Mit dem `get-context-keys-for-custom-policy` Befehl können Sie auch die Liste der Kontextschlüssel abrufen, die von einer beliebigen JSON Richtlinie verwendet werden.

```
aws iam get-context-keys-for-principal-policy \  
--policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

Ausgabe:

```
{  
  "ContextKeyNames": [  
    "aws:username",  
    "aws:CurrentTime"  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwenden des IAM Richtlinien-simulators \(AWS CLI und AWS API\)](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [GetContextKeysForPrincipalPolicy](#) in der AWS CLI Befehlsreferenz.

get-credential-report

Das folgende Codebeispiel zeigt die Verwendung `get-credential-report`.

AWS CLI

So rufen Sie einen Bericht zu Anmeldeinformationen ab

In diesem Beispiel wird der zurückgegebene Bericht geöffnet und als Array von Textzeilen an die Pipeline ausgegeben.

```
aws iam get-credential-report
```

Ausgabe:

```
{
  "GeneratedTime": "2015-06-17T19:11:50Z",
  "ReportFormat": "text/csv"
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Berichte über Anmeldeinformationen für Ihr AWS Konto abrufen](#).

- API-Einheiten finden Sie unter [GetCredentialReport AWS CLI Befehlsreferenz](#).

get-group-policy

Das folgende Codebeispiel zeigt die Verwendung `get-group-policy`.

AWS CLI

Um Informationen über eine Richtlinie abzurufen, die einer IAM Gruppe zugeordnet ist

Mit dem folgenden `get-group-policy` Befehl werden Informationen über die angegebene Richtlinie abgerufen, die der genannten Gruppe zugeordnet ist `Test-Group`.

```
aws iam get-group-policy \
  --group-name Test-Group \
  --policy-name S3-ReadOnly-Policy
```

Ausgabe:

```
{
  "GroupName": "Test-Group",
  "PolicyDocument": {
    "Statement": [
      {
        "Action": [
          "s3:Get*",
          "s3:List*"
        ],
        "Resource": "*",
        "Effect": "Allow"
      }
    ]
  },
  "PolicyName": "S3-ReadOnly-Policy"
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRichtlinien verwalten](#).

- APIEinzelheiten finden Sie [GetGroupPolicy](#) in der AWS CLI Befehlsreferenz.

get-group

Das folgende Codebeispiel zeigt die Verwendung `get-group`.

AWS CLI

Um eine IAM Gruppe zu bekommen

In diesem Beispiel werden Details zur IAM Gruppe zurückgegeben `Admins`.

```
aws iam get-group \
  --group-name Admins
```

Ausgabe:

```
{
  "Group": {
    "Path": "/",
    "CreateDate": "2015-06-16T19:41:48Z",
    "GroupId": "AIDGPM9R04H3FEXAMPLE",
```

```

    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "GroupName": "Admins"
  },
  "Users": []
}

```

Weitere Informationen finden Sie unter [IAM-Identitäten \(Benutzer, Benutzergruppen und Rollen\)](#) im AWS IAM-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetGroup AWS CLI-Befehlsreferenz](#).

get-instance-profile

Das folgende Codebeispiel zeigt die Verwendung von `get-instance-profile`.

AWS CLI

Um Informationen über ein Instanzprofil abzurufen

Der folgende `get-instance-profile` Befehl ruft Informationen über das angegebene Instanzprofil `ExampleInstanceProfile` ab.

```

aws iam get-instance-profile \
  --instance-profile-name ExampleInstanceProfile

```

Ausgabe:

```

{
  "InstanceProfile": {
    "InstanceId": "AID2MAB8DPLSRHEXAMPLE",
    "Roles": [
      {
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
        "RoleId": "AIDGPM9R04H3FEXAMPLE",
        "CreateDate": "2013-01-09T06:33:26Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "Arn": "arn:aws:iam::336924118301:role/Test-Role"
      }
    ],
    "CreateDate": "2013-06-12T23:52:02Z",
    "InstanceProfileName": "ExampleInstanceProfile",
  }
}

```

```
    "Path": "/",
    "Arn": "arn:aws:iam::336924118301:instance-profile/ExampleInstanceProfile"
  }
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch [unter Verwenden von Instanzprofilen](#).

- API Einheiten finden Sie [GetInstanceProfile](#) in der AWS CLI Befehlsreferenz.

get-login-profile

Das folgende Codebeispiel zeigt die Verwendung `get-login-profile`.

AWS CLI

Um Passwortinformationen für einen IAM Benutzer abzurufen

Der folgende `get-login-profile` Befehl ruft Informationen über das Passwort für den genannten IAM Benutzer ab `Bob`.

```
aws iam get-login-profile \
  --user-name Bob
```

Ausgabe:

```
{
  "LoginProfile": {
    "UserName": "Bob",
    "CreateDate": "2012-09-21T23:03:39Z"
  }
}
```

Der `get-login-profile` Befehl kann verwendet werden, um zu überprüfen, ob ein IAM Benutzer ein Passwort hat. Der Befehl gibt einen `NoSuchEntity` Fehler zurück, wenn kein Passwort für den Benutzer definiert ist.

Mit diesem Befehl können Sie kein Passwort anzeigen. Wenn das Passwort verloren gegangen ist, können Sie das Passwort (`update-login-profile`) für den Benutzer zurücksetzen. Alternativ können Sie das Anmeldeprofil (`delete-login-profile`) für den Benutzer löschen und dann ein neues erstellen (`create-login-profile`).

Weitere Informationen finden Sie im Benutzerhandbuch unter [Passwörter für IAM AWS IAM Benutzer `verwalten`](#).

- API-Einheiten finden Sie [GetLoginProfile](#) in der AWS CLI Befehlsreferenz.

`get-mfa-device`

Das folgende Codebeispiel zeigt die Verwendung `get-mfa-device`.

AWS CLI

Um Informationen über einen FIDO Sicherheitsschlüssel abzurufen

Das folgende `get-mfa-device` Befehlsbeispiel ruft Informationen über den angegebenen FIDO Sicherheitsschlüssel ab.

```
aws iam get-mfa-device \
  --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-EXAMPLEBN5FHTECLFG7EXAMPLE
```

Ausgabe:

```
{
  "UserName": "alice",
  "SerialNumber": "arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-EXAMPLEBN5FHTECLFG7EXAMPLE",
  "EnableDate": "2023-09-19T01:49:18+00:00",
  "Certifications": {
    "FIDO": "L1"
  }
}
```

Weitere Informationen finden Sie [im AWS IAM Benutzerhandbuch unter Verwenden der Multi-Faktor-Authentifizierung \(MFA\)](#).

- API-Einheiten finden Sie [GetMfaDevice](#) in der AWS CLI Befehlsreferenz.

`get-open-id-connect-provider`

Das folgende Codebeispiel zeigt die Verwendung `get-open-id-connect-provider`.


```
--job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

Ausgabe:

```
{
  "JobStatus": "COMPLETED",
  "JobCreationDate": "2019-09-30T06:53:36.187Z",
  "JobCompletionDate": "2019-09-30T06:53:37.547Z",
  "NumberOfServicesAccessible": 188,
  "NumberOfServicesNotAccessed": 171,
  "AccessDetails": [
    {
      "ServiceName": "Alexa for Business",
      "ServiceNamespace": "a4b",
      "TotalAuthenticatedEntities": 0
    },
    ...
  ]
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Verfeinerung von Berechtigungen bei der AWS Verwendung von Informationen, auf die zuletzt zugegriffen wurde](#).

- APIEinzelheiten finden Sie [GetOrganizationsAccessReport](#) in der AWS CLI Befehlsreferenz.

get-policy-version

Das folgende Codebeispiel zeigt die Verwendung `get-policy-version`.

AWS CLI

So rufen Sie Informationen über die angegebene Version der angegebenen verwalteten Richtlinie ab

In diesem Beispiel wird das Richtliniendokument für die Version 2 der Richtlinie zurückgegeben, deren ARN ist `arn:aws:iam::123456789012:policy/MyManagedPolicy`.

```
aws iam get-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Ausgabe:

```
{
  "PolicyVersion": {
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": "iam:*",
          "Resource": "*"
        }
      ]
    },
    "VersionId": "v2",
    "IsDefaultVersion": true,
    "CreateDate": "2023-04-11T00:22:54+00:00"
  }
}
```

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [GetPolicyVersion](#) in der AWS CLI Befehlsreferenz.

get-policy

Das folgende Codebeispiel zeigt die Verwendung `get-policy`.

AWS CLI

So rufen Sie Informationen über die angegebene verwaltete Richtlinie ab

In diesem Beispiel werden Details zu der verwalteten Richtlinie zurückgegeben, deren ARN ist `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Ausgabe:

```
{
  "Policy": {
```

```
    "PolicyName": "MySamplePolicy",
    "CreateDate": "2015-06-17T19:23:32Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "Z27SI6FQMGNO2EXAMPLE1",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
    "UpdateDate": "2015-06-17T19:23:32Z"
  }
}
```

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [GetPolicy](#) in der AWS CLI Befehlsreferenz.

get-role-policy

Das folgende Codebeispiel zeigt die Verwendung `get-role-policy`.

AWS CLI

Um Informationen über eine Richtlinie abzurufen, die einer IAM Rolle zugeordnet ist

Der folgende `get-role-policy` Befehl ruft Informationen über die angegebene Richtlinie ab, die der genannten Rolle zugeordnet ist `Test-Rolle`.

```
aws iam get-role-policy \
  --role-name Test-Role \
  --policy-name ExamplePolicy
```

Ausgabe:

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
    "Statement": [
      {
        "Action": [
          "s3:ListBucket",
          "s3:Put*",

```

```

        "s3:Get*",
        "s3:*MultipartUpload*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "1"
  }
]
}
"PolicyName": "ExamplePolicy"
}

```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRollen erstellen](#).

- APIEinzelheiten finden Sie [GetRolePolicy](#) in der AWS CLI Befehlsreferenz.

get-role

Das folgende Codebeispiel zeigt die Verwendung `get-role`.

AWS CLI

Um Informationen über eine IAM Rolle zu erhalten

Mit dem folgenden `get-role`-Befehl werden Informationen über die Rolle mit dem Namen `Test-Role` abgerufen.

```

aws iam get-role \
  --role-name Test-Role

```

Ausgabe:

```

{
  "Role": {
    "Description": "Test Role",
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "MaxSessionDuration": 3600,
    "RoleId": "AROA1234567890EXAMPLE",
    "CreateDate": "2019-11-13T16:45:56Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "RoleLastUsed": {

```

```
    "Region": "us-east-1",
    "LastUsedDate": "2019-11-13T17:14:00Z"
  },
  "Arn": "arn:aws:iam::123456789012:role/Test-Role"
}
}
```

Der Befehl zeigt die Vertrauensrichtlinie an, die der Rolle zugeordnet ist. Verwenden Sie den `list-role-policies`-Befehl, um die einer Rolle zugeordneten Berechtigungsrichtlinien aufzulisten.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRollen erstellen](#).

- APIEinzelheiten finden Sie [GetRole](#) in der AWS CLI Befehlsreferenz.

get-saml-provider

Das folgende Codebeispiel zeigt die Verwendung `get-saml-provider`.

AWS CLI

Um das SAML Provider-Metadokument abzurufen

In diesem Beispiel werden die Details über den SAML 2.0-Anbieter abgerufen, dessen ARM `arn:aws:iam::123456789012:saml-provider/SAMLADFS` Die Antwort enthält das Metadatendokument, das Sie vom Identitätsanbieter zur Erstellung der AWS SAML Anbieterentität erhalten haben, sowie die Erstellungs- und Ablaufdaten.

```
aws iam get-saml-provider \
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Ausgabe:

```
{
  "SAMLMetadataDocument": "...SAMLMetadataDocument-XML...",
  "CreateDate": "2017-03-06T22:29:46+00:00",
  "ValidUntil": "2117-03-06T22:29:46.433000+00:00",
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    }
  ]
}
```

```
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Creating IAM SAML Identity Providers](#).

- API Einzelheiten finden Sie [GetSamlProvider](#) unter AWS CLI Befehlsreferenz.

get-server-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-server-certificate`.

AWS CLI

Um Details zu einem Serverzertifikat in Ihrem AWS Konto abzurufen

Mit dem folgenden `get-server-certificate` Befehl werden alle Details zum angegebenen Serverzertifikat in Ihrem AWS Konto abgerufen.

```
aws iam get-server-certificate \
  --server-certificate-name myUpdatedServerCertificate
```

Ausgabe:

```
{
  "ServerCertificate": {
    "ServerCertificateMetadata": {
      "Path": "/",
      "ServerCertificateName": "myUpdatedServerCertificate",
      "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:server-certificate/myUpdatedServerCertificate",
      "UploadDate": "2019-04-22T21:13:44+00:00",
      "Expiration": "2019-10-15T22:23:16+00:00"
    },
    "CertificateBody": "-----BEGIN CERTIFICATE-----
      MIICiTCCAFICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
```

```

VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhdLQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJIIJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
"CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCcAfICQD6md
7oRw0uX0jANBgkqhkiG9w0BAQQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGT
AlldBMRAwDgYDVQQHEwdTZWF0drGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAs
TC0lBTSBDb25zb2x1MRIwEAYDVQsQQDEwLUZXN0Q21sYWxhZAdBgkqhkiG9w0BCQ
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgsYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAsTC0lBTSBDb2d5zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxh
ZAdBgkqhkiG9w0BCQEWEG5vb251QGfFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gjpEIbb30hjZncvcQAaRHhdLQWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FlkbFFbjvSfpJIIJ00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEWEG5vb251QGftsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
}
}

```

Verwenden Sie den `list-server-certificates` Befehl, um die in Ihrem AWS Konto verfügbaren Serverzertifikate aufzulisten.

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Verwalten von Serverzertifikaten](#).

- APIEinzelheiten finden Sie [GetServerCertificate](#) in der AWS CLI Befehlsreferenz.

get-service-last-accessed-details-with-entities

Das folgende Codebeispiel zeigt die Verwendung `get-service-last-accessed-details-with-entities`.

AWS CLI

Um einen Servicezugriffsbericht mit Details für einen Dienst abzurufen

Im folgenden `get-service-last-accessed-details-with-entities` Beispiel wird ein Bericht abgerufen, der Details zu IAM Benutzern und anderen Entitäten enthält, die auf den angegebenen Dienst zugegriffen haben. Verwenden Sie den `generate-service-last-accessed-details` Befehl, um einen Bericht zu generieren. Um eine Liste der Dienste abzurufen, auf die über Namespaces zugegriffen wird, verwenden Sie `get-service-last-accessed-details`

```
aws iam get-service-last-accessed-details-with-entities \  
  --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \  
  --service-namespace Lambda
```

Ausgabe:

```
{  
  "JobStatus": "COMPLETED",  
  "JobCreationDate": "2019-10-01T03:55:41.756Z",  
  "JobCompletionDate": "2019-10-01T03:55:42.533Z",  
  "EntityDetailsList": [  
    {  
      "EntityInfo": {  
        "Arn": "arn:aws:iam::123456789012:user/admin",  
        "Name": "admin",  
        "Type": "USER",  
        "Id": "AIDAI02XMPLENQEXAMPLE",  
        "Path": "/"  
      },  
      "LastAuthenticated": "2019-09-30T23:02:00Z"  
    },  
    {  
      "EntityInfo": {  
        "Arn": "arn:aws:iam::123456789012:user/developer",  
        "Name": "developer",  
        "Type": "USER",  
        "Id": "AIDAIBEYXMPL2YEXAMPLE",  
        "Path": "/"  
      },  
      "LastAuthenticated": "2019-09-16T19:34:00Z"  
    }  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verfeinerung von Berechtigungen bei der AWS Verwendung von Informationen, auf die zuletzt zugegriffen wurde](#). AWS IAM

- API-Einheiten finden Sie [GetServiceLastAccessedDetailsWithEntities](#) in der AWS CLI Befehlsreferenz.

get-service-last-accessed-details

Das folgende Codebeispiel zeigt die Verwendung von `get-service-last-accessed-details`.

AWS CLI

Um einen Servicezugriffsbericht abzurufen

Im folgenden `get-service-last-accessed-details` Beispiel wird ein zuvor generierter Bericht abgerufen, der die Dienste auflistet, auf die IAM Entitäten zugreifen. Verwenden Sie den `generate-service-last-accessed-details` Befehl, um einen Bericht zu generieren.

```
aws iam get-service-last-accessed-details \
  --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

Ausgabe:

```
{
  "JobStatus": "COMPLETED",
  "JobCreationDate": "2019-10-01T03:50:35.929Z",
  "ServicesLastAccessed": [
    ...
    {
      "ServiceName": "AWS Lambda",
      "LastAuthenticated": "2019-09-30T23:02:00Z",
      "ServiceNamespace": "lambda",
      "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/admin",
      "TotalAuthenticatedEntities": 6
    },
  ]
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Verfeinerung von Berechtigungen bei der AWS Verwendung von Informationen, auf die zuletzt zugegriffen wurde](#).

- APIEinzelheiten finden Sie [GetServiceLastAccessedDetails](#)in der AWS CLI Befehlsreferenz.

get-service-linked-role-deletion-status

Das folgende Codebeispiel zeigt die Verwendungget-service-linked-role-deletion-status.

AWS CLI

So überprüfen Sie den Status einer Anfrage zum Löschen einer serviceverknüpften Rolle

Im folgenden get-service-linked-role-deletion-status-Beispiel wird der Status einer früheren Anfrage zum Löschen einer serviceverknüpften Rolle angezeigt. Der Löschvorgang erfolgt asynchron. Wenn Sie die Anfrage stellen, erhalten Sie einen DeletionTaskId-Wert, den Sie als Parameter für diesen Befehl angeben.

```
aws iam get-service-linked-role-deletion-status \
  --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

Ausgabe:

```
{
  "Status": "SUCCEEDED"
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Verwenden von serviceverknüpften Rollen](#).

- APIEinzelheiten finden Sie [GetServiceLinkedRoleDeletionStatus](#)in der AWS CLI Befehlsreferenz.

get-ssh-public-key

Das folgende Codebeispiel zeigt die Verwendungget-ssh-public-key.

AWS CLI

Beispiel 1: Um einen SSH öffentlichen Schlüssel abzurufen, der an einen IAM Benutzer in SSH codierter Form angehängt ist

Der folgende `get-ssh-public-key` Befehl ruft den angegebenen SSH öffentlichen Schlüssel vom IAM Benutzer ab. `sofia` Die Ausgabe erfolgt in SSH Kodierung.

```
aws iam get-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE \  
  --encoding SSH
```

Ausgabe:

```
{  
  "SSHPublicKey": {  
    "UserName": "sofia",  
    "SSHPublicKeyId": "APKA123456789EXAMPLE",  
    "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",  
    "SSHPublicKeyBody": "ssh-rsa <<long encoded SSH string>>",  
    "Status": "Inactive",  
    "UploadDate": "2019-04-18T17:04:49+00:00"  
  }  
}
```

Beispiel 2: Um einen SSH öffentlichen Schlüssel abzurufen, der an einen IAM Benutzer in PEM codierter Form angehängt ist

Der folgende `get-ssh-public-key` Befehl ruft den angegebenen SSH öffentlichen Schlüssel vom IAM Benutzer ab. `sofia` Die Ausgabe erfolgt in PEM Kodierung.

```
aws iam get-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE \  
  --encoding PEM
```

Ausgabe:

```
{
```

```

    "SSHPublicKey": {
      "UserName": "sofia",
      "SSHPublicKeyId": "APKA123456789EXAMPLE",
      "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
      "SSHPublicKeyBody": ""-----BEGIN PUBLIC KEY-----\n<<long encoded PEM
string>>\n-----END PUBLIC KEY-----\n"",
      "Status": "Inactive",
      "UploadDate": "2019-04-18T17:04:49+00:00"
    }
  }
}

```

Weitere Informationen finden Sie unter [SSHTasten und SSH mit verwenden CodeCommit](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [GetSshPublicKey](#) in der AWS CLI Befehlsreferenz.

get-user-policy

Das folgende Codebeispiel zeigt die Verwendung `get-user-policy`.

AWS CLI

Um Richtlinien­details für einen IAM Benutzer aufzulisten

Der folgende `get-user-policy` Befehl listet die Details der angegebenen Richtlinie auf, die dem genannten IAM Benutzer zugeordnet ist Bob.

```

aws iam get-user-policy \
  --user-name Bob \
  --policy-name ExamplePolicy

```

Ausgabe:

```

{
  "UserName": "Bob",
  "PolicyName": "ExamplePolicy",
  "PolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": "*",
        "Resource": "*"
      }
    ]
  }
}

```

```
        "Effect": "Allow"
      }
    ]
  }
}
```

Verwenden Sie den `list-user-policies` Befehl, um eine Liste der Richtlinien für einen IAM Benutzer abzurufen.

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Richtlinien und Berechtigungen](#).

- API-Einheiten finden Sie [GetUserPolicy](#) in der AWS CLI Befehlsreferenz.

get-user

Das folgende Codebeispiel zeigt die Verwendung `get-user`.

AWS CLI

Um Informationen über einen IAM Benutzer zu erhalten

Mit dem folgenden `get-user` Befehl werden Informationen über den genannten IAM Benutzer abgerufen `Paulo`.

```
aws iam get-user \
  --user-name Paulo
```

Ausgabe:

```
{
  "User": {
    "UserName": "Paulo",
    "Path": "/",
    "CreateDate": "2019-09-21T23:03:13Z",
    "UserId": "AIDA123456789EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Paulo"
  }
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter IAM Benutzer [verwalten](#).

- API-Einheiten finden Sie [GetUser](#) in der AWS CLI Befehlsreferenz.

list-access-keys

Das folgende Codebeispiel zeigt die Verwendung `list-access-keys`.

AWS CLI

Um den Zugriffsschlüssel IDs für einen IAM Benutzer aufzulisten

Der folgende `list-access-keys` Befehl listet die Zugriffsschlüssel IDs für den genannten IAM Benutzer auf Bob.

```
aws iam list-access-keys \
  --user-name Bob
```

Ausgabe:

```
{
  "AccessKeyMetadata": [
    {
      "UserName": "Bob",
      "Status": "Active",
      "CreateDate": "2013-06-04T18:17:34Z",
      "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    {
      "UserName": "Bob",
      "Status": "Inactive",
      "CreateDate": "2013-06-06T20:42:26Z",
      "AccessKeyId": "AKIAI44QH8DHBEXAMPLE"
    }
  ]
}
```

Sie können die geheimen Zugriffsschlüssel für IAM Benutzer nicht auflisten. Bei Verlust der geheimen Zugangsschlüssel müssen Sie mit dem `create-access-keys`-Befehl neue Zugangsschlüssel erstellen.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verwaltung von Zugriffsschlüsseln für IAM AWS IAM Benutzer](#).

- API-Einheiten finden Sie [ListAccessKeys](#) in der AWS CLI Befehlsreferenz.

list-account-aliases

Das folgende Codebeispiel zeigt die Verwendung `list-account-aliases`.

AWS CLI

So listen Sie Konto-Aliase auf

Der folgende `list-account-aliases`-Befehl listet die Aliase für das aktuelle Konto auf.

```
aws iam list-account-aliases
```

Ausgabe:

```
{
  "AccountAliases": [
    "mycompany"
  ]
}
```

Weitere Informationen finden Sie unter [Ihre AWS Konto-ID und ihr Alias](#) im AWS IAM-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAccountAliases](#) in der AWS CLI Befehlsreferenz.

list-attached-group-policies

Das folgende Codebeispiel zeigt die Verwendung `list-attached-group-policies`.

AWS CLI

Um alle verwalteten Richtlinien aufzulisten, die der angegebenen Gruppe zugeordnet sind

In diesem Beispiel werden die Namen und die Namen ARNs der verwalteten Richtlinien zurückgegeben, die der Admins im AWS Konto genannten IAM Gruppe zugeordnet sind.

```
aws iam list-attached-group-policies \
  --group-name Admins
```

Ausgabe:


```
{
  "AttachedPolicies": [
    {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ],
  "IsTruncated": false
}
```

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Richtlinien und Berechtigungen](#).

- API-Einheiten finden Sie [ListAttachedGroupPolicies](#) in der AWS CLI Befehlsreferenz.

list-attached-role-policies

Das folgende Codebeispiel zeigt die Verwendung `list-attached-role-policies`.

AWS CLI

So listen Sie alle verwalteten Richtlinien auf, die der angegebenen Rolle angefügt sind

Dieser Befehl gibt die Namen und die Namen ARNs der verwalteten Richtlinien zurück, die der `SecurityAuditRole` im AWS Konto genannten IAM Rolle zugeordnet sind.

```
aws iam list-attached-role-policies \
  --role-name SecurityAuditRole
```

Ausgabe:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ],
}
```

```
"IsTruncated": false
}
```

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [ListAttachedRolePolicies](#) in der AWS CLI Befehlsreferenz.

list-attached-user-policies

Das folgende Codebeispiel zeigt die Verwendung `list-attached-user-policies`.

AWS CLI

Um alle verwalteten Richtlinien aufzulisten, die dem angegebenen Benutzer zugeordnet sind

Dieser Befehl gibt die Namen und ARNs die verwalteten Richtlinien für den Bob im AWS Konto genannten IAM Benutzer zurück.

```
aws iam list-attached-user-policies \
  --user-name Bob
```

Ausgabe:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ],
  "IsTruncated": false
}
```

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [ListAttachedUserPolicies](#) in der AWS CLI Befehlsreferenz.

list-entities-for-policy

Das folgende Codebeispiel zeigt die Verwendung `list-entities-for-policy`.

AWS CLI

Um alle Benutzer, Gruppen und Rollen aufzulisten, denen die angegebene verwaltete Richtlinie zugeordnet ist

In diesem Beispiel wird eine Liste von IAM Gruppen, Rollen und Benutzern zurückgegeben, denen die Richtlinie `arn:aws:iam::123456789012:policy/TestPolicy` angehängt ist.

```
aws iam list-entities-for-policy \
  --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

Ausgabe:

```
{
  "PolicyGroups": [
    {
      "GroupName": "Admins",
      "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
    }
  ],
  "PolicyUsers": [
    {
      "UserName": "Alice",
      "UserId": "AIDACKCEVSQ6C2EXAMPLE"
    }
  ],
  "PolicyRoles": [
    {
      "RoleName": "DevRole",
      "RoleId": "AR0ADBQP57FF2AEXAMPLE"
    }
  ],
  "IsTruncated": false
}
```

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Richtlinien und Berechtigungen](#).

- API-Einheiten finden Sie [ListEntitiesForPolicy](#) in der AWS CLI Befehlsreferenz.

list-group-policies

Das folgende Codebeispiel zeigt die Verwendung `list-group-policies`.

AWS CLI

Um alle Inline-Richtlinien aufzulisten, die der angegebenen Gruppe zugeordnet sind

Der folgende `list-group-policies` Befehl listet die Namen der Inline-Richtlinien auf, die der Admins im aktuellen Konto genannten IAM Gruppe zugeordnet sind.

```
aws iam list-group-policies \  
  --group-name Admins
```

Ausgabe:

```
{  
  "PolicyNames": [  
    "AdminRoot",  
    "ExamplePolicy"  
  ]  
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [IAM Richtlinien verwalten](#).

- API Einheiten finden Sie [ListGroupPolicies](#) in der AWS CLI Befehlsreferenz.

list-groups-for-user

Das folgende Codebeispiel zeigt die Verwendung `list-groups-for-user`.

AWS CLI

Um die Gruppen aufzulisten, zu denen ein IAM Benutzer gehört

Der folgende `list-groups-for-user` Befehl zeigt die Gruppen an, zu denen der angegebene IAM Benutzer Bob gehört.

```
aws iam list-groups-for-user \  
  --user-name Bob
```

Ausgabe:

```
{
  "Groups": [
    {
      "Path": "/",
      "CreateDate": "2013-05-06T01:18:08Z",
      "GroupId": "AKIAIOSFODNN7EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/Admin",
      "GroupName": "Admin"
    },
    {
      "Path": "/",
      "CreateDate": "2013-05-06T01:37:28Z",
      "GroupId": "AKIAI44QH8DHBEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/s3-Users",
      "GroupName": "s3-Users"
    }
  ]
}
```

Weitere Informationen finden Sie im [IAM Benutzerhandbuch unter Benutzergruppen verwalten](#). AWS IAM

- API-Einheiten finden Sie [ListGroupsForUser](#) in der AWS CLI Befehlsreferenz.

list-groups

Das folgende Codebeispiel zeigt die Verwendung `list-groups`.

AWS CLI

Um die IAM Gruppen für das Girokonto aufzulisten

Der folgende `list-groups` Befehl listet die IAM Gruppen im aktuellen Konto auf.

```
aws iam list-groups
```

Ausgabe:

```
{
  "Groups": [
    {
      "Path": "/",
```

```

    "CreateDate": "2013-06-04T20:27:27.972Z",
    "GroupId": "AIDACKCEVSQ6C2EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "GroupName": "Admins"
  },
  {
    "Path": "/",
    "CreateDate": "2013-04-16T20:30:42Z",
    "GroupId": "AIDGPMS9R04H3FEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
    "GroupName": "S3-Admins"
  }
]
}

```

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Benutzergruppen verwalten](#).AWS IAM

- APIEinzelheiten finden Sie [ListGroups](#)in der AWS CLI Befehlsreferenz.

list-instance-profile-tags

Das folgende Codebeispiel zeigt die Verwendung `list-instance-profile-tags`.

AWS CLI

Um die einem Instanzprofil angehängten Tags aufzulisten

Mit dem folgenden `list-instance-profile-tags` Befehl wird die Liste der Tags abgerufen, die dem angegebenen Instanzprofil zugeordnet sind.

```

aws iam list-instance-profile-tags \
  --instance-profile-name deployment-role

```

Ausgabe:

```

{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {

```

```

        "Key": "Department",
        "Value": "Accounting"
    }
]
}

```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [ListInstanceProfileTags](#)in der AWS CLI Befehlsreferenz.

list-instance-profiles-for-role

Das folgende Codebeispiel zeigt die Verwendung `list-instance-profiles-for-role`.

AWS CLI

Um die Instanzprofile für eine IAM Rolle aufzulisten

Der folgende `list-instance-profiles-for-role` Befehl listet die Instanzprofile auf, die der Rolle zugeordnet sind `Test-Rolle`.

```

aws iam list-instance-profiles-for-role \
    --role-name Test-Role

```

Ausgabe:

```

{
  "InstanceProfiles": [
    {
      "InstanceId": "AIDGPM9R04H3FEXAMPLE",
      "Roles": [
        {
          "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
          "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
          "CreateDate": "2013-06-07T20:42:15Z",
          "RoleName": "Test-Role",
          "Path": "/",
          "Arn": "arn:aws:iam::123456789012:role/Test-Role"
        }
      ],
      "CreateDate": "2013-06-07T21:05:24Z",
      "InstanceProfileName": "ExampleInstanceProfile",
      "Path": "/",
    }
  ]
}

```

```
    "Arn": "arn:aws:iam::123456789012:instance-profile/
ExampleInstanceProfile"
  }
]
}
```

Weitere Informationen finden Sie unter [Verwenden von Instanzprofilen](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [ListInstanceProfilesForRole](#) in der AWS CLI Befehlsreferenz.

list-instance-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-instance-profiles`.

AWS CLI

Um die Instanzprofile für das Konto aufzulisten

Der folgende `list-instance-profiles` Befehl listet die Instanzprofile auf, die dem aktuellen Konto zugeordnet sind.

```
aws iam list-instance-profiles
```

Ausgabe:

```
{
  "InstanceProfiles": [
    {
      "Path": "/",
      "InstanceProfileName": "example-dev-role",
      "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:instance-profile/example-dev-role",
      "CreateDate": "2023-09-21T18:17:41+00:00",
      "Roles": [
        {
          "Path": "/",
          "RoleName": "example-dev-role",
          "RoleId": "AR0AJ520TH4H7LEXAMPLE",
          "Arn": "arn:aws:iam::123456789012:role/example-dev-role",
          "CreateDate": "2023-09-21T18:17:40+00:00",
          "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
```



```
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "ec2.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}
],
{
  "Path": "/",
  "InstanceProfileName": "example-s3-role",
  "InstanceProfileId": "AIPAJVJVNRIQFEXAMPLE",
  "Arn": "arn:aws:iam::123456789012:instance-profile/example-s3-role",
  "CreateDate": "2023-09-21T18:18:50+00:00",
  "Roles": [
    {
      "Path": "/",
      "RoleName": "example-s3-role",
      "RoleId": "AROAINUBC507XLEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:role/example-s3-role",
      "CreateDate": "2023-09-21T18:18:49+00:00",
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      }
    }
  ]
}
]
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch [unter Verwenden von Instanzprofilen](#).

- APIEinzelheiten finden Sie [ListInstanceProfiles](#) in der AWS CLI Befehlsreferenz.

list-mfa-device-tags

Das folgende Codebeispiel zeigt die Verwendung `list-mfa-device-tags`.

AWS CLI

Um die an ein MFA Gerät angeschlossenen Tags aufzulisten

Mit dem folgenden `list-mfa-device-tags` Befehl wird die Liste der Tags abgerufen, die dem angegebenen MFA Gerät zugeordnet sind.

```
aws iam list-mfa-device-tags \  
  --serial-number arn:aws:iam::123456789012:mfa/alice
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im [AWS IAMBenutzerhandbuch unter IAM Ressourcen](#) kennzeichnen.

- APIEinzelheiten finden Sie [ListMfaDeviceTags](#) in der AWS CLI Befehlsreferenz.

list-mfa-devices

Das folgende Codebeispiel zeigt die Verwendung `list-mfa-devices`.

AWS CLI

Um alle MFA Geräte für einen bestimmten Benutzer aufzulisten

In diesem Beispiel werden Details zu dem dem IAM Benutzer zugewiesenen MFA Gerät zurückgegebenBob.

```
aws iam list-mfa-devices \  
  --user-name Bob
```

Ausgabe:

```
{  
  "MFADevices": [  
    {  
      "UserName": "Bob",  
      "SerialNumber": "arn:aws:iam::123456789012:mfa/Bob",  
      "EnableDate": "2019-10-28T20:37:09+00:00"  
    },  
    {  
      "UserName": "Bob",  
      "SerialNumber": "GAKT12345678",  
      "EnableDate": "2023-02-18T21:44:42+00:00"  
    },  
    {  
      "UserName": "Bob",  
      "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/  
fidosecuritykey1-7XNL7NFNLZ123456789EXAMPLE",  
      "EnableDate": "2023-09-19T02:25:35+00:00"  
    },  
    {  
      "UserName": "Bob",  
      "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/  
fidosecuritykey2-VDRQTDBBN5123456789EXAMPLE",  
      "EnableDate": "2023-09-19T01:49:18+00:00"  
    }  
  ]  
}
```

Weitere Informationen finden Sie [im AWS IAM Benutzerhandbuch unter Verwenden der Multi-Faktor-Authentifizierung \(MFA\)](#).

- API Einheiten finden Sie [ListMfaDevices](#) in der AWS CLI Befehlsreferenz.

list-open-id-connect-provider-tags

Das folgende Codebeispiel zeigt die Verwendung `list-open-id-connect-provider-tags`.

AWS CLI

Um die Tags aufzulisten, die an einen OpenID Connect (OIDC) -kompatiblen Identitätsanbieter angehängt sind

Der folgende `list-open-id-connect-provider-tags` Befehl ruft die Liste der Tags ab, die dem angegebenen OIDC Identitätsanbieter zugeordnet sind.

```
aws iam list-open-id-connect-provider-tags \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [ListOpenIdConnectProviderTags](#) in der AWS CLI Befehlsreferenz.

list-open-id-connect-providers

Das folgende Codebeispiel zeigt die Verwendung `list-open-id-connect-providers`.

AWS CLI

Um Informationen über die OpenID Connect-Anbieter im AWS Konto aufzulisten

Dieses Beispiel gibt eine Liste ARNS aller OpenID Connect-Anbieter zurück, die im aktuellen AWS Konto definiert sind.

```
aws iam list-open-id-connect-providers
```

Ausgabe:

```
{
  "OpenIDConnectProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erstellen von OpenID Connect \(OIDC\) -Identitätsanbietern](#) im AWS IAM Benutzerhandbuch.

- API Einzelheiten finden Sie unter [ListOpenIdConnectProviders AWS CLI Befehlsreferenz](#).

list-policies-granting-service-access

Das folgende Codebeispiel zeigt die Verwendung `list-policies-granting-service-access`.

AWS CLI

Um die Richtlinien aufzulisten, die einem Prinzipalzugriff auf den angegebenen Dienst gewähren

Im folgenden `list-policies-granting-service-access` Beispiel wird die Liste der Richtlinien abgerufen, die dem IAM Benutzer `sofia` Zugriff auf den AWS CodeCommit Dienst gewähren.

```
aws iam list-policies-granting-service-access \
  --arn arn:aws:iam::123456789012:user/sofia \
  --service-namespaces codecommit
```

Ausgabe:

```
{
  "PoliciesGrantingServiceAccess": [
```

```
{
  "ServiceNamespace": "codecommit",
  "Policies": [
    {
      "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
      "PolicyType": "INLINE",
      "EntityType": "USER",
      "EntityName": "sofia"
    }
  ]
},
"IsTruncated": false
}
```

Weitere Informationen finden Sie im Benutzerhandbuch [unter Verwenden IAM mit CodeCommit: Git-Anmeldeinformationen, SSH Schlüsseln und AWS Zugriffsschlüsseln](#).AWS IAM

- API-Einheiten finden Sie [ListPoliciesGrantingServiceAccess](#) in der AWS CLI Befehlsreferenz.

list-policies

Das folgende Codebeispiel zeigt die Verwendung `list-policies`.

AWS CLI

Um verwaltete Richtlinien aufzulisten, die für Ihr AWS Konto verfügbar sind

In diesem Beispiel wird eine Sammlung der ersten beiden verwalteten Richtlinien zurückgegeben, die im aktuellen AWS Konto verfügbar sind.

```
aws iam list-policies \
  --max-items 3
```

Ausgabe:

```
{
  "Policies": [
    {
      "PolicyName": "AWSCloudTrailAccessPolicy",
      "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",
      "Path": "/",

```

```

    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2019-09-04T17:43:42+00:00",
    "UpdateDate": "2019-09-04T17:43:42+00:00"
  },
  {
    "PolicyName": "AdministratorAccess",
    "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
    "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 6,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2015-02-06T18:39:46+00:00",
    "UpdateDate": "2015-02-06T18:39:46+00:00"
  },
  {
    "PolicyName": "PowerUserAccess",
    "PolicyId": "ANPAJYRXTIB4FOVS3ZXS",
    "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
    "Path": "/",
    "DefaultVersionId": "v5",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2015-02-06T18:39:47+00:00",
    "UpdateDate": "2023-07-06T22:04:00+00:00"
  }
],
"NextToken": "EXAMPLErZXIi0iBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQi0iA4fQ=="
}

```

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [ListPolicies](#) in der AWS CLI Befehlsreferenz.

list-policy-tags

Das folgende Codebeispiel zeigt die Verwendung `list-policy-tags`.

AWS CLI

Um die mit einer verwalteten Richtlinie verknüpften Tags aufzulisten

Mit dem folgenden `list-policy-tags` Befehl wird die Liste der Tags abgerufen, die der angegebenen verwalteten Richtlinie zugeordnet sind.

```
aws iam list-policy-tags \  
  --policy-arn arn:aws:iam::123456789012:policy/billing-access
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im [AWS IAM Benutzerhandbuch unter IAM Ressourcen](#) kennzeichnen.

- API-Einheiten finden Sie [ListPolicyTags](#) in der AWS CLI Befehlsreferenz.

`list-policy-versions`

Das folgende Codebeispiel zeigt die Verwendung `list-policy-versions`.

AWS CLI

Um Informationen zu den Versionen der angegebenen verwalteten Richtlinie aufzulisten

In diesem Beispiel wird die Liste der verfügbaren Versionen der Richtlinie zurückgegeben, deren ARN ist `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam list-policy-versions \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```



```
--policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Ausgabe:

```
{
  "IsTruncated": false,
  "Versions": [
    {
      "VersionId": "v2",
      "IsDefaultVersion": true,
      "CreateDate": "2015-06-02T23:19:44Z"
    },
    {
      "VersionId": "v1",
      "IsDefaultVersion": false,
      "CreateDate": "2015-06-02T22:30:47Z"
    }
  ]
}
```

Weitere Informationen finden Sie [IAMim AWS IAMBenutzerhandbuch unter Richtlinien und Berechtigungen](#).

- APIEinzelheiten finden Sie [ListPolicyVersions](#) in der AWS CLI Befehlsreferenz.

list-role-policies

Das folgende Codebeispiel zeigt die Verwendung `list-role-policies`.

AWS CLI

Um die einer IAM Rolle zugewiesenen Richtlinien aufzulisten

Der folgende `list-role-policies` Befehl listet die Namen der Berechtigungsrichtlinien für die angegebene IAM Rolle auf.

```
aws iam list-role-policies \  
  --role-name Test-Role
```

Ausgabe:

```
{
```

```
"PolicyNames": [  
    "ExamplePolicy"  
]  
}
```

Verwenden Sie den `get-role`-Befehl, um die einer Rolle angefügten Vertrauensrichtlinie anzuzeigen. Verwenden Sie den `get-role-policy`-Befehl, um die Details einer Berechtigungsrichtlinie anzuzeigen.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRollen erstellen](#).

- APIEinzelheiten finden Sie [ListRolePolicies](#) in der AWS CLI Befehlsreferenz.

list-role-tags

Das folgende Codebeispiel zeigt die Verwendung `list-role-tags`.

AWS CLI

Um die einer Rolle zugewiesenen Tags aufzulisten

Mit dem folgenden `list-role-tags` Befehl wird die Liste der Tags abgerufen, die der angegebenen Rolle zugeordnet sind.

```
aws iam list-role-tags \  
  --role-name production-role
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {  
      "Key": "DeptID",  
      "Value": "12345"  
    }  
  ],  
  "IsTruncated": false  
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [ListRoleTags](#)in der AWS CLI Befehlsreferenz.

list-roles

Das folgende Codebeispiel zeigt die Verwendung `list-roles`.

AWS CLI

Um IAM Rollen für das aktuelle Konto aufzulisten

Der folgende `list-roles` Befehl listet die IAM Rollen für das aktuelle Konto auf.

```
aws iam list-roles
```

Ausgabe:

```
{
  "Roles": [
    {
      "Path": "/",
      "RoleName": "ExampleRole",
      "RoleId": "AR0AJ520TH4H7LEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
      "CreateDate": "2017-09-12T19:23:36+00:00",
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "MaxSessionDuration": 3600
    },
    {
```

```

    "Path": "/example_path/",
    "RoleName": "ExampleRoleWithPath",
    "RoleId": "AROAI4QRP7UFT7EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:role/example_path/
ExampleRoleWithPath",
    "CreateDate": "2023-09-21T20:29:38+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    },
    "MaxSessionDuration": 3600
  }
]
}

```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRollen erstellen](#).

- APIEinzelheiten finden Sie [ListRoles](#) in der AWS CLI Befehlsreferenz.

list-saml-provider-tags

Das folgende Codebeispiel zeigt die Verwendung `list-saml-provider-tags`.

AWS CLI

Um die an einen SAML Anbieter angehängten Tags aufzulisten

Mit dem folgenden `list-saml-provider-tags` Befehl wird die Liste der Tags abgerufen, die dem angegebenen SAML Anbieter zugeordnet sind.

```

aws iam list-saml-provider-tags \
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS

```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [ListSamlProviderTags](#) in der AWS CLI Befehlsreferenz.

list-saml-providers

Das folgende Codebeispiel zeigt die Verwendung `list-saml-providers`.

AWS CLI

Um die SAML Anbieter im AWS Konto aufzulisten

In diesem Beispiel wird die Liste der SAML 2.0-Anbieter abgerufen, die im AWS Girokonto erstellt wurde.

```
aws iam list-saml-providers
```

Ausgabe:

```
{
  "SAMLProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
      "ValidUntil": "2015-06-05T22:45:14Z",
      "CreateDate": "2015-06-05T22:45:14Z"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Creating IAM SAML Identity Providers](#).

- APIEinzelheiten finden Sie unter [ListSAMLProviders](#) in der AWS CLI Befehlsreferenz.

list-server-certificate-tags

Das folgende Codebeispiel zeigt, wie man `list-server-certificate-tags`.

AWS CLI

Um die an ein Serverzertifikat angehängten Tags aufzulisten

Mit dem folgenden `list-server-certificate-tags` Befehl wird die Liste der Tags abgerufen, die dem angegebenen Serverzertifikat zugeordnet sind.

```
aws iam list-server-certificate-tags \  
  --server-certificate-name ExampleCertificate
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Weitere Informationen finden Sie im [AWS IAMBenutzerhandbuch unter IAM Ressourcen](#) kennzeichnen.

- APIEinzelheiten finden Sie [ListServerCertificateTags](#) in der AWS CLI Befehlsreferenz.

list-server-certificates

Das folgende Codebeispiel zeigt die Verwendung `list-server-certificates`.

AWS CLI

Um die Serverzertifikate in Ihrem AWS Konto aufzulisten

Der folgende `list-server-certificates` Befehl listet alle Serverzertifikate auf, die in Ihrem AWS Konto gespeichert sind und zur Verwendung verfügbar sind.

```
aws iam list-server-certificates
```

Ausgabe:

```
{
  "ServerCertificateMetadataList": [
    {
      "Path": "/",
      "ServerCertificateName": "myUpdatedServerCertificate",
      "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:server-certificate/myUpdatedServerCertificate",
      "UploadDate": "2019-04-22T21:13:44+00:00",
      "Expiration": "2019-10-15T22:23:16+00:00"
    },
    {
      "Path": "/cloudfront/",
      "ServerCertificateName": "MyTestCert",
      "ServerCertificateId": "ASCAEXAMPLE456EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:server-certificate/Org1/Org2/MyTestCert",
      "UploadDate": "2015-04-21T18:14:16+00:00",
      "Expiration": "2018-01-14T17:52:36+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Verwalten von Serverzertifikaten](#).

- API-Einheiten finden Sie [ListServerCertificates](#) in der AWS CLI Befehlsreferenz.

list-service-specific-credential

Das folgende Codebeispiel zeigt die Verwendung `list-service-specific-credential`.

AWS CLI

Beispiel 1: Listet die dienstspezifischen Anmeldeinformationen für einen Benutzer auf

Im folgenden `list-service-specific-credentials` Beispiel werden alle dienstspezifischen Anmeldeinformationen angezeigt, die dem angegebenen Benutzer zugewiesen wurden. Passwörter sind nicht in der Antwort enthalten.

```
aws iam list-service-specific-credentials \  
  --user-name sofia
```

Ausgabe:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",  
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

Beispiel 2: Listet die dienstspezifischen Anmeldeinformationen für einen Benutzer auf, der nach einem bestimmten Dienst gefiltert wurde

Im folgenden `list-service-specific-credentials` Beispiel werden die dienstspezifischen Anmeldeinformationen angezeigt, die dem Benutzer zugewiesen wurden, der die Anfrage stellt. Die Liste wird so gefiltert, dass sie nur die Anmeldeinformationen für den angegebenen Dienst enthält. Passwörter sind nicht in der Antwort enthalten.

```
aws iam list-service-specific-credentials \  
  --service-name codecommit.amazonaws.com
```

Ausgabe:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",
```



```
"ServiceName": "codecommit.amazonaws.com",
"ServiceUserName": "sofia-at-123456789012",
"ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
"UserName": "sofia",
"Status": "Active"
}
}
```

Weitere Informationen finden Sie CodeCommit im AWS CodeCommit Benutzerhandbuch unter [Erstellen von Git-Anmeldeinformationen für HTTPS Verbindungen zu](#).

- API-Einheiten finden Sie [ListServiceSpecificCredential](#) in der AWS CLI Befehlsreferenz.

list-service-specific-credentials

Das folgende Codebeispiel zeigt die Verwendung `list-service-specific-credentials`.

AWS CLI

Um eine Liste mit Anmeldeinformationen abzurufen

Das folgende `list-service-specific-credentials` Beispiel listet die Anmeldeinformationen auf, die für den HTTPS Zugriff auf AWS CodeCommit Repositorys für einen Benutzer mit dem Namen `developer` generiert wurden.

```
aws iam list-service-specific-credentials \
  --user-name developer \
  --service-name codecommit.amazonaws.com
```

Ausgabe:

```
{
  "ServiceSpecificCredentials": [
    {
      "UserName": "developer",
      "Status": "Inactive",
      "ServiceUserName": "developer-at-123456789012",
      "CreateDate": "2019-10-01T04:31:41Z",
      "ServiceSpecificCredentialId": "ACCAQFODXMPL4YFHP7DZE",
      "ServiceName": "codecommit.amazonaws.com"
    },
    {
```

```

        "UserName": "developer",
        "Status": "Active",
        "ServiceUserName": "developer+1-at-123456789012",
        "CreateDate": "2019-10-01T04:31:45Z",
        "ServiceSpecificCredentialId": "ACCAQFOXMPL6VW57M7AJP",
        "ServiceName": "codecommit.amazonaws.com"
    }
]
}

```

Weitere Informationen finden Sie CodeCommit im AWS CodeCommit Benutzerhandbuch unter [Erstellen von Git-Anmeldeinformationen für HTTPS Verbindungen zu](#).

- API-Einheiten finden Sie [ListServiceSpecificCredentials](#) in der AWS CLI Befehlsreferenz.

list-signing-certificates

Das folgende Codebeispiel zeigt die Verwendung `list-signing-certificates`.

AWS CLI

Um die Signaturzertifikate für einen IAM Benutzer aufzulisten

Der folgende `list-signing-certificates` Befehl listet die Signaturzertifikate für den genannten IAM Benutzer auf Bob.

```
aws iam list-signing-certificates \
  --user-name Bob
```

Ausgabe:

```

{
  "Certificates": [
    {
      "UserName": "Bob",
      "Status": "Inactive",
      "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----
END CERTIFICATE-----",
      "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",
      "UploadDate": "2013-06-06T21:40:08Z"
    }
  ]
}

```

```
}
```

Weitere Informationen finden Sie unter [Signaturzertifikate verwalten](#) im EC2Amazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [ListSigningCertificates](#) in der AWS CLI Befehlsreferenz.

list-ssh-public-keys

Das folgende Codebeispiel zeigt die Verwendung `list-ssh-public-keys`.

AWS CLI

Um die SSH öffentlichen Schlüssel aufzulisten, die einem IAM Benutzer zugewiesen sind

Das folgende `list-ssh-public-keys` Beispiel listet die SSH öffentlichen Schlüssel auf, die dem IAM Benutzer zugewiesen sind `sofia`.

```
aws iam list-ssh-public-keys \  
  --user-name sofia
```

Ausgabe:

```
{  
  "SSHPublicKeys": [  
    {  
      "UserName": "sofia",  
      "SSHPublicKeyId": "APKA1234567890EXAMPLE",  
      "Status": "Inactive",  
      "UploadDate": "2019-04-18T17:04:49+00:00"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwenden von SSH Schlüsseln und SSH mit CodeCommit](#) im AWS IAM Benutzerhandbuch

- APIEinzelheiten finden Sie [ListSshPublicKeys](#) in der AWS CLI Befehlsreferenz.

list-user-policies

Das folgende Codebeispiel zeigt die Verwendung `list-user-policies`.

AWS CLI

Um Richtlinien für einen IAM Benutzer aufzulisten

Der folgende `list-user-policies` Befehl listet die Richtlinien auf, die dem genannten IAM Benutzer zugeordnet sindBob.

```
aws iam list-user-policies \  
  --user-name Bob
```

Ausgabe:

```
{  
  "PolicyNames": [  
    "ExamplePolicy",  
    "TestPolicy"  
  ]  
}
```

Weitere Informationen finden Sie [im IAM Benutzerhandbuch unter Einen Benutzer in Ihrem AWS Konto erstellen](#).AWS IAM

- APIEinzelheiten finden Sie [ListUserPolicies](#) in der AWS CLI Befehlsreferenz.

list-user-tags

Das folgende Codebeispiel zeigt die Verwendung `list-user-tags`.

AWS CLI

Um die einem Benutzer zugewiesenen Tags aufzulisten

Mit dem folgenden `list-user-tags` Befehl wird die Liste der Tags abgerufen, die dem angegebenen IAM Benutzer zugeordnet sind.

```
aws iam list-user-tags \  
  --user-name alice
```

Ausgabe:

```
{
```

```
"Tags": [
  {
    "Key": "Department",
    "Value": "Accounting"
  },
  {
    "Key": "DeptID",
    "Value": "12345"
  }
],
"IsTruncated": false
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [ListUserTags](#)in der AWS CLI Befehlsreferenz.

list-users

Das folgende Codebeispiel zeigt die Verwendung `list-users`.

AWS CLI

Um IAM Benutzer aufzulisten

Der folgende `list-users` Befehl listet die IAM Benutzer im aktuellen Konto auf.

```
aws iam list-users
```

Ausgabe:

```
{
  "Users": [
    {
      "UserName": "Adele",
      "Path": "/",
      "CreateDate": "2013-03-07T05:14:48Z",
      "UserId": "AKIAI44QH8DHBEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/Adele"
    },
    {
```

```
    "UserName": "Bob",
    "Path": "/",
    "CreateDate": "2012-09-21T23:03:13Z",
    "UserId": "AKIAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Bob"
  }
]
```

Weitere Informationen finden Sie im Benutzerhandbuch unter IAM AWS IAM Benutzer [auflisten](#).

- API-Einheiten finden Sie [ListUsers](#) in der AWS CLI Befehlsreferenz.

list-virtual-mfa-devices

Das folgende Codebeispiel zeigt die Verwendung `list-virtual-mfa-devices`.

AWS CLI

Um virtuelle MFA Geräte aufzulisten

Der folgende `list-virtual-mfa-devices` Befehl listet die virtuellen MFA Geräte auf, die für das aktuelle Konto konfiguriert wurden.

```
aws iam list-virtual-mfa-devices
```

Ausgabe:

```
{
  "VirtualMFADevices": [
    {
      "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
    },
    {
      "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Aktivieren eines Geräts mit virtueller Multifaktor-Authentifizierung \(MFA\)](#).

- API-Einheiten finden Sie [ListVirtualMfaDevices](#) in der AWS CLI Befehlsreferenz.

put-group-policy

Das folgende Codebeispiel zeigt die Verwendung `put-group-policy`.

AWS CLI

So fügen Sie eine Richtlinie zu einer Gruppe hinzu

Mit dem folgenden `put-group-policy` Befehl wird der IAM Gruppe mit dem Namen eine Richtlinie hinzugefügt `Admins`.

```
aws iam put-group-policy \  
  --group-name Admins \  
  --policy-document file://AdminPolicy.json \  
  --policy-name AdminRoot
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Die Richtlinie ist als JSON Dokument in der `AdminPolicyJSON`-Datei definiert. (Der Dateiname und die Erweiterung sind nicht von Bedeutung.)

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [IAM Richtlinien verwalten](#).

- API-Einheiten finden Sie [PutGroupPolicy](#) in der AWS CLI Befehlsreferenz.

put-role-permissions-boundary

Das folgende Codebeispiel zeigt die Verwendung `put-role-permissions-boundary`.

AWS CLI

Beispiel 1: Um eine auf einer benutzerdefinierten Richtlinie basierende Berechtigungsgrenze auf eine IAM Rolle anzuwenden

Im folgenden `put-role-permissions-boundary` Beispiel wird die benutzerdefinierte Richtlinie angewendet, die `intern-boundary` als Berechtigungsgrenze für die angegebene IAM Rolle bezeichnet wird.

```
aws iam put-role-permissions-boundary \  
  --role-name AdminRole \  
  --policy-name AdminRoot \  
  --policy-document file://AdminPolicy.json
```

```
--permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
--role-name lambda-application-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um eine auf einer AWS verwalteten Richtlinie basierende Berechtigungsgrenze auf eine IAM Rolle anzuwenden

Im folgenden `put-role-permissions-boundary` Beispiel AWS wird die verwaltete `PowerUserAccess` Richtlinie als Berechtigungsgrenze für die angegebene IAM Rolle angewendet.

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --role-name x-account-admin
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Ändern einer Rolle](#).

- API Einheiten finden Sie [PutRolePermissionsBoundary](#) in der AWS CLI Befehlsreferenz.

put-role-policy

Das folgende Codebeispiel zeigt die Verwendung `put-role-policy`.

AWS CLI

Um einer IAM Rolle eine Berechtigungsrichtlinie zuzuweisen

Der folgende `put-role-policy`-Befehl fügt der Rolle mit dem Namen `Test-Role` eine Berechtigungsrichtlinie hinzu.

```
aws iam put-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Die Richtlinie ist als JSON Dokument in der `AdminPolicyJSON`-Datei definiert. (Der Dateiname und die Erweiterung sind nicht von Bedeutung.)

Verwenden Sie den `update-assume-role-policy`-Befehl, um einer Rolle eine Vertrauensrichtlinie anzufügen.

Weitere Informationen finden Sie unter [Ändern einer Rolle](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [PutRolePolicy](#) in der AWS CLI Befehlsreferenz.

put-user-permissions-boundary

Das folgende Codebeispiel zeigt die Verwendung `put-user-permissions-boundary`.

AWS CLI

Beispiel 1: Um einem IAM Benutzer eine auf einer benutzerdefinierten Richtlinie basierende Berechtigungsgrenze zuzuweisen

Im folgenden `put-user-permissions-boundary` Beispiel wird eine benutzerdefinierte Richtlinie angewendet, die `intern-boundary` als Berechtigungsgrenze für den angegebenen IAM Benutzer bezeichnet wird.

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --user-name intern
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um eine auf einer AWS verwalteten Richtlinie basierende Berechtigungsgrenze auf einen IAM Benutzer anzuwenden

Im folgenden `put-user-permissions-boundary` Beispiel AWS wird die verwaltete Richtlinie, die `PowerUserAccess` als Berechtigungsgrenze bezeichnet wird, für den angegebenen IAM Benutzer angewendet.

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --user-name developer
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Hinzufügen und Entfernen von IAM Identitätsberechtigungen](#).

- API-Einheiten finden Sie [PutUserPermissionsBoundary](#) in der AWS CLI Befehlsreferenz.

put-user-policy

Das folgende Codebeispiel zeigt die Verwendung `put-user-policy`.

AWS CLI

Um eine Richtlinie an einen IAM Benutzer anzuhängen

Mit dem folgenden `put-user-policy` Befehl wird dem IAM Benutzer mit dem Namen Bob eine Richtlinie zugewiesen.

```
aws iam put-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Die Richtlinie ist als JSON Dokument in der AdminPolicyJSON-Datei definiert. (Der Dateiname und die Erweiterung sind nicht von Bedeutung.)

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Hinzufügen und Entfernen von IAM Identitätsberechtigungen](#).

- API-Einheiten finden Sie [PutUserPolicy](#) in der AWS CLI Befehlsreferenz.

remove-client-id-from-open-id-connect-provider

Das folgende Codebeispiel zeigt die Verwendung `remove-client-id-from-open-id-connect-provider`.

AWS CLI

Um die angegebene Client-ID aus der Liste der Clients zu entfernen, die für den angegebenen IAM OpenID Connect-Anbieter IDs registriert sind

In diesem Beispiel wird die Client-ID `My-TestApp-3` aus der Liste der Clients entfernt, die dem IAM OIDC Anbieter IDs zugeordnet sind, dessen ARN `istarn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com`.

```
aws iam remove-client-id-from-open-id-connect-provider
  --client-id My-TestApp-3 \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen von OpenID Connect \(OIDC\) -Identitätsanbietern](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie unter [RemoveClientIdFromOpenIdConnectProvider AWS CLIBefehlsreferenz](#).

remove-role-from-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `remove-role-from-instance-profile`.

AWS CLI

Um eine Rolle aus einem Instanzprofil zu entfernen

Der folgende `remove-role-from-instance-profile` Befehl entfernt die benannte Rolle `Test-Role` aus dem genannten Instanzprofil `ExampleInstanceProfile`.

```
aws iam remove-role-from-instance-profile \
  --instance-profile-name ExampleInstanceProfile \
  --role-name Test-Role
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch [unter Verwenden von Instanzprofilen](#).

- APIEinzelheiten finden Sie [RemoveRoleFromInstanceProfile](#) in der AWS CLI Befehlsreferenz.

remove-user-from-group

Das folgende Codebeispiel zeigt die Verwendung `remove-user-from-group`.

AWS CLI

Um einen Benutzer aus einer IAM Gruppe zu entfernen

Mit dem folgenden `remove-user-from-group` Befehl wird der angegebene Benutzer Bob aus der genannten IAM Gruppe entfernt `Admins`.

```
aws iam remove-user-from-group \  
  --user-name Bob \  
  --group-name Admins
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Hinzufügen und Entfernen von Benutzern in einer IAM AWS IAM Benutzergruppe](#).

- API-Einheiten finden Sie [RemoveUserFromGroup](#) unter AWS CLI Befehlsreferenz.

reset-service-specific-credential

Das folgende Codebeispiel zeigt die Verwendung `reset-service-specific-credential`.

AWS CLI

Beispiel 1: Setzen Sie das Passwort für einen dienstspezifischen Berechtigungsnachweis zurück, der dem Benutzer, der die Anfrage gestellt hat, zugeordnet ist

Im folgenden `reset-service-specific-credential` Beispiel wird ein neues kryptografisch sicheres Passwort für die angegebenen dienstspezifischen Anmeldeinformationen generiert, die dem Benutzer zugeordnet sind, der die Anfrage stellt.

```
aws iam reset-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Ausgabe:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServicePassword": "+oaFsNk7tLco+C/obP9Ghhc0zGcK0ayTmE3LnAmAmH4=",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",  
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

```
}

```

Beispiel 2: Setzt das Passwort für dienstspezifische Anmeldeinformationen zurück, die einem bestimmten Benutzer zugewiesen sind

Im folgenden `reset-service-specific-credential` Beispiel wird ein neues kryptografisch sicheres Passwort für dienstspezifische Anmeldeinformationen generiert, die dem angegebenen Benutzer zugewiesen sind.

```
aws iam reset-service-specific-credential \
  --user-name sofia \
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Ausgabe:

```
{
  "ServiceSpecificCredential": {
    "CreateDate": "2019-04-18T20:45:36+00:00",
    "ServiceName": "codecommit.amazonaws.com",
    "ServiceUserName": "sofia-at-123456789012",
    "ServicePassword": "+oaFsNk7tLco+C/obP9Ghhc0zGcK0ayTmE3LnAmAmH4=",
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
    "UserName": "sofia",
    "Status": "Active"
  }
}
```

Weitere Informationen finden Sie CodeCommit im AWS CodeCommit Benutzerhandbuch unter [Erstellen von Git-Anmeldeinformationen für HTTPS Verbindungen zu](#).

- API-Einheiten finden Sie [ResetServiceSpecificCredential](#) in der AWS CLI Befehlsreferenz.

resync-mfa-device

Das folgende Codebeispiel zeigt die Verwendung `resync-mfa-device`.

AWS CLI

Um ein MFA Gerät zu synchronisieren

Im folgenden `resync-mfa-device` Beispiel wird das MFA Gerät synchronisiert, das dem IAM Benutzer zugeordnet ist Bob und dessen Gerät `arn:aws:iam::123456789012:mfa/`

BobsMFADevice mit einem Authentifizierungsprogramm betrieben ARN wird, das die beiden Authentifizierungs-codes bereitgestellt hat.

```
aws iam resync-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 987654
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [im AWS Benutzerhandbuch unter Verwenden der Multi-Faktor-Authentifizierung \(MFA\)](#).AWS IAM

- API-Einheiten finden Sie [ResyncMfaDevice](#) in der AWS CLI Befehlsreferenz.

set-default-policy-version

Das folgende Codebeispiel zeigt die Verwendung `set-default-policy-version`.

AWS CLI

Um die angegebene Version der angegebenen Richtlinie als Standardversion der Richtlinie festzulegen.

In diesem Beispiel wird die v2 Version der Richtlinie festgelegt, bei der ARN es `arn:aws:iam::123456789012:policy/MyPolicy` sich um die aktive Standardversion handelt.

```
aws iam set-default-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Richtlinien und Berechtigungen](#).

- API-Einheiten finden Sie [SetDefaultPolicyVersion](#) in der AWS CLI Befehlsreferenz.

set-security-token-service-preferences

Das folgende Codebeispiel zeigt die Verwendung `set-security-token-service-preferences`.

AWS CLI

Um die globale Endpunkt-Token-Version festzulegen

Im folgenden `set-security-token-service-preferences` Beispiel wird Amazon so konfiguriert STS, dass bei der Authentifizierung am globalen Endpunkt Tokens der Version 2 verwendet werden.

```
aws iam set-security-token-service-preferences \  
  --global-endpoint-token-version v2Token
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Verwaltung AWS STS in einer AWS Region](#).

- API Einzelheiten finden Sie [SetSecurityTokenServicePreferences](#) in der AWS CLI Befehlsreferenz.

`simulate-custom-policy`

Das folgende Codebeispiel zeigt die Verwendung `simulate-custom-policy`.

AWS CLI

Beispiel 1: Um die Auswirkungen aller IAM Richtlinien zu simulieren, die einem IAM Benutzer oder einer Rolle zugeordnet sind

Im Folgenden wird `simulate-custom-policy` gezeigt, wie Sie sowohl die Richtlinie als auch die Definition von Variablenwerten angeben und einen API Aufruf simulieren, um festzustellen, ob er zulässig oder verweigert ist. Das folgende Beispiel zeigt eine Richtlinie, die den Datenbankzugriff erst nach einem bestimmten Datum und einer bestimmten Uhrzeit ermöglicht. Die Simulation ist erfolgreich, weil die simulierten Aktionen und die angegebene `aws:CurrentTime` Variable alle den Anforderungen der Richtlinie entsprechen.

```
aws iam simulate-custom-policy \  
  --policy-input-list '{"Version":"2012-10-17","Statement":  
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":  
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}' \  
  --action-names dynamodb>CreateBackup \  
  \
```

```
--context-  
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2019-04-25T11:00:00Z',ContextKey
```

Ausgabe:

```
{  
  "EvaluationResults": [  
    {  
      "EvalActionName": "dynamodb:CreateBackup",  
      "EvalResourceName": "*",  
      "EvalDecision": "allowed",  
      "MatchedStatements": [  
        {  
          "SourcePolicyId": "PolicyInputList.1",  
          "StartPosition": {  
            "Line": 1,  
            "Column": 38  
          },  
          "EndPosition": {  
            "Line": 1,  
            "Column": 167  
          }  
        }  
      ],  
      "MissingContextValues": []  
    }  
  ]  
}
```

Beispiel 2: Um einen Befehl zu simulieren, der durch die Richtlinie verboten ist

Das folgende `simulate-custom-policy` Beispiel zeigt die Ergebnisse der Simulation eines Befehls, der durch die Richtlinie verboten ist. In diesem Beispiel liegt das angegebene Datum vor dem Datum, das gemäß der Richtlinienbedingung erforderlich ist.

```
aws iam simulate-custom-policy \  
  --policy-input-list '{"Version":"2012-10-17","Statement":  
{"Effect": "Allow", "Action": "dynamodb:*", "Resource": "*", "Condition":  
{"DateGreaterThan": {"aws:CurrentTime": "2018-08-16T12:00:00Z"}  
}}}' \  
  --action-names dynamodb:CreateBackup \  
  --context-  
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2014-04-25T11:00:00Z',ContextKey
```


Ausgabe:

```
{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb:CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "implicitDeny",
      "MatchedStatements": [],
      "MissingContextValues": []
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Testen von IAM IAM Richtlinien mit dem Richtlinien Simulator](#).

- API Einheiten finden Sie [SimulateCustomPolicy](#) in der AWS CLI Befehlsreferenz.

simulate-principal-policy

Das folgende Codebeispiel zeigt die Verwendung `simulate-principal-policy`.

AWS CLI

Beispiel 1: Um die Auswirkungen einer willkürlichen IAM Richtlinie zu simulieren

Im Folgenden `simulate-principal-policy` wird gezeigt, wie ein Benutzer simuliert, der eine API Aktion aufruft und ermittelt, ob die diesem Benutzer zugewiesenen Richtlinien die Aktion zulassen oder verweigern. Im folgenden Beispiel verfügt der Benutzer über eine Richtlinie, die nur die `codecommit:ListRepositories` Aktion zulässt.

```
aws iam simulate-principal-policy \
  --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
  --action-names codecommit:ListRepositories
```

Ausgabe:

```
{
  "EvaluationResults": [
    {
```

```

    "EvalActionName": "codecommit:ListRepositories",
    "EvalResourceName": "*",
    "EvalDecision": "allowed",
    "MatchedStatements": [
      {
        "SourcePolicyId": "Grant-Access-To-CodeCommit-ListRepo",
        "StartPosition": {
          "Line": 3,
          "Column": 19
        },
        "EndPosition": {
          "Line": 9,
          "Column": 10
        }
      }
    ],
    "MissingContextValues": []
  }
]
}

```

Beispiel 2: Um die Auswirkungen eines verbotenen Befehls zu simulieren

Das folgende `simulate-custom-policy` Beispiel zeigt die Ergebnisse der Simulation eines Befehls, der durch eine der Benutzerrichtlinien verboten ist. Im folgenden Beispiel verfügt der Benutzer über eine Richtlinie, die den Zugriff auf eine DynamoDB-Datenbank erst nach einem bestimmten Datum und einer bestimmten Uhrzeit erlaubt. Bei der Simulation versucht der Benutzer, mit einem `aws:CurrentTime` Wert auf die Datenbank zuzugreifen, der vor der Bedingung der Richtlinie liegt.

```

aws iam simulate-principal-policy \
  --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
  --action-names dynamodb:CreateBackup \
  --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2018-04-25T11:00:00Z',ContextKey

```

Ausgabe:

```

{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb:CreateBackup",

```

```
        "EvalResourceName": "*",
        "EvalDecision": "implicitDeny",
        "MatchedStatements": [],
        "MissingContextValues": []
    }
]
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Testen von IAM IAM Richtlinien mit dem Richtlinien Simulator](#).

- APIEinzelheiten finden Sie [SimulatePrincipalPolicy](#) in der AWS CLI Befehlsreferenz.

tag-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `tag-instance-profile`.

AWS CLI

Um einem Instanzprofil ein Tag hinzuzufügen

Der folgende `tag-instance-profile` Befehl fügt dem angegebenen Instanzprofil ein Tag mit einem Abteilungsnamen hinzu.

```
aws iam tag-instance-profile \  
  --instance-profile-name deployment-role \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [TagInstanceProfile](#) in der AWS CLI Befehlsreferenz.

tag-mfa-device

Das folgende Codebeispiel zeigt die Verwendung `tag-mfa-device`.

AWS CLI

Um einem MFA Gerät ein Tag hinzuzufügen

Mit dem folgenden `tag-mfa-device` Befehl wird dem angegebenen MFA Gerät ein Tag mit einem Abteilungsnamen hinzugefügt.

```
aws iam tag-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen](#) kennzeichnen.

- APIEinzelheiten finden Sie [TagMfaDevice](#) in der AWS CLI Befehlsreferenz.

tag-open-id-connect-provider

Das folgende Codebeispiel zeigt die Verwendung `tag-open-id-connect-provider`.

AWS CLI

Um einem OpenID Connect (OIDC) -kompatiblen Identitätsanbieter ein Tag hinzuzufügen

Der folgende `tag-open-id-connect-provider` Befehl fügt dem angegebenen OIDC Identitätsanbieter ein Tag mit einem Abteilungsnamen hinzu.

```
aws iam tag-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen](#) kennzeichnen.

- APIEinzelheiten finden Sie [TagOpenIdConnectProvider](#) in der AWS CLI Befehlsreferenz.

tag-policy

Das folgende Codebeispiel zeigt die Verwendung `tag-policy`.

AWS CLI

Um einer vom Kunden verwalteten Richtlinie ein Tag hinzuzufügen

Mit dem folgenden `tag-policy` Befehl wird der angegebenen, vom Kunden verwalteten Richtlinie ein Tag mit einem Abteilungsnamen hinzugefügt.

```
aws iam tag-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/billing-access \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen](#) kennzeichnen.

- APIEinzelheiten finden Sie [TagPolicy](#) in der AWS CLI Befehlsreferenz.

tag-role

Das folgende Codebeispiel zeigt die Verwendung `tag-role`.

AWS CLI

Um einer Rolle ein Tag hinzuzufügen

Der folgende `tag-role` Befehl fügt der angegebenen Rolle ein Tag mit einem Abteilungsnamen hinzu.

```
aws iam tag-role --role-name my-role \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch [unter IAM Ressourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [TagRole](#) in der AWS CLI Befehlsreferenz.

tag-saml-provider

Das folgende Codebeispiel zeigt die Verwendung `tag-saml-provider`.

AWS CLI

Um einem SAML Anbieter ein Tag hinzuzufügen

Mit dem folgenden `tag-saml-provider` Befehl wird dem angegebenen SAML Anbieter ein Tag mit einem Abteilungsnamen hinzugefügt.

```
aws iam tag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch [unter IAM Ressourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [TagSamlProvider](#) in der AWS CLI Befehlsreferenz.

tag-server-certificate

Das folgende Codebeispiel zeigt die Verwendung `tag-server-certificate`.

AWS CLI

Um einem Serverzertifikat ein Tag hinzuzufügen

Der folgende `tag-saml-provider` Befehl fügt dem angegebenen Serverzertifikat ein Tag mit einem Abteilungsnamen hinzu.

```
aws iam tag-server-certificate \  
  --server-certificate-name ExampleCertificate \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [AWS IAMBenutzerhandbuch unter IAM Ressourcen](#) kennzeichnen.

- APIEinzelheiten finden Sie [TagServerCertificate](#) in der AWS CLI Befehlsreferenz.

tag-user

Das folgende Codebeispiel zeigt die Verwendung `tag-user`.

AWS CLI

Um einem Benutzer ein Tag hinzuzufügen

Mit dem folgenden `tag-user` Befehl wird dem angegebenen Benutzer ein Tag mit der zugehörigen Abteilung hinzugefügt.

```
aws iam tag-user \  
  --user-name alice \  
  --tags '{"Key": "Department", "Value": "Accounting"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [TagUser](#) in der AWS CLI Befehlsreferenz.

untag-instance-profile

Das folgende Codebeispiel zeigt die Verwendung `untag-instance-profile`.

AWS CLI

Um ein Tag aus einem Instanzprofil zu entfernen

Mit dem folgenden `untag-instance-profile` Befehl werden alle Tags mit dem Schlüsselnamen „Department“ aus dem angegebenen Instanzprofil entfernt.

```
aws iam untag-instance-profile \  
  --instance-profile-name deployment-role \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [UntagInstanceProfile](#) in der AWS CLI Befehlsreferenz.

untag-mfa-device

Das folgende Codebeispiel zeigt die Verwendung `untag-mfa-device`.

AWS CLI

Um ein Tag von einem MFA Gerät zu entfernen

Mit dem folgenden `untag-mfa-device` Befehl werden alle Tags mit dem Schlüsselnamen „Abteilung“ vom angegebenen MFA Gerät entfernt.

```
aws iam untag-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [UntagMfaDevice](#) in der AWS CLI Befehlsreferenz.

untag-open-id-connect-provider

Das folgende Codebeispiel zeigt die Verwendung `untag-open-id-connect-provider`.

AWS CLI

Um ein Tag von einem OIDC Identitätsanbieter zu entfernen

Mit dem folgenden `untag-open-id-connect-provider` Befehl werden alle Tags mit dem Schlüsselnamen „Abteilung“ aus dem angegebenen OIDC Identitätsanbieter entfernt.

```
aws iam untag-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [UntagOpenIdConnectProvider](#) in der AWS CLI Befehlsreferenz.

untag-policy

Das folgende Codebeispiel zeigt die Verwendung `untag-policy`.

AWS CLI

Um ein Tag aus einer vom Kunden verwalteten Richtlinie zu entfernen

Mit dem folgenden `untag-policy` Befehl werden alle Tags mit dem Schlüsselnamen „Abteilung“ aus der angegebenen, vom Kunden verwalteten Richtlinie entfernt.

```
aws iam untag-policy \  
  --policy-arn arn:aws:iam::452925170507:policy/billing-access \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch [unter IAM Ressourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [UntagPolicy](#) in der AWS CLI Befehlsreferenz.

untag-role

Das folgende Codebeispiel zeigt die Verwendung `untag-role`.

AWS CLI

Um ein Tag aus einer Rolle zu entfernen

Mit dem folgenden `untag-role` Befehl werden alle Tags mit dem Schlüsselnamen „Department“ aus der angegebenen Rolle entfernt.

```
aws iam untag-role \  
  --role-name my-role \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [UntagRole](#) in der AWS CLI Befehlsreferenz.

untag-saml-provider

Das folgende Codebeispiel zeigt die Verwendung `untag-saml-provider`.

AWS CLI

Um ein Tag von einem SAML Anbieter zu entfernen

Mit dem folgenden `untag-saml-provider` Befehl werden alle Tags mit dem Schlüsselnamen „Department“ aus dem angegebenen Instanzprofil entfernt.

```
aws iam untag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [UntagSamlProvider](#) in der AWS CLI Befehlsreferenz.

untag-server-certificate

Das folgende Codebeispiel zeigt die Verwendung `untag-server-certificate`.

AWS CLI

Um ein Tag aus einem Serverzertifikat zu entfernen

Mit dem folgenden `untag-server-certificate` Befehl werden alle Tags mit dem Schlüsselnamen „Department“ aus dem angegebenen Serverzertifikat entfernt.

```
aws iam untag-server-certificate \  
  --server-certificate-name ExampleCertificate \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch [unter IAM Ressourcen mit Tags](#) versehen.

- APIEinzelheiten finden Sie [UntagServerCertificate](#) in der AWS CLI Befehlsreferenz.

untag-user

Das folgende Codebeispiel zeigt die Verwendung `untag-user`.

AWS CLI

Um ein Tag von einem Benutzer zu entfernen

Mit dem folgenden `untag-user` Befehl werden alle Tags mit dem Schlüsselnamen „Department“ vom angegebenen Benutzer entfernt.

```
aws iam untag-user \  
  --user-name alice \  
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRessourcen taggen](#).

- APIEinzelheiten finden Sie [UntagUser](#) in der AWS CLI Befehlsreferenz.

update-access-key

Das folgende Codebeispiel zeigt die Verwendung `update-access-key`.

AWS CLI

Um einen Zugriffsschlüssel für einen IAM Benutzer zu aktivieren oder zu deaktivieren

Der folgende `update-access-key` Befehl deaktiviert den angegebenen Zugriffsschlüssel (Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel) für den genannten IAM Bob Benutzer.

```
aws iam update-access-key \  
  --access-key-id AKIAIOSFODNN7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Die Deaktivierung des Schlüssels bedeutet, dass er nicht für den programmatischen Zugriff auf verwendet werden kann. AWS Der Schlüssel ist jedoch weiterhin verfügbar und kann erneut aktiviert werden.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Verwaltung von Zugriffstasten für IAM AWS IAMBenutzer](#).

- API-Einheiten finden Sie [UpdateAccessKey](#) in der AWS CLI Befehlsreferenz.

update-account-password-policy

Das folgende Codebeispiel zeigt die Verwendung `update-account-password-policy`.

AWS CLI

Um die Passwortrichtlinie für das aktuelle Konto festzulegen oder zu ändern

Mit dem folgenden `update-account-password-policy` Befehl wird für die Kennwortrichtlinie eine Mindestlänge von acht Zeichen und eine oder mehrere Zahlen im Kennwort festgelegt.

```
aws iam update-account-password-policy \  
  --minimum-password-length 8 \  
  --require-numbers
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Änderungen an der Kennwortrichtlinie eines Kontos wirken sich auf alle neuen Passwörter aus, die für IAM Benutzer des Kontos erstellt werden. Änderungen der Passwortrichtlinie wirken sich nicht auf bestehende Passwörter aus.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Einrichten einer Kontokennwortrichtlinie für IAM AWS IAM Benutzer](#).

- API-Einheiten finden Sie [UpdateAccountPasswordPolicy](#) in der AWS CLI Befehlsreferenz.

update-assume-role-policy

Das folgende Codebeispiel zeigt die Verwendung `update-assume-role-policy`.

AWS CLI

Um die Vertrauensrichtlinie für eine IAM Rolle zu aktualisieren

Der folgende `update-assume-role-policy` Befehl aktualisiert die Vertrauensrichtlinie für die angegebene Rolle `Test-Role`.

```
aws iam update-assume-role-policy \  
  --role-name Test-Role
```

```
--role-name Test-Role \  
--policy-document file://Test-Role-Trust-Policy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Die Vertrauensrichtlinie ist als JSON Dokument in der Datei `Test-role-trust-policy.json` definiert. (Der Dateiname und die Erweiterung sind nicht von Bedeutung.) Die Vertrauensrichtlinie muss einen Prinzipal angeben.

Verwenden Sie den Befehl, um die Berechtigungsrichtlinie für eine Rolle zu aktualisieren. `put-role-policy`

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [IAMRollen erstellen](#).

- APIEinzelheiten finden Sie [UpdateAssumeRolePolicy](#) in der AWS CLI Befehlsreferenz.

update-group

Das folgende Codebeispiel zeigt die Verwendung `update-group`.

AWS CLI

Um eine IAM Gruppe umzubenennen

Mit dem folgenden `update-group` Befehl wird der Name der IAM Gruppe `Test` in `geändertTest-1` geändert.

```
aws iam update-group \  
--group-name Test \  
--new-group-name Test-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Umbenennen einer AWS IAMBenutzergruppe](#).

- APIEinzelheiten finden Sie [UpdateGroup](#) in der AWS CLI Befehlsreferenz.

update-login-profile

Das folgende Codebeispiel zeigt die Verwendung `update-login-profile`.

AWS CLI

Um das Passwort für einen IAM Benutzer zu aktualisieren

Der folgende `update-login-profile` Befehl erstellt ein neues Passwort für den IAM Benutzer mit dem Namen Bob.

```
aws iam update-login-profile \  
  --user-name Bob \  
  --password <password>
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Verwenden Sie den `update-account-password-policy` Befehl, um eine Kennwortrichtlinie für das Konto festzulegen. Wenn das neue Passwort gegen die Passwortrichtlinie für das Konto verstößt, gibt der Befehl einen `PasswordPolicyViolation` Fehler zurück.

Wenn die Kontokennwortrichtlinie dies zulässt, können IAM Benutzer ihre eigenen Passwörter mithilfe des `change-password` Befehls ändern.

Bewahren Sie das Passwort an einem sicheren Ort auf. Wenn das Passwort verloren geht, kann es nicht wiederhergestellt werden, und Sie müssen mit dem `create-login-profile` Befehl ein neues erstellen.

Weitere Informationen finden Sie im Benutzerhandbuch unter [Passwörter für IAM AWS IAM Benutzer verwalten](#).

- API-Einheiten finden Sie [UpdateLoginProfile](#) in der AWS CLI Befehlsreferenz.

update-open-id-connect-provider-thumbprint

Das folgende Codebeispiel zeigt die Verwendung `update-open-id-connect-provider-thumbprint`.

AWS CLI

Um die bestehende Liste der Fingerabdrücke von Serverzertifikaten durch eine neue Liste zu ersetzen

In diesem Beispiel wird die Liste der Fingerabdrücke für Zertifikate für den OIDC Anbieter aktualisiert, der einen neuen Fingerabdruck verwenden ARN `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` soll.

```
aws iam update-open-id-connect-provider-thumbprint \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
example.oidcprovider.com \  
  --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen von OpenID Connect \(OIDC\) -Identitätsanbietern](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie unter [UpdateOpenIdConnectProviderThumbprint AWS CLI](#)Befehlsreferenz.

update-role-description

Das folgende Codebeispiel zeigt die Verwendungupdate-role-description.

AWS CLI

Um die Beschreibung einer IAM Rolle zu ändern

Mit dem folgenden update-role Befehl wird die Beschreibung der IAM Rolle production-role in geändertMain production role.

```
aws iam update-role-description \  
  --role-name production-role \  
  --description 'Main production role'
```

Ausgabe:

```
{  
  "Role": {  
    "Path": "/",  
    "RoleName": "production-role",  
    "RoleId": "AROA1234567890EXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:role/production-role",  
    "CreateDate": "2017-12-06T17:16:37+00:00",  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Effect": "Allow",
```

```
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
        },
        "Action": "sts:AssumeRole",
        "Condition": {}
    }
]
},
"Description": "Main production role"
}
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ändern einer Rolle](#).

- APIEinzelheiten finden Sie [UpdateRoleDescription](#) in der AWS CLI Befehlsreferenz.

update-role

Das folgende Codebeispiel zeigt die Verwendung `update-role`.

AWS CLI

Um die Beschreibung einer IAM Rolle oder die Sitzungsdauer zu ändern

Mit dem folgenden `update-role` Befehl wird die Beschreibung der IAM Rolle `production-role` auf 12 Stunden geändert `Main production role` und die maximale Sitzungsdauer wird auf 12 Stunden festgelegt.

```
aws iam update-role \
  --role-name production-role \
  --description 'Main production role' \
  --max-session-duration 43200
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ändern einer Rolle](#).

- APIEinzelheiten finden Sie [UpdateRole](#) in der AWS CLI Befehlsreferenz.

update-saml-provider

Das folgende Codebeispiel zeigt die Verwendung `update-saml-provider`.

AWS CLI

Um das Metadatendokument für einen vorhandenen SAML Anbieter zu aktualisieren

In diesem Beispiel wird der SAML Anbieter IAM, in dessen Verzeichnis ARN sich befindet, `arn:aws:iam::123456789012:saml-provider/SAMLADFS` mit einem neuen SAML Metadatendokument aus der Datei `aktualisiertSAMLMetaData.xml`.

```
aws iam update-saml-provider \  
  --saml-metadata-document file://SAMLMetaData.xml \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Ausgabe:

```
{  
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"  
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Creating IAM SAML Identity Providers](#).

- API Einzelheiten finden Sie [UpdateSamlProvider](#) unter AWS CLI Befehlsreferenz.

update-server-certificate

Das folgende Codebeispiel zeigt die Verwendung `update-server-certificate`.

AWS CLI

Um den Pfad oder Namen eines Serverzertifikats in Ihrem AWS Konto zu ändern

Mit dem folgenden `update-server-certificate`-Befehl wird der Name des Zertifikats von `myServerCertificate` in `myUpdatedServerCertificate` geändert. Außerdem wird der Pfad geändert, `/cloudfront/` sodass der CloudFront Amazon-Service darauf zugreifen kann. Mit diesem Befehl wird keine Ausgabe zurückgegeben. Sie können die Ergebnisse der Aktualisierung anzeigen, indem Sie den `list-server-certificates`-Befehl ausführen.

```
aws-iam update-server-certificate \  
  --server-certificate-name myServerCertificate \  
  --new-server-certificate-name myUpdatedServerCertificate \  
  --path /cloudfront/
```

```
--new-path /cloudfront/
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [IAM im AWS IAM Benutzerhandbuch unter Serverzertifikate verwalten](#).

- API-Einheiten finden Sie [UpdateServerCertificate](#) in der AWS CLI Befehlsreferenz.

update-service-specific-credential

Das folgende Codebeispiel zeigt die Verwendung `update-service-specific-credential`.

AWS CLI

Beispiel 1: Um den Status der dienstspezifischen Anmeldeinformationen des anfragenden Benutzers zu aktualisieren

Im folgenden `update-service-specific-credential` Beispiel wird der Status der angegebenen Anmeldeinformationen für den Benutzer geändert, an den die Anfrage gestellt wird. `Inactive`

```
aws iam update-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um den Status der dienstspezifischen Anmeldeinformationen eines angegebenen Benutzers zu aktualisieren

Im folgenden `update-service-specific-credential` Beispiel wird der Status der Anmeldeinformationen des angegebenen Benutzers in Inaktiv geändert.

```
aws iam update-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Create Git Credentials for HTTPS Connections to CodeCommit](#) im AWS CodeCommit Benutzerhandbuch

- API-Einheiten finden Sie [UpdateServiceSpecificCredential](#) in der AWS CLI Befehlsreferenz.

update-signing-certificate

Das folgende Codebeispiel zeigt die Verwendung `update-signing-certificate`.

AWS CLI

Um ein Signaturzertifikat für einen IAM Benutzer zu aktivieren oder zu deaktivieren

Der folgende `update-signing-certificate` Befehl deaktiviert das angegebene Signaturzertifikat für den genannten IAM Bob Benutzer.

```
aws iam update-signing-certificate \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

Verwenden Sie den `list-signing-certificates` Befehl, um die ID für ein Signaturzertifikat abzurufen.

Weitere Informationen finden Sie unter [Signaturzertifikate verwalten](#) im EC2 Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSigningCertificate](#) in der AWS CLI Befehlsreferenz.

update-ssh-public-key

Das folgende Codebeispiel zeigt die Verwendung `update-ssh-public-key`.

AWS CLI

Um den Status eines SSH öffentlichen Schlüssels zu ändern

Der folgende `update-ssh-public-key` Befehl ändert den Status des angegebenen öffentlichen Schlüssels in `Inactive`.

```
aws iam update-ssh-public-key \  
  --user-name sofia \  
  --public-key
```

```
--ssh-public-key-id APKA1234567890EXAMPLE \  
--status Inactive
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von SSH Schlüsseln und SSH mit CodeCommit](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [UpdateSshPublicKey](#) in der AWS CLI Befehlsreferenz.

update-user

Das folgende Codebeispiel zeigt die Verwendung `update-user`.

AWS CLI

Um den Namen eines IAM Benutzers zu ändern

Mit dem folgenden `update-user` Befehl wird der Name des IAM Benutzers Bob in geändertRobert.

```
aws iam update-user \  
--user-name Bob \  
--new-user-name Robert
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im [IAMBenutzerhandbuch unter Umbenennen einer AWS IAMBenutzergruppe](#).

- APIEinzelheiten finden Sie [UpdateUser](#) in der AWS CLI Befehlsreferenz.

upload-server-certificate

Das folgende Codebeispiel zeigt die Verwendung `upload-server-certificate`.

AWS CLI

Um ein Serverzertifikat auf Ihr AWS Konto hochzuladen

Mit dem folgenden `upload-server-certificate` Befehl wird ein Serverzertifikat auf Ihr AWS Konto hochgeladen. In diesem Beispiel befindet sich das Zertifikat in der Datei

`public_key_cert_file.pem`, der zugehörige private Schlüssel in der Datei `my_private_key.pem` und die von der Zertifizierungsstelle (CA) bereitgestellte Zertifikatskette befindet sich in der `my_certificate_chain_file.pem`-Datei. Wenn der Upload der Datei abgeschlossen ist, ist sie unter dem Namen verfügbar. `myServerCertificate` Parameter, die mit `file://` beginnen, weisen den Befehl an, den Inhalt der Datei zu lesen und diesen als Parameterwert anstelle des Dateinamens selbst zu verwenden.

```
aws iam upload-server-certificate \  
  --server-certificate-name myServerCertificate \  
  --certificate-body file://public_key_cert_file.pem \  
  --private-key file://my_private_key.pem \  
  --certificate-chain file://my_certificate_chain_file.pem
```

Ausgabe:

```
{  
  "ServerCertificateMetadata": {  
    "Path": "/",  
    "ServerCertificateName": "myServerCertificate",  
    "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",  
    "Arn": "arn:aws:iam:1234567989012:server-certificate/myServerCertificate",  
    "UploadDate": "2019-04-22T21:13:44+00:00",  
    "Expiration": "2019-10-15T22:23:16+00:00"  
  }  
}
```

Weitere Informationen finden Sie im Nutzerhandbuch unter [Serverzertifikate erstellen, hochladen und löschen](#). IAM

- API-Einheiten finden Sie unter [UploadServerCertificate AWS CLI Befehlsreferenz](#).

upload-signing-certificate

Das folgende Codebeispiel zeigt die Verwendung `upload-signing-certificate`.

AWS CLI

Um ein Signaturzertifikat für einen IAM Benutzer hochzuladen

Mit dem folgenden `upload-signing-certificate` Befehl wird ein Signaturzertifikat für den genannten IAM Bob Benutzer hochgeladen.

```
aws iam upload-signing-certificate \  
  --user-name Bob \  
  --certificate-body file://certificate.pem
```

Ausgabe:

```
{  
  "Certificate": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END  
CERTIFICATE-----",  
    "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",  
    "UploadDate": "2013-06-06T21:40:08.121Z"  
  }  
}
```

Das Zertifikat befindet sich in einer Datei mit dem Namen `certificate.pem` im Format. PEM

Weitere Informationen finden Sie im Nutzerhandbuch unter [Benutzersignaturzertifikate erstellen und hochladen. IAM](#)

- API-Einheiten finden Sie unter [UploadSigningCertificate AWS CLI Befehlsreferenz](#).

upload-ssh-public-key

Das folgende Codebeispiel zeigt die Verwendung `upload-ssh-public-key`.

AWS CLI

Um einen SSH öffentlichen Schlüssel hochzuladen und ihn einem Benutzer zuzuordnen

Mit dem folgenden `upload-ssh-public-key` Befehl wird der in der Datei gefundene öffentliche Schlüssel hochgeladen `sshkey.pub` und an den Benutzer angehängt. `sofia`

```
aws iam upload-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-body file://sshkey.pub
```

Ausgabe:

```
{
  "SSHPublicKey": {
    "UserName": "sofia",
    "SSHPublicKeyId": "APKA1234567890EXAMPLE",
    "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",
    "SSHPublicKeyBody": "ssh-rsa <<long string generated by ssh-keygen
command>>",
    "Status": "Active",
    "UploadDate": "2019-04-18T17:04:49+00:00"
  }
}
```

Weitere Informationen zum Generieren von Schlüsseln in einem für diesen Befehl geeigneten Format finden Sie unter [SSHLinux, macOS oder Unix: Richten Sie die öffentlichen und privaten Schlüssel für Git ein und CodeCommit](#) oder [SSHund Windows: Richten Sie die öffentlichen und privaten Schlüssel für Git ein und CodeCommit](#) im AWS CodeCommit Benutzerhandbuch.

- APIEinzelheiten finden Sie [UploadSshPublicKey](#) in der AWS CLI Befehlsreferenz.

IAMAccess Analyzer-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit IAM Access Analyzer Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

apply-archive-rule

Das folgende Codebeispiel zeigt die Verwendung `apply-archive-rule`.

AWS CLI

Um eine Archivierungsregel auf vorhandene Ergebnisse anzuwenden, die die Kriterien der Archivierungsregel erfüllen

Im folgenden `apply-archive-rule` Beispiel wird eine Archivierungsregel auf vorhandene Ergebnisse angewendet, die die Archivregelkriterien erfüllen.

```
aws accessanalyzer apply-archive-rule \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Archivierungsregeln](#).

- APIEinzelheiten finden Sie [ApplyArchiveRule](#) in der AWS CLI Befehlsreferenz.

cancel-policy-generation

Das folgende Codebeispiel zeigt die Verwendung `cancel-policy-generation`.

AWS CLI

Um die angeforderte Richtliniengenerierung abubrechen

Im folgenden `cancel-policy-generation` Beispiel wird die angeforderte Job-ID für die Richtliniengenerierung storniert.

```
aws accessanalyzer cancel-policy-generation \  
  --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Generierung von IAM Access Analyzer-Richtlinien](#).

- APIEinzelheiten finden Sie [CancelPolicyGeneration](#) in der AWS CLI Befehlsreferenz.

check-access-not-granted

Das folgende Codebeispiel zeigt die Verwendung `check-access-not-granted`.

AWS CLI

Um zu überprüfen, ob der angegebene Zugriff durch eine Richtlinie nicht zulässig ist

Im folgenden `check-access-not-granted` Beispiel wird geprüft, ob der angegebene Zugriff durch eine Richtlinie nicht zulässig ist.

```
aws accessanalyzer check-access-not-granted \  
  --policy-document file://myfile.json \  
  --access actions="s3:DeleteBucket","s3:GetBucketLocation" \  
  --policy-type IDENTITY_POLICY
```

Inhalt von `myfile.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ]  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "result": "PASS",  
  "message": "The policy document does not grant access to perform one or more of  
the listed actions."  
}
```

Weitere Informationen finden Sie APIs im AWS IAMBenutzerhandbuch unter [Vorschau des IAM Zugriffs mit Access Analyzer](#).

- API-Einheiten finden Sie [CheckAccessNotGranted](#) in der AWS CLI Befehlsreferenz.

check-no-new-access

Das folgende Codebeispiel zeigt die Verwendung `check-no-new-access`.

AWS CLI

Um zu überprüfen, ob für eine aktualisierte Richtlinie im Vergleich zur vorhandenen Richtlinie neuer Zugriff zulässig ist

Im folgenden `check-no-new-access` Beispiel wird geprüft, ob für eine aktualisierte Richtlinie im Vergleich zur vorhandenen Richtlinie neuer Zugriff zulässig ist.

```
aws accessanalyzer check-no-new-access \  
  --existing-policy-document file://existing-policy.json \  
  --new-policy-document file://new-policy.json \  
  --policy-type IDENTITY_POLICY
```

Inhalt von `existing-policy.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ]  
    }  
  ]  
}
```

Inhalt von `new-policy.json`:

```
{  
  "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectAcl",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
  }
]
```

Ausgabe:

```
{
  "result": "FAIL",
  "message": "The modified permissions grant new access compared to your existing policy.",
  "reasons": [
    {
      "description": "New access in the statement with index: 0.",
      "statementIndex": 0
    }
  ]
}
```

Weitere Informationen finden Sie unter [Vorschau des Zugriffs mit IAM Access Analyzer APIs](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [CheckNoNewAccess](#) in der AWS CLI Befehlsreferenz.

check-no-public-access

Das folgende Codebeispiel zeigt die Verwendung `check-no-public-access`.

AWS CLI

Um zu überprüfen, ob eine Ressourcenrichtlinie öffentlichen Zugriff auf den angegebenen Ressourcentyp gewähren kann

Im folgenden `check-no-public-access` Beispiel wird geprüft, ob eine Ressourcenrichtlinie öffentlichen Zugriff auf den angegebenen Ressourcentyp gewähren kann.

```
aws accessanalyzer check-no-public-access \  
  --policy-document file://check-no-public-access-myfile.json \  
  --resource-type AWS::S3::Bucket
```

Inhalt von `myfile.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "CheckNoPublicAccess",  
      "Effect": "Allow",  
      "Principal": { "AWS": "arn:aws:iam::111122223333:user/JohnDoe" },  
      "Action": [  
        "s3:GetObject"  
      ]  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "result": "PASS",  
  "message": "The resource policy does not grant public access for the given  
resource type."  
}
```

Weitere Informationen finden Sie APIs im AWS IAMBenutzerhandbuch unter [Vorschau des IAM Zugriffs mit Access Analyzer](#).

- APIEinzelheiten finden Sie [CheckNoPublicAccess](#) in der AWS CLI Befehlsreferenz.

create-access-preview

Das folgende Codebeispiel zeigt die Verwendung `create-access-preview`.

AWS CLI

Um eine Zugriffsvorschau zu erstellen, mit der Sie eine Vorschau der IAM Access Analyzer-Ergebnisse für Ihre Ressource anzeigen können, bevor Sie Ressourcenberechtigungen bereitstellen

Im folgenden `create-access-preview` Beispiel wird eine Zugriffsvorschau erstellt, mit der Sie eine Vorschau der IAM Access Analyzer-Ergebnisse für Ihre Ressource anzeigen können, bevor Sie Ressourcenberechtigungen für Ihr AWS Konto bereitstellen.

```
aws accessanalyzer create-access-preview \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --configurations file://myfile.json
```

Inhalt von `myfile.json`:

```
{  
  "arn:aws:s3:::DOC-EXAMPLE-BUCKET": {  
    "s3Bucket": {  
      "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":"  
\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},\"Action\":"  
\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":[\"arn:aws:s3:::DOC-EXAMPLE-  
BUCKET/*\"]}]}",  
      "bucketPublicAccessBlock": {  
        "ignorePublicAcls": true,  
        "restrictPublicBuckets": true  
      },  
      "bucketAclGrants": [  
        {  
          "grantee": {  
            "id":  
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"  
          },  
          "permission": "READ"  
        }  
      ]  
    }  
  }  
}
```

Ausgabe:

```
{
  "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"
}
```

Weitere Informationen finden Sie unter [Vorschau des Zugriffs mit IAM Access Analyzer APIs](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [CreateAccessPreview](#) in der AWS CLI Befehlsreferenz.

create-analyzer

Das folgende Codebeispiel zeigt die Verwendung `create-analyzer`.

AWS CLI

Um einen Analyzer zu erstellen

Das folgende `create-analyzer` Beispiel erstellt einen Analyzer in Ihrem AWS Konto.

```
aws accessanalyzer create-analyzer \
  --analyzer-name example \
  --type ACCOUNT
```

Ausgabe:

```
{
  "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit AWS Identity and Access Management Access Analyzer](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [CreateAnalyzer](#) unter AWS CLI Befehlsreferenz.

create-archive-rule

Das folgende Codebeispiel zeigt die Verwendung `create-archive-rule`.

AWS CLI

Um eine Archivierungsregel für den angegebenen Analyzer zu erstellen

Im folgenden `create-archive-rule` Beispiel wird eine Archivierungsregel für den angegebenen Analyser in Ihrem AWS Konto erstellt.

```
aws accessanalyzer create-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":  
  ["AWS::IAM::Role"]}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Archivierungsregeln](#).

- APIEinzelheiten finden Sie [CreateArchiveRule](#) in der AWS CLI Befehlsreferenz.

delete-analyzer

Das folgende Codebeispiel zeigt die Verwendung `delete-analyzer`.

AWS CLI

Um den angegebenen Analysator zu löschen

Im folgenden `delete-analyzer` Beispiel wird der angegebene Analysator in Ihrem AWS Konto gelöscht.

```
aws accessanalyzer delete-analyzer \  
  --analyzer-name example
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Archivierungsregeln](#).

- APIEinzelheiten finden Sie [DeleteAnalyzer](#) in der AWS CLI Befehlsreferenz.

delete-archive-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-archive-rule`.

AWS CLI

Um die angegebene Archivierungsregel zu löschen

Im folgenden `delete-archive-rule` Beispiel wird die angegebene Archivierungsregel in Ihrem AWS Konto gelöscht.

```
aws accessanalyzer delete-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Archivierungsregeln](#).

- API Einheiten finden Sie [DeleteArchiveRule](#) in der AWS CLI Befehlsreferenz.

get-access-preview

Das folgende Codebeispiel zeigt die Verwendung `get-access-preview`.

AWS CLI

Ruft Informationen über eine Zugriffsvorschau für den angegebenen Analysator ab

Im folgenden `get-access-preview` Beispiel werden Informationen über eine Zugriffsvorschau für den angegebenen Analyzer in Ihrem AWS Konto abgerufen.

```
aws accessanalyzer get-access-preview \  
  --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

Ausgabe:

```
{  
  "accessPreview": {  
    "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",  
    "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account",  
    "configurations": {  
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET": {  
        "s3Bucket": {  
          "bucketPolicy": "{\n\"Version\":\n\"2012-10-17\",\n\"Statement\":  
[{\n\"Effect\":\n\"Allow\",\n\"Principal\":{\n\"AWS\":[\n\"arn:aws:iam::111122223333:root\"  
]}],  
          "}
```



```

{"Action":["s3:PutObject","s3:PutObjectAcl"],"Resource":["arn:aws:s3::DOC-EXAMPLE-BUCKET/*"]},
  "bucketAclGrants": [
    {
      "permission": "READ",
      "grantee": {
        "id":
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
      }
    }
  ],
  "bucketPublicAccessBlock": {
    "ignorePublicAcls": true,
    "restrictPublicBuckets": true
  }
}
},
"createdAt": "2024-02-17T00:18:44+00:00",
"status": "COMPLETED"
}
}

```

Weitere Informationen finden Sie unter [Vorschau des Zugriffs mit IAM Access Analyzer APIs](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [GetAccessPreview](#) in der AWS CLI Befehlsreferenz.

get-analyzed-resource

Das folgende Codebeispiel zeigt die Verwendung `get-analyzed-resource`.

AWS CLI

Um Informationen über eine Ressource abzurufen, die analysiert wurde

Im folgenden `get-analyzed-resource` Beispiel werden Informationen über eine Ressource abgerufen, die in Ihrem AWS Konto analysiert wurde.

```

aws accessanalyzer get-analyzed-resource \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ConsoleAnalyzer-account \
  --resource-arn arn:aws:s3::DOC-EXAMPLE-BUCKET

```

Ausgabe:

```
{
  "resource": {
    "analyzedAt": "2024-02-15T18:01:53.002000+00:00",
    "isPublic": false,
    "resourceArn": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
    "resourceOwnerAccount": "111122223333",
    "resourceType": "AWS::S3::Bucket"
  }
}
```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [GetAnalyzedResource](#) unter AWS CLI Befehlsreferenz.

get-analyzer

Das folgende Codebeispiel zeigt die Verwendung `get-analyzer`.

AWS CLI

Um Informationen über den angegebenen Analysator abzurufen

Im folgenden `get-analyzer` Beispiel werden Informationen über den angegebenen Analysator in Ihrem AWS Konto abgerufen.

```
aws accessanalyzer get-analyzer \
  --analyzer-name ConsoleAnalyzer-account
```

Ausgabe:

```
{
  "analyzer": {
    "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
    "createdAt": "2019-12-03T07:28:17+00:00",
    "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-topic",
    "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
    "name": "ConsoleAnalyzer-account",
    "status": "ACTIVE",
```

```
    "tags": {
      "auto-delete": "no"
    },
    "type": "ACCOUNT"
  }
}
```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API Einzelheiten finden Sie [GetAnalyzer](#) unter AWS CLI Befehlsreferenz.

get-archive-rule

Das folgende Codebeispiel zeigt die Verwendung `get-archive-rule`.

AWS CLI

Um Informationen über eine Archivierungsregel abzurufen

Im folgenden `get-archive-rule` Beispiel werden Informationen zu einer Archivierungsregel in Ihrem AWS Konto abgerufen.

```
aws accessanalyzer get-archive-rule \
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
  --rule-name MyArchiveRule
```

Ausgabe:

```
{
  "archiveRule": {
    "createdAt": "2024-02-15T00:49:27+00:00",
    "filter": {
      "resource": {
        "contains": [
          "Cognito"
        ]
      },
      "resourceType": {
        "eq": [
          "AWS::IAM::Role"
        ]
      }
    }
  }
}
```

```

    }
  },
  "ruleName": "MyArchiveRule",
  "updatedAt": "2024-02-15T00:49:27+00:00"
}
}

```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Archivierungsregeln](#).

- APIEinzelheiten finden Sie [GetArchiveRule](#) in der AWS CLI Befehlsreferenz.

get-finding-v2

Das folgende Codebeispiel zeigt die Verwendung `get-finding-v2`.

AWS CLI

Um Informationen über den angegebenen Befund abzurufen

Im folgenden `get-finding-v2` Beispiel werden Informationen zum angegebenen Ergebnis in Ihrem AWS Konto abgerufen.

```

aws accessanalyzer get-finding-v2 \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  

ConsoleAnalyzer-organization \
  --id 0910eedb-381e-4e95-adda-0d25c19e6e90

```

Ausgabe:

```

{
  "findingDetails": [
    {
      "externalAccessDetails": {
        "action": [
          "sts:AssumeRoleWithWebIdentity"
        ],
        "condition": {
          "cognito-identity.amazonaws.com:aud": "us-  
west-2:EXAMPLE0-0000-0000-0000-000000000000"
        },
        "isPublic": false,
        "principal": {
          "Federated": "cognito-identity.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    }
  }
}
],
"resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
"status": "ACTIVE",
"error": null,
"createdAt": "2021-02-26T21:17:50.905000+00:00",
"resourceType": "AWS::IAM::Role",
"findingType": "ExternalAccess",
"resourceOwnerAccount": "111122223333",
"analyzedAt": "2024-02-16T18:17:47.888000+00:00",
"id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
"updatedAt": "2021-02-26T21:17:50.905000+00:00"
}

```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ergebnisse überprüfen](#).

- APIEinzelheiten finden Sie unter [GetFindingV2](#) in der AWS CLI Befehlsreferenz.

get-finding

Das folgende Codebeispiel zeigt die Verwendung `get-finding`.

AWS CLI

Um Informationen über den angegebenen Befund abzurufen

Im folgenden `get-finding` Beispiel werden Informationen zum angegebenen Ergebnis in Ihrem AWS Konto abgerufen.

```

aws accessanalyzer get-finding \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  

ConsoleAnalyzer-organization \
  --id 0910eedb-381e-4e95-adda-0d25c19e6e90

```

Ausgabe:

```

{
  "finding": {
    "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
    "principal": {
      "Federated": "cognito-identity.amazonaws.com"
    }
  }
}

```

```
    },
    "action": [
      "sts:AssumeRoleWithWebIdentity"
    ],
    "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
    "isPublic": false,
    "resourceType": "AWS::IAM::Role",
    "condition": {
      "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
    },
    "createdAt": "2021-02-26T21:17:50.905000+00:00",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "updatedAt": "2021-02-26T21:17:50.905000+00:00",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333"
  }
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Ergebnisse überprüfen](#).

- APIEinzelheiten finden Sie [GetFinding](#) in der AWS CLI Befehlsreferenz.

get-generated-policy

Das folgende Codebeispiel zeigt die Verwendung `get-generated-policy`.

AWS CLI

Um die Richtlinie abzurufen, die mit dem `StartPolicyGeneration` generiert wurde API

Im folgenden `get-generated-policy` Beispiel wird die Richtlinie abgerufen, die mithilfe von `StartPolicyGeneration` API in Ihrem AWS Konto generiert wurde.

```
aws accessanalyzer get-generated-policy \
  --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

Ausgabe:

```
{
  "generatedPolicyResult": {
    "generatedPolicies": [
      {
```

```

        "policy": "{\\"Version\\":\\"2012-10-17\\",\\"Statement\\":
[{\\"Sid\\":\\"SupportedServiceSid0\\",\\"Effect\\":\\"Allow\\",\\"Action\\":
[\\"access-analyzer:GetAnalyzer\\",\\"access-analyzer:ListAnalyzers\\",
\\"access-analyzer:ListArchiveRules\\",\\"access-analyzer:ListFindings
\\",\\"cloudtrail:DescribeTrails\\",\\"cloudtrail:GetEventDataStore\\",
\\"cloudtrail:GetEventSelectors\\",\\"cloudtrail:GetInsightSelectors
\\",\\"cloudtrail:GetTrailStatus\\",\\"cloudtrail:ListChannels\\",
\\"cloudtrail:ListEventDataStores\\",\\"cloudtrail:ListQueries\\",\\"cloudtrail:ListTags
\\",\\"cloudtrail:LookupEvents\\",\\"ec2:DescribeRegions\\",\\"iam:GetAccountSummary
\\",\\"iam:GetOpenIDConnectProvider\\",\\"iam:GetRole\\",\\"iam:ListAccessKeys\\",
\\"iam:ListAccountAliases\\",\\"iam:ListOpenIDConnectProviders\\",\\"iam:ListRoles
\\",\\"iam:ListSAMLProviders\\",\\"kms:ListAliases\\",\\"s3:GetBucketLocation\\",
\\"s3:ListAllMyBuckets\\"],\\"Resource\\":\\"*\\"}]}"
    }
  ],
  "properties": {
    "cloudTrailProperties": {
      "endTime": "2024-02-14T22:44:40+00:00",
      "startTime": "2024-02-13T00:30:00+00:00",
      "trailProperties": [
        {
          "allRegions": true,
          "cloudTrailArn": "arn:aws:cloudtrail:us-
west-2:111122223333:trail/my-trail",
          "regions": []
        }
      ]
    },
    "isComplete": false,
    "principalArn": "arn:aws:iam::111122223333:role/Admin"
  }
},
"jobDetails": {
  "completedOn": "2024-02-14T22:47:01+00:00",
  "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
  "startedOn": "2024-02-14T22:44:41+00:00",
  "status": "SUCCEEDED"
}
}

```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Generierung von IAM Access Analyzer-Richtlinien](#).

- API-Einheiten finden Sie [GetGeneratedPolicy](#) in der AWS CLI Befehlsreferenz.

list-access-preview-findings

Das folgende Codebeispiel zeigt die Verwendung `list-access-preview-findings`.

AWS CLI

Um eine Liste von Access-Preview-Ergebnissen abzurufen, die mit der angegebenen Access-Preview generiert wurden

Im folgenden `list-access-preview-findings` Beispiel wird eine Liste mit Ergebnissen der Zugriffsvorschau abgerufen, die mit der angegebenen Zugriffsvorschau in Ihrem AWS Konto generiert wurden.

```
aws accessanalyzer list-access-preview-findings \  
  --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

Ausgabe:

```
{  
  "findings": [  
    {  
      "id": "e22fc158-1c87-4c32-9464-e7f405ce8d74",  
      "principal": {  
        "AWS": "111122223333"  
      },  
      "action": [  
        "s3:PutObject",  
        "s3:PutObjectAcl"  
      ],  
      "condition": {},  
      "resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
      "isPublic": false,  
      "resourceType": "AWS::S3::Bucket",  
      "createdAt": "2024-02-17T00:18:46+00:00",  
      "changeType": "NEW",  
      "status": "ACTIVE",  
      "resourceOwnerAccount": "111122223333",  
      "sources": [  
        {  
          "type": "POLICY"  
        }  
      ]  
    }  
  ]  
}
```



```

    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Vorschau des Zugriffs mit IAM Access Analyzer APIs](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [ListAccessPreviewFindings](#) in der AWS CLI Befehlsreferenz.

list-access-previews

Das folgende Codebeispiel zeigt die Verwendung `list-access-previews`.

AWS CLI

Um eine Liste von Zugriffsvorschauen für den angegebenen Analysator abzurufen

Im folgenden `list-access-previews` Beispiel wird eine Liste der Zugriffsvorschauen für den angegebenen Analysator in Ihrem Konto abgerufen. AWS

```

aws accessanalyzer list-access-previews \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account

```

Ausgabe:

```

{
  "accessPreviews": [
    {
      "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",
      "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
      "createdAt": "2024-02-17T00:18:44+00:00",
      "status": "COMPLETED"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Vorschau des Zugriffs mit IAM Access Analyzer APIs](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [ListAccessPreviews](#) in der AWS CLI Befehlsreferenz.

list-analyzed-resources

Das folgende Codebeispiel zeigt die Verwendung `list-analyzed-resources`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-analyzed-resources` Beispiel listet die verfügbaren Widgets in Ihrem AWS Konto auf.

```
aws accessanalyzer list-analyzed-resources \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
  --resource-type AWS::IAM::Role
```

Ausgabe:

```
{
  "analyzedResources": [
    {
      "resourceArn": "arn:aws:sns:us-west-2:111122223333:Validation-Email",
      "resourceOwnerAccount": "111122223333",
      "resourceType": "AWS::SNS::Topic"
    },
    {
      "resourceArn": "arn:aws:sns:us-west-2:111122223333:admin-alerts",
      "resourceOwnerAccount": "111122223333",
      "resourceType": "AWS::SNS::Topic"
    },
    {
      "resourceArn": "arn:aws:sns:us-west-2:111122223333:config-topic",
      "resourceOwnerAccount": "111122223333",
      "resourceType": "AWS::SNS::Topic"
    },
    {
      "resourceArn": "arn:aws:sns:us-west-2:111122223333:inspector-topic",
      "resourceOwnerAccount": "111122223333",
      "resourceType": "AWS::SNS::Topic"
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API Einheiten finden Sie [ListAnalyzedResources](#) unter AWS CLI Befehlsreferenz.

list-analyzers

Das folgende Codebeispiel zeigt die Verwendung `list-analyzers`.

AWS CLI

Um eine Liste von Analysatoren abzurufen

Im folgenden `list-analyzers` Beispiel wird eine Liste der Analysatoren in Ihrem Konto abgerufen. AWS

```
aws accessanalyzer list-analyzers
```

Ausgabe:

```
{
  "analyzers": [
    {
      "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/UnusedAccess-ConsoleAnalyzer-organization",
      "createdAt": "2024-02-15T00:46:40+00:00",
      "name": "UnusedAccess-ConsoleAnalyzer-organization",
      "status": "ACTIVE",
      "tags": {
        "auto-delete": "no"
      },
      "type": "ORGANIZATION_UNUSED_ACCESS"
    },
    {
      "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ConsoleAnalyzer-organization",
      "createdAt": "2020-04-25T07:43:28+00:00",
      "lastResourceAnalyzed": "arn:aws:s3::DOC-EXAMPLE-BUCKET",
      "lastResourceAnalyzedAt": "2024-02-15T21:51:56.517000+00:00",
      "name": "ConsoleAnalyzer-organization",
    }
  ]
}
```

```

        "status": "ACTIVE",
        "tags": {
            "auto-delete": "no"
        },
        "type": "ORGANIZATION"
    },
    {
        "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
        "createdAt": "2019-12-03T07:28:17+00:00",
        "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-
topic",
        "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
        "name": "ConsoleAnalyzer-account",
        "status": "ACTIVE",
        "tags": {
            "auto-delete": "no"
        },
        "type": "ACCOUNT"
    }
]
}

```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [ListAnalyzers](#) unter AWS CLI Befehlsreferenz.

list-archive-rules

Das folgende Codebeispiel zeigt die Verwendung `list-archive-rules`.

AWS CLI

Um eine Liste von Archivregeln abzurufen, die für den angegebenen Analyzer erstellt wurden

Im folgenden `list-archive-rules` Beispiel wird eine Liste von Archivregeln abgerufen, die für den angegebenen Analyzer in Ihrem AWS Konto erstellt wurden.

```

aws accessanalyzer list-archive-rules \
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization

```

Ausgabe:

```
{
  "archiveRules": [
    {
      "createdAt": "2024-02-15T00:49:27+00:00",
      "filter": {
        "resource": {
          "contains": [
            "Cognito"
          ]
        },
        "resourceType": {
          "eq": [
            "AWS::IAM::Role"
          ]
        }
      },
      "ruleName": "MyArchiveRule",
      "updatedAt": "2024-02-15T00:49:27+00:00"
    },
    {
      "createdAt": "2024-02-15T23:27:45+00:00",
      "filter": {
        "findingType": {
          "eq": [
            "UnusedIAMUserAccessKey"
          ]
        }
      },
      "ruleName": "ArchiveRule-56125a39-e517-4ff8-afb1-ef06f58db612",
      "updatedAt": "2024-02-15T23:27:45+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [ListArchiveRules](#) unter AWS CLI Befehlsreferenz.

list-findings-v2

Das folgende Codebeispiel zeigt die Verwendung `list-findings-v2`.

AWS CLI

Um eine Liste von Ergebnissen abzurufen, die vom angegebenen Analysator generiert wurden

Im folgenden `list-findings-v2` Beispiel wird eine Liste von Ergebnissen abgerufen, die vom angegebenen Analysator in Ihrem AWS Konto generiert wurden. In diesem Beispiel werden die Ergebnisse so gefiltert, dass sie nur IAM Rollen enthalten, deren Name enthält `Cognito`.

```
aws accessanalyzer list-findings-v2 \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
["AWS::IAM::Role"]}}'
```

Ausgabe:

```
{
  "findings": [
    {
      "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
      "createdAt": "2021-02-26T21:17:24.710000+00:00",
      "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
      "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
      "resourceType": "AWS::IAM::Role",
      "resourceOwnerAccount": "111122223333",
      "status": "ACTIVE",
      "updatedAt": "2021-02-26T21:17:24.710000+00:00",
      "findingType": "ExternalAccess"
    },
    {
      "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
      "createdAt": "2021-02-26T21:17:50.905000+00:00",
      "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
      "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
      "resourceType": "AWS::IAM::Role",
      "resourceOwnerAccount": "111122223333",
      "status": "ACTIVE",
      "updatedAt": "2021-02-26T21:17:50.905000+00:00",
      "findingType": "ExternalAccess"
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API Einzelheiten finden Sie unter [ListFindingsV2](#) in der AWS CLI Befehlsreferenz.

list-findings

Das folgende Codebeispiel zeigt die Verwendung `list-findings`.

AWS CLI

Um eine Liste von Ergebnissen abzurufen, die vom angegebenen Analysator generiert wurden

Im folgenden `list-findings` Beispiel wird eine Liste von Ergebnissen abgerufen, die vom angegebenen Analysator in Ihrem AWS Konto generiert wurden. In diesem Beispiel werden die Ergebnisse so gefiltert, dass sie nur IAM Rollen enthalten, deren Name enthält `Cognito`.

```
aws accessanalyzer list-findings \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ConsoleAnalyzer-account \
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq": ["AWS::IAM::Role"]}]'
```

Ausgabe:

```
{
  "findings": [
    {
      "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
      "principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "action": [
        "sts:AssumeRoleWithWebIdentity"
      ],
      "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolUnauth_Role",
      "isPublic": false,
      "resourceType": "AWS::IAM::Role",
```

```

    "condition": {
      "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
    },
    "createdAt": "2021-02-26T21:17:24.710000+00:00",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "updatedAt": "2021-02-26T21:17:24.710000+00:00",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333"
  },
  {
    "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
    "principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "action": [
      "sts:AssumeRoleWithWebIdentity"
    ],
    "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
    "isPublic": false,
    "resourceType": "AWS::IAM::Role",
    "condition": {
      "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
    },
    "createdAt": "2021-02-26T21:17:50.905000+00:00",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "updatedAt": "2021-02-26T21:17:50.905000+00:00",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333"
  }
]
}

```

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API-Einheiten finden Sie [ListFindings](#) unter AWS CLI Befehlsreferenz.

list-policy-generations

Das folgende Codebeispiel zeigt die Verwendung `list-policy-generations`.

AWS CLI

Um alle in den letzten sieben Tagen angeforderten Richtliniengenerationen aufzulisten

Im folgenden `list-policy-generations` Beispiel werden alle Versicherungsgenerationen aufgeführt, die in den letzten sieben Tagen in Ihrem AWS Konto angefordert wurden.

```
aws accessanalyzer list-policy-generations
```

Ausgabe:

```
{
  "policyGenerations": [
    {
      "completedOn": "2024-02-14T23:43:38+00:00",
      "jobId": "923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2",
      "principalArn": "arn:aws:iam::111122223333:role/Admin",
      "startedOn": "2024-02-14T23:43:02+00:00",
      "status": "CANCELED"
    },
    {
      "completedOn": "2024-02-14T22:47:01+00:00",
      "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
      "principalArn": "arn:aws:iam::111122223333:role/Admin",
      "startedOn": "2024-02-14T22:44:41+00:00",
      "status": "SUCCEEDED"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Generierung von IAM Access Analyzer-Richtlinien](#).

- API Einheiten finden Sie [ListPolicyGenerations](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um eine Liste von Tags abzurufen, die auf die angegebene Ressource angewendet wurden

Im folgenden `list-tags-for-resource` Beispiel wird eine Liste von Tags abgerufen, die auf die angegebene Ressource in Ihrem AWS Konto angewendet wurden.

```
aws accessanalyzer list-tags-for-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

Ausgabe:

```
{  
  "tags": {  
    "Zone-of-trust": "Account",  
    "Name": "ConsoleAnalyzer"  
  }  
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Generierung von IAM Access Analyzer-Richtlinien](#).

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-policy-generation

Das folgende Codebeispiel zeigt die Verwendung `start-policy-generation`.

AWS CLI

Um eine Anfrage zur Richtliniengenerierung zu starten

Im folgenden `start-policy-generation` Beispiel wird eine Anfrage zur Generierung von Richtlinien in Ihrem AWS Konto gestartet.

```
aws accessanalyzer start-policy-generation \  
  --policy-generation-details '{"principalArn": "arn:aws:iam::111122223333:role/  
Admin"}' \  
  --cloud-trail-details file://myfile.json
```

Inhalt von `myfile.json`:

```
{
```

```
"accessRole": "arn:aws:iam::111122223333:role/service-role/
AccessAnalyzerMonitorServiceRole",
"startTime": "2024-02-13T00:30:00Z",
"trails": [
  {
    "allRegions": true,
    "cloudTrailArn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/my-
trail"
  }
]
```

Ausgabe:

```
{
  "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"
}
```

Weitere Informationen finden Sie unter [Generierung von IAM Access Analyzer-Richtlinien](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [StartPolicyGeneration](#) in der AWS CLI Befehlsreferenz.

start-resource-scan

Das folgende Codebeispiel zeigt die Verwendung `start-resource-scan`.

AWS CLI

Um sofort einen Scan der Richtlinien zu starten, die auf die angegebene Ressource angewendet wurden

Im folgenden `start-resource-scan` Beispiel wird sofort ein Scan der Richtlinien gestartet, die auf die angegebene Ressource in Ihrem AWS Konto angewendet wurden.

```
aws accessanalyzer start-resource-scan \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
  --resource-arn arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Generierung von IAM Access Analyzer-Richtlinien](#).

- APIEinzelheiten finden Sie [StartResourceScan](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendungtag-resource.

AWS CLI

Um der angegebenen Ressource ein Tag hinzuzufügen

Im folgenden tag-resource Beispiel wird der angegebenen Ressource in Ihrem AWS Konto ein Tag hinzugefügt.

```
aws accessanalyzer tag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tags Environment=dev,Purpose=testing
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendunguntag-resource.

AWS CLI

Um Tags aus den angegebenen Ressourcen zu entfernen

Im folgenden untag-resource Beispiel werden Tags aus der angegebenen Ressource in Ihrem AWS Konto entfernt.

```
aws accessanalyzer untag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tag-key Environment
```

```
--tag-keys Environment Purpose
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-archive-rule

Das folgende Codebeispiel zeigt die Verwendung `update-archive-rule`.

AWS CLI

Um die Kriterien und Werte für die angegebene Archivierungsregel zu aktualisieren

Im folgenden `update-archive-rule` Beispiel werden die Kriterien und Werte für die angegebene Archivierungsregel in Ihrem AWS Konto aktualisiert.

```
aws accessanalyzer update-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":  
  ["AWS::IAM::Role"]}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Archivierungsregeln](#).

- APIEinzelheiten finden Sie [UpdateArchiveRule](#) in der AWS CLI Befehlsreferenz.

update-findings

Das folgende Codebeispiel zeigt die Verwendung `update-findings`.

AWS CLI

Um den Status der angegebenen Ergebnisse zu aktualisieren

Im folgenden `update-findings` Beispiel wird der Status der angegebenen Ergebnisse in Ihrem AWS Konto aktualisiert.

```
aws accessanalyzer update-findings \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
UnusedAccess-ConsoleAnalyzer-organization \  
  --ids 4f319ac3-2e0c-4dc4-bf51-7013a086b6ae 780d586a-2cce-4f72-aff6-359d450e7500  
 \  
  --status ARCHIVED
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von AWS Identity and Access Management Access Analyzer](#) im AWS IAM Benutzerhandbuch.

- API Einzelheiten finden Sie [UpdateFindings](#) unter AWS CLI Befehlsreferenz.

validate-policy

Das folgende Codebeispiel zeigt die Verwendung `validate-policy`.

AWS CLI

Um die Validierung einer Richtlinie anzufordern und eine Ergebnisliste zurückzugeben

Das folgende `validate-policy` Beispiel fordert die Validierung einer Richtlinie an und gibt eine Ergebnisliste zurück. Die Richtlinie im Beispiel ist eine Rollenvertrauensrichtlinie für eine Amazon Cognito Cognito-Rolle, die für den Web-Identitätsverbund verwendet wird. Die Ergebnisse der Vertrauensrichtlinie beziehen sich auf einen leeren `Sid` Elementwert und ein nicht übereinstimmendes Grundprinzip der Richtlinie, da die falsche Aktion „Rolle übernehmen“ verwendet wurde. `sts:AssumeRole` Die richtige Aktion „Rolle annehmen“ für die Verwendung mit Cognito ist `sts:AssumeRoleWithWebIdentity`.

```
aws accessanalyzer validate-policy \  
  --policy-document file://myfile.json \  
  --policy-type RESOURCE_POLICY
```

Inhalt von `myfile.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "",
```

```

    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": [
      "sts:AssumeRole",
      "sts:TagSession"
    ],
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "us-west-2_EXAMPLE"
      }
    }
  }
]
}

```

Ausgabe:

```

{
  "findings": [
    {
      "findingDetails": "Add a value to the empty string in the Sid element.",
      "findingType": "SUGGESTION",
      "issueCode": "EMPTY_SID_VALUE",
      "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-checks-suggestion-empty-sid-value",
      "locations": [
        {
          "path": [
            {
              "value": "Statement"
            },
            {
              "index": 0
            },
            {
              "value": "Sid"
            }
          ],
          "span": {
            "end": {

```

```

        "column": 21,
        "line": 5,
        "offset": 81
      },
      "start": {
        "column": 19,
        "line": 5,
        "offset": 79
      }
    }
  ]
},
{
  "findingDetails": "The sts:AssumeRole action is invalid with the
following principal(s): cognito-identity.amazonaws.com. Use a SAML provider
principal with the sts:AssumeRoleWithSAML action or use an OIDC provider principal
with the sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if
you use either of the two options.",
  "findingType": "ERROR",
  "issueCode": "MISMATCHED_ACTION_FOR_PRINCIPAL",
  "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-mismatched-action-for-principal",
  "locations": [
    {
      "path": [
        {
          "value": "Statement"
        },
        {
          "index": 0
        },
        {
          "value": "Action"
        },
        {
          "index": 0
        }
      ],
      "span": {
        "end": {
          "column": 32,
          "line": 11,

```



```
        "offset": 274
      },
      "start": {
        "column": 16,
        "line": 11,
        "offset": 258
      }
    }
  },
  {
    "path": [
      {
        "value": "Statement"
      },
      {
        "index": 0
      },
      {
        "value": "Principal"
      },
      {
        "value": "Federated"
      }
    ],
    "span": {
      "end": {
        "column": 61,
        "line": 8,
        "offset": 202
      },
      "start": {
        "column": 29,
        "line": 8,
        "offset": 170
      }
    }
  }
]
},
{
  "findingDetails": "The following actions: sts:TagSession are not supported by the condition key cognito-identity.amazonaws.com:aud. The condition will not be evaluated for these actions. We recommend that you move these actions to a different statement without this condition key.",
```

```
"findingType": "ERROR",
"issueCode": "UNSUPPORTED_ACTION_FOR_CONDITION_KEY",
"learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-unsupported-action-for-condition-key",
"locations": [
  {
    "path": [
      {
        "value": "Statement"
      },
      {
        "index": 0
      },
      {
        "value": "Action"
      },
      {
        "index": 1
      }
    ],
    "span": {
      "end": {
        "column": 32,
        "line": 12,
        "offset": 308
      },
      "start": {
        "column": 16,
        "line": 12,
        "offset": 292
      }
    }
  },
  {
    "path": [
      {
        "value": "Statement"
      },
      {
        "index": 0
      },
      {
        "value": "Condition"
```

```
    },
    {
      "value": "StringEquals"
    },
    {
      "value": "cognito-identity.amazonaws.com:aud"
    }
  ],
  "span": {
    "end": {
      "column": 79,
      "line": 16,
      "offset": 464
    },
    "start": {
      "column": 58,
      "line": 16,
      "offset": 443
    }
  }
}
]
}
]
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Prüfungen zur Validierung von Richtlinien](#).

- APIEinzelheiten finden Sie [ValidatePolicy](#) in der AWS CLI Befehlsreferenz.

Image Builder Builder-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Image Builder Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-component

Das folgende Codebeispiel zeigt die Verwendung `create-component`.

AWS CLI

Um eine Komponente zu erstellen

Das folgende `create-component` Beispiel erstellt eine Komponente, die eine JSON Dokumentdatei verwendet und auf ein Komponentendokument in einem YAML Format verweist, das in einen Amazon S3 S3-Bucket hochgeladen wird.

```
aws imagebuilder create-component \  
  --cli-input-json file://create-component.json
```

Inhalt von `create-component.json`:

```
{  
  "name": "MyExampleComponent",  
  "semanticVersion": "2019.12.02",  
  "description": "An example component that builds, validates and tests an image",  
  "changeDescription": "Initial version.",  
  "platform": "Windows",  
  "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"  
}
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/examplecomponent/2019.12.02/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateComponent](#) unter AWS CLI Befehlsreferenz.

create-distribution-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-distribution-configuration`.

AWS CLI

Um eine Verteilungskonfiguration zu erstellen

Im folgenden `create-distribution-configuration` Beispiel wird mithilfe einer JSON-Datei eine Verteilungskonfiguration erstellt.

```
aws imagebuilder create-distribution-configuration \  
  --cli-input-json file:/create-distribution-configuration.json
```

Inhalt von `create-distribution-configuration.json`:

```
{  
  "name": "MyExampleDistribution",  
  "description": "Copies AMI to eu-west-1",  
  "distributions": [  
    {  
      "region": "us-west-2",  
      "amiDistributionConfiguration": {  
        "name": "Name {{imagebuilder:buildDate}}",  
        "description": "An example image name with parameter references",  
        "amiTags": {  
          "KeyName": "{{ssm:parameter_name}}"  
        },  
        "launchPermission": {  
          "userIds": [  
            "123456789012"  
          ]  
        }  
      }  
    },  
    {  
      "region": "eu-west-1",
```

```

        "amiDistributionConfiguration": {
            "name": "My {{imagebuilder:buildVersion}} image
            {{imagebuilder:buildDate}}",
            "amiTags": {
                "KeyName": "Value"
            },
            "launchPermission": {
                "userIds": [
                    "123456789012"
                ]
            }
        }
    ]
}

```

Ausgabe:

```

{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution"
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateDistributionConfiguration](#) unter AWS CLI Befehlsreferenz.

create-image-pipeline

Das folgende Codebeispiel zeigt die Verwendung `create-image-pipeline`.

AWS CLI

Um eine Image-Pipeline zu erstellen

Im folgenden `create-image-pipeline` Beispiel wird mithilfe einer JSON-Datei eine Image-Pipeline erstellt.

```
aws imagebuilder create-image-pipeline \
```

```
--cli-input-json file://create-image-pipeline.json
```

Inhalt von `create-image-pipeline.json`:

```
{
  "name": "MyWindows2016Pipeline",
  "description": "Builds Windows 2016 Images",
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/mybasicrecipe/2019.12.03",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexampledistribution",
  "imageTestsConfiguration": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 60
  },
  "schedule": {
    "scheduleExpression": "cron(0 0 * * SUN)",
    "pipelineExecutionStartCondition":
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
  },
  "status": "ENABLED"
}
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline"
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateImagePipeline](#) unter AWS CLI Befehlsreferenz.

create-image-recipe

Das folgende Codebeispiel zeigt die Verwendung `create-image-recipe`.

AWS CLI

Um ein Rezept zu erstellen

Im folgenden `create-image-recipe` Beispiel wird mithilfe einer JSON Datei ein Bildrezept erstellt. Die Komponenten werden in der Reihenfolge installiert, in der sie angegeben sind.

```
aws imagebuilder create-image-recipe \  
  --cli-input-json file://create-image-recipe.json
```

Inhalt von `create-image-recipe.json`:

```
{  
  "name": "MyBasicRecipe",  
  "description": "This example image recipe creates a Windows 2016 image.",  
  "semanticVersion": "2019.12.03",  
  "components":  
  [  
    {  
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/  
myexamplecomponent/2019.12.02/1"  
    },  
    {  
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/  
myimportedcomponent/1.0.0/1"  
    }  
  ],  
  "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-  
english-full-base-x86/xxxx.x.x"  
}
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateImageRecipe](#) unter AWS CLI Befehlsreferenz.

create-image

Das folgende Codebeispiel zeigt die Verwendung `create-image`.

AWS CLI

Um ein Bild zu erstellen

Im folgenden `create-image` Beispiel wird ein Bild erstellt.

```
aws imagebuilder create-image \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03 \  
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateImage](#) unter AWS CLI Befehlsreferenz.

create-infrastructure-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-infrastructure-configuration`.

AWS CLI

Um eine Infrastrukturkonfiguration zu erstellen

Im folgenden `create-infrastructure-configuration` Beispiel wird mithilfe einer JSON Datei eine Infrastrukturkonfiguration erstellt.

```
aws imagebuilder create-infrastructure-configuration \  
--cli-input-json file://create-infrastructure-configuration.json
```

Inhalt von create-infrastructure-configuration.json:

```
{  
  "name": "MyExampleInfrastructure",  
  "description": "An example that will retain instances of failed builds",  
  "instanceTypes": [  
    "m5.large", "m5.xlarge"  
  ],  
  "instanceProfileName": "EC2InstanceProfileForImageBuilder",  
  "securityGroupIds": [  
    "sg-a1b2c3d4"  
  ],  
  "subnetId": "subnet-a1b2c3d4",  
  "logging": {  
    "s3Logs": {  
      "s3BucketName": "bucket-name",  
      "s3KeyPrefix": "bucket-path"  
    }  
  },  
  "keyPair": "key-pair-name",  
  "terminateInstanceOnFailure": false,  
  "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-topic-name"  
}
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [CreateInfrastructureConfiguration](#) unter AWS CLI Befehlsreferenz.

delete-component

Das folgende Codebeispiel zeigt die Verwendung `delete-component`.

AWS CLI

Um eine Komponente zu löschen

Im folgenden `delete-component` Beispiel wird eine Build-Version einer Komponente gelöscht, indem ihre ARN angegeben wird.

```
aws imagebuilder delete-component \  
  --component-build-version-arn arn:aws:imagebuilder:us-  
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DeleteComponent](#) unter AWS CLI Befehlsreferenz.

delete-image-pipeline

Das folgende Codebeispiel zeigt die Verwendung `delete-image-pipeline`.

AWS CLI

Um eine Image-Pipeline zu löschen

Im folgenden `delete-image-pipeline` Beispiel wird eine Image-Pipeline gelöscht, indem ihre ARN angegeben wird.

```
aws imagebuilder delete-image-pipeline \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/myexamplepipeline
```

```
--image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
my-example-pipeline
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DeleteImagePipeline](#) unter AWS CLI Befehlsreferenz.

delete-image-recipe

Das folgende Codebeispiel zeigt die Verwendung `delete-image-recipe`.

AWS CLI

Um ein Bildrezept zu löschen

Im folgenden `delete-image-recipe` Beispiel wird ein Bildrezept gelöscht, indem es angegeben wird. ARN

```
aws imagebuilder delete-image-recipe \  
--image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/  
mybasicrecipe/2019.12.03
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DeleteImageRecipe](#) unter AWS CLI Befehlsreferenz.

delete-image

Das folgende Codebeispiel zeigt die Verwendung `delete-image`.

AWS CLI

Um ein Bild zu löschen

Im folgenden `delete-image` Beispiel wird eine Image-Build-Version gelöscht, indem ihre ARN angegeben wird.

```
aws imagebuilder delete-image \  
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/  
example-image/2019.12.02/1
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DeleteImage](#) unter AWS CLI Befehlsreferenz.

delete-infrastructure-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-infrastructure-configuration`.

AWS CLI

Um eine Infrastrukturkonfiguration zu löschen

Im folgenden `delete-infrastructure-configuration` Beispiel wird eine Image-Pipeline gelöscht, indem ihre ARN angegeben wird.

```
aws imagebuilder delete-infrastructure-configuration \  
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-  
east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DeleteInfrastructureConfiguration](#) unter AWS CLI Befehlsreferenz.

get-component-policy

Das folgende Codebeispiel zeigt die Verwendung `get-component-policy`.

AWS CLI

Um Details zur Komponentenrichtlinie abzurufen

Im folgenden `get-component-policy` Beispiel werden die Details einer Komponentenrichtlinie aufgeführt, indem ihre angegeben wird ARN.

```
aws imagebuilder get-component-policy \  
  --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-  
example-component/2019.12.03/1
```

Ausgabe:

```
{  
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
\"Allow\", \"Principal\": { \"AWS\": [ \"123456789012\" ] }, \"Action\":  
[ \"imagebuilder:GetComponent\", \"imagebuilder:ListComponents\" ], \"Resource\":  
[ \"arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-  
component/2019.12.03/1\" ] } ] }"  
}
```

Weitere Informationen finden Sie unter Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI < <https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html> >`__ im EC2Image Builder Builder-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [GetComponentPolicy](#).AWS CLI

get-component

Das folgende Codebeispiel zeigt die Verwendung `get-component`.

AWS CLI

Um Komponentendetails abzurufen

Das folgende `get-component` Beispiel listet die Details einer Komponente auf, indem ihre angegeben wird ARN.

```
aws imagebuilder get-component \
  --component-build-version-arn arn:aws:imagebuilder:us-
  west-2:123456789012:component/component-name/1.0.0/1
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "component": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0/1",
    "name": "component-name",
    "version": "1.0.0",
    "type": "TEST",
    "platform": "Linux",
    "owner": "123456789012",
    "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
testing document.\nschemaVersion: 1.0\n\nphases:\n - name: test\n  steps:\n
- name: HelloWorldStep\n      action: ExecuteBash\n      inputs:\n
commands:\n      - echo \"Hello World! Test.\"\n",
    "encrypted": true,
    "dateCreated": "2020-01-27T20:43:30.306Z",
    "tags": {}
  }
}
```

```
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [GetComponent](#) unter AWS CLI Befehlsreferenz.

get-distribution-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-distribution-configuration`.

AWS CLI

Um die Details einer Distributionskonfiguration abzurufen

Im folgenden `get-distribution-configuration` Beispiel werden die Details einer Verteilungskonfiguration angezeigt, indem ihre angegeben wird ARN.

```
aws imagebuilder get-distribution-configuration \
  --distribution-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/myexampledistribution",
    "name": "MyExampleDistribution",
    "description": "Copies AMI to eu-west-1 and exports to S3",
    "distributions": [
      {
        "region": "us-west-2",
        "amiDistributionConfiguration": {
          "name": "Name {{imagebuilder:buildDate}}",
          "description": "An example image name with parameter
references",
          "amiTags": {
            "KeyName": "{{ssm:parameter_name}}"
          },
          "launchPermission": {
            "userIds": [
```



```

        "123456789012"
      ]
    }
  },
  {
    "region": "eu-west-1",
    "amiDistributionConfiguration": {
      "name": "My {{imagebuilder:buildVersion}} image
{{imagebuilder:buildDate}}",
      "amiTags": {
        "KeyName": "Value"
      },
      "launchPermission": {
        "userIds": [
          "123456789012"
        ]
      }
    }
  }
],
"dateCreated": "2020-02-19T18:40:10.529Z",
"tags": {}
}
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [GetDistributionConfiguration](#) unter AWS CLI Befehlsreferenz.

get-image-pipeline

Das folgende Codebeispiel zeigt die Verwendung `get-image-pipeline`.

AWS CLI

Um Details zur Image-Pipeline abzurufen

Im folgenden `get-image-pipeline` Beispiel werden die Details einer Image-Pipeline aufgeführt, indem ihre angegeben wird ARN.

```
aws imagebuilder get-image-pipeline \
```

```
--image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imagePipeline": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline",
    "name": "MyWindows2016Pipeline",
    "description": "Builds Windows 2016 Images",
    "platform": "Windows",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/mybasicrecipe/2019.12.03",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexampledistribution",
    "imageTestsConfiguration": {
      "imageTestsEnabled": true,
      "timeoutMinutes": 60
    },
    "schedule": {
      "scheduleExpression": "cron(0 0 * * SUN)",
      "pipelineExecutionStartCondition": "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "ENABLED",
    "dateCreated": "2020-02-19T19:04:01.253Z",
    "dateUpdated": "2020-02-19T19:04:01.253Z",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [GetImagePipeline](#) unter AWS CLI Befehlsreferenz.

get-image-policy

Das folgende Codebeispiel zeigt die Verwendung von `get-image-policy`.

AWS CLI

Um Details zur Bildrichtlinie abzurufen

Im folgenden `get-image-policy` Beispiel werden die Details einer Image-Richtlinie aufgeführt, indem ihre angegeben wirdARN.

```
aws imagebuilder get-image-policy \  
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-  
image/2019.12.03/1
```

Ausgabe:

```
{  
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow\",  
\"Principal\": { \"AWS\": [ \"123456789012\" ] }, \"Action\": [ \"imagebuilder:GetImage\",  
\"imagebuilder:ListImages\" ], \"Resource\": [ \"arn:aws:imagebuilder:us-  
west-2:123456789012:image/my-example-image/2019.12.03/1\" ] } ] }"
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- APIEinzelheiten finden Sie [GetImagePolicy](#) unter AWS CLI Befehlsreferenz.

`get-image-recipe-policy`

Das folgende Codebeispiel zeigt die Verwendung `get-image-recipe-policy`.

AWS CLI

Um Einzelheiten zur Bildrezeptrichtlinie abzurufen

Im folgenden `get-image-recipe-policy` Beispiel werden die Details einer Bildrezeptrichtlinie aufgeführt, indem ihre angegeben wirdARN.

```
aws imagebuilder get-image-recipe-policy \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-  
example-image-recipe/2019.12.03/1
```

Ausgabe:

```
{
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":
\"Allow\", \"Principal\": { \"AWS\": [ \"123456789012\" ] }, \"Action\":
[ \"imagebuilder:GetImageRecipe\", \"imagebuilder:ListImageRecipes\" ], \"Resource\":
[ \"arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-
recipe/2019.12.03/1\" ] } ] }"
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [GetImageRecipePolicy](#) unter AWS CLI Befehlsreferenz.

get-image

Das folgende Codebeispiel zeigt die Verwendung von `get-image`.

AWS CLI

Um Bilddetails zu erhalten

Im folgenden `get-image` Beispiel werden die Details eines Bildes aufgeführt, indem es angegeben wird ARN.

```
aws imagebuilder get-image \
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "image": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1",
    "name": "MyBasicRecipe",
    "version": "2019.12.03/1",
    "platform": "Windows",
    "state": {
      "status": "BUILDING"
    },
    "imageRecipe": {
```

```
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03",
    "name": "MyBasicRecipe",
    "description": "This example image recipe creates a Windows 2016
image.",
    "platform": "Windows",
    "version": "2019.12.03",
    "components": [
      {
        "componentArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
      },
      {
        "componentArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myimportedcomponent/1.0.0/1"
      }
    ],
    "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-
server-2016-english-full-base-x86/2019.12.17/1",
    "dateCreated": "2020-02-14T19:46:16.904Z",
    "tags": {}
  },
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/myexampleinfrastructure",
    "name": "MyExampleInfrastructure",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
      "m5.large",
      "m5.xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageFactory",
    "securityGroupIds": [
      "sg-a1b2c3d4"
    ],
    "subnetId": "subnet-a1b2c3d4",
    "logging": {
      "s3Logs": {
        "s3BucketName": "bucket-name",
        "s3KeyPrefix": "bucket-path"
      }
    }
  },
  "keyPair": "Sam",
  "terminateInstanceOnFailure": false,
```

```

        "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
        "dateCreated": "2020-02-14T21:21:05.098Z",
        "tags": {}
    },
    "imageTestsConfiguration": {
        "imageTestsEnabled": true,
        "timeoutMinutes": 720
    },
    "dateCreated": "2020-02-14T23:14:13.597Z",
    "outputResources": {
        "amis": []
    },
    "tags": {}
}
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [GetImage](#) unter AWS CLI Befehlsreferenz.

get-infrastructure-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-infrastructure-configuration`.

AWS CLI

Um Details zur Infrastrukturkonfiguration abzurufen

Im folgenden `get-infrastructure-configuration` Beispiel werden die Details einer Infrastrukturkonfiguration aufgeführt, indem ihre angegeben wird ARN.

```

aws imagebuilder get-infrastructure-configuration \
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure

```

Ausgabe:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",

```

```
    "name": "MyExampleInfrastructure",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
      "m5.large",
      "m5.xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageBuilder",
    "securityGroupIds": [
      "sg-a48c95ef"
    ],
    "subnetId": "subnet-a48c95ef",
    "logging": {
      "s3Logs": {
        "s3BucketName": "bucket-name",
        "s3KeyPrefix": "bucket-path"
      }
    },
    "keyPair": "Name",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
    "dateCreated": "2020-02-19T19:11:51.858Z",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [GetInfrastructureConfiguration](#) unter AWS CLI Befehlsreferenz.

import-component

Das folgende Codebeispiel zeigt die Verwendung `import-component`.

AWS CLI

Um eine Komponente zu importieren

Im folgenden `import-component` Beispiel wird ein bereits vorhandenes Skript mithilfe einer JSON-Datei importiert.

```
aws imagebuilder import-component \  
  --cli-input-json file://import-component.json
```

Inhalt von `import-component.json`:

```
{
  "name": "MyImportedComponent",
  "semanticVersion": "1.0.0",
  "description": "An example of how to import a component",
  "changeDescription": "First commit message.",
  "format": "SHELL",
  "platform": "Windows",
  "type": "BUILD",
  "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/myimportedcomponent/1.0.0/1"
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ImportComponent](#) unter AWS CLI Befehlsreferenz.

list-component-build-versions

Das folgende Codebeispiel zeigt die Verwendung `list-component-build-versions`.

AWS CLI

Um die Build-Versionen von Komponenten aufzulisten

Das folgende `list-component-build-versions` Beispiel listet die Build-Versionen von Komponenten mit einer bestimmten semantischen Version auf.

```
aws imagebuilder list-component-build-versions --component-  
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/  
myexamplecomponent/2019.12.02
```


Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1",
      "name": "MyExampleComponent",
      "version": "2019.12.02",
      "platform": "Windows",
      "type": "BUILD",
      "owner": "123456789012",
      "description": "An example component that builds, validates and tests an
image",
      "changeDescription": "Initial version.",
      "dateCreated": "2020-02-19T18:53:45.940Z",
      "tags": {
        "KeyName": "KeyValue"
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListComponentBuildVersions](#) unter AWS CLI Befehlsreferenz.

list-components

Das folgende Codebeispiel zeigt die Verwendung `list-components`.

AWS CLI

Um alle semantischen Versionen der Komponenten aufzulisten

Das folgende `list-components` Beispiel listet alle semantischen Versionen der Komponenten auf, auf die Sie Zugriff haben. Sie können optional danach filtern, ob Sie Komponenten auflisten möchten, die Ihnen oder Amazon gehören oder die von anderen Konten mit Ihnen geteilt wurden.

```
aws imagebuilder list-components
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0",
      "name": "component-name",
      "version": "1.0.0",
      "platform": "Linux",
      "type": "TEST",
      "owner": "123456789012",
      "dateCreated": "2020-01-27T20:43:30.306Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListComponents](#) unter AWS CLI Befehlsreferenz.

list-distribution-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-distribution-configurations`.

AWS CLI

Um Distributionen aufzulisten

Das folgende `list-distribution-configurations` Beispiel listet alle Ihre Distributionen auf.

```
aws imagebuilder list-distribution-configurations
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "distributionConfigurationSummaryList": [
    {
```

```

        "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-
configuration/myexampledistribution",
        "name": "MyExampleDistribution",
        "description": "Copies AMI to eu-west-1 and exports to S3",
        "dateCreated": "2020-02-19T18:40:10.529Z",
        "tags": {
            "KeyName": "KeyValue"
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListDistributionConfigurations](#) unter AWS CLI Befehlsreferenz.

list-image-build-versions

Das folgende Codebeispiel zeigt die Verwendung `list-image-build-versions`.

AWS CLI

Um Image-Build-Versionen aufzulisten

Das folgende `list-image-build-versions` Beispiel listet alle Image-Build-Versionen mit einer semantischen Version auf.

```

aws imagebuilder list-image-build-versions \
  --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03

```

Ausgabe:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/7",
      "name": "MyBasicRecipe",
      "version": "2019.12.03/7",
      "platform": "Windows",

```

```

    "state": {
      "status": "FAILED",
      "reason": "Can't start SSM Automation for arn
arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/7 during
building. Parameter \"iamInstanceProfileName\" has a null value."
    },
    "owner": "123456789012",
    "dateCreated": "2020-02-19T18:56:11.511Z",
    "outputResources": {
      "amis": []
    },
    "tags": {}
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/6",
    "name": "MyBasicRecipe",
    "version": "2019.12.03/6",
    "platform": "Windows",
    "state": {
      "status": "FAILED",
      "reason": "An internal error has occurred."
    },
    "owner": "123456789012",
    "dateCreated": "2020-02-18T22:49:08.142Z",
    "outputResources": {
      "amis": [
        {
          "region": "us-west-2",
          "image": "ami-a1b2c3d4567890ab",
          "name": "MyBasicRecipe 2020-02-18T22-49-38.704Z",
          "description": "This example image recipe creates a Windows
2016 image."
        },
        {
          "region": "us-west-2",
          "image": "ami-a1b2c3d4567890ab",
          "name": "Name 2020-02-18T22-49-08.131Z",
          "description": "Copies AMI to eu-west-2 and exports to S3"
        },
        {
          "region": "eu-west-2",
          "image": "ami-a1b2c3d4567890ab",
          "name": "My 6 image 2020-02-18T22-49-08.131Z",

```

```

        "description": "Copies AMI to eu-west-2 and exports to S3"
      }
    ]
  },
  "tags": {}
},
{
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/5",
  "name": "MyBasicRecipe",
  "version": "2019.12.03/5",
  "platform": "Windows",
  "state": {
    "status": "AVAILABLE"
  },
  "owner": "123456789012",
  "dateCreated": "2020-02-18T16:51:48.403Z",
  "outputResources": {
    "amis": [
      {
        "region": "us-west-2",
        "image": "ami-a1b2c3d4567890ab",
        "name": "MyBasicRecipe 2020-02-18T16-52-18.965Z",
        "description": "This example image recipe creates a Windows
2016 image."
      }
    ]
  },
  "tags": {}
},
{
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/4",
  "name": "MyBasicRecipe",
  "version": "2019.12.03/4",
  "platform": "Windows",
  "state": {
    "status": "AVAILABLE"
  },
  "owner": "123456789012",
  "dateCreated": "2020-02-18T16:50:01.827Z",
  "outputResources": {
    "amis": [
      {

```

```

        "region": "us-west-2",
        "image": "ami-a1b2c3d4567890ab",
        "name": "MyBasicRecipe 2020-02-18T16-50-32.280Z",
        "description": "This example image recipe creates a Windows
2016 image."
    }
  ]
},
"tags": {}
},
{
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/3",
  "name": "MyBasicRecipe",
  "version": "2019.12.03/3",
  "platform": "Windows",
  "state": {
    "status": "AVAILABLE"
  },
  "owner": "123456789012",
  "dateCreated": "2020-02-14T23:14:13.597Z",
  "outputResources": {
    "amis": [
      {
        "region": "us-west-2",
        "image": "ami-a1b2c3d4567890ab",
        "name": "MyBasicRecipe 2020-02-14T23-14-44.243Z",
        "description": "This example image recipe creates a Windows
2016 image."
      }
    ]
  },
  "tags": {}
},
{
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/2",
  "name": "MyBasicRecipe",
  "version": "2019.12.03/2",
  "platform": "Windows",
  "state": {
    "status": "FAILED",
    "reason": "SSM execution 'a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'
failed with status = 'Failed' and failure message = 'Step fails when it is

```

```

verifying the command has completed. Command a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
returns unexpected invocation result: \n{Status=[Failed], ResponseCode=[1],
Output=[\n-----ERROR-----\nfailed to run commands: exit status 1],
OutputPayload=[{"Status\":"Failed","\nResponseCode\":"1","\nOutput\":"\n
\n-----ERROR-----\nfailed to run commands: exit status 1","\nCommandId\":"
\na1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}], CommandId=[a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111]}. Please refer to Automation Service Troubleshooting Guide for more
diagnosis details.'"
    },
    "owner": "123456789012",
    "dateCreated": "2020-02-14T22:57:42.593Z",
    "outputResources": {
      "amis": []
    },
    },
    "tags": {}
  }
]
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListImageBuildVersions](#) unter AWS CLI Befehlsreferenz.

list-image-pipeline-images

Das folgende Codebeispiel zeigt die Verwendung `list-image-pipeline-images`.

AWS CLI

Um Pipeline-Bilder von Image-Pipelines aufzulisten

Im folgenden `list-image-pipeline-images` Beispiel werden alle Bilder aufgeführt, die mit einer bestimmten Image-Pipeline erstellt wurden.

```

aws imagebuilder list-image-pipeline-images \
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline

```

Ausgabe:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",

```

```
"imagePipelineList": [  
  {  
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline",  
    "name": "MyWindows2016Pipeline",  
    "description": "Builds Windows 2016 Images",  
    "platform": "Windows",  
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-  
recipe/mybasicrecipe/2019.12.03",  
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
    "imageTestsConfiguration": {  
      "imageTestsEnabled": true,  
      "timeoutMinutes": 60  
    },  
    "schedule": {  
      "scheduleExpression": "cron(0 0 * * SUN)",  
      "pipelineExecutionStartCondition":  
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"  
    },  
    "status": "ENABLED",  
    "dateCreated": "2020-02-19T19:04:01.253Z",  
    "dateUpdated": "2020-02-19T19:04:01.253Z",  
    "tags": {  
      "KeyName": "KeyValue"  
    }  
  },  
  {  
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/sam",  
    "name": "PipelineName",  
    "platform": "Linux",  
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-  
recipe/recipe-name-a1b2c3d45678/1.0.0",  
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/infrastructureconfiguration-name-  
a1b2c3d45678",  
    "imageTestsConfiguration": {  
      "imageTestsEnabled": true,  
      "timeoutMinutes": 720  
    },  
    "status": "ENABLED",  
    "dateCreated": "2019-12-16T18:19:02.068Z",
```



```

        "dateUpdated": "2019-12-16T18:19:02.068Z",
        "tags": {
            "KeyName": "KeyValue"
        }
    ]
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListImagePipelineImages](#) unter AWS CLI Befehlsreferenz.

list-image-recipes

Das folgende Codebeispiel zeigt die Verwendung `list-image-recipes`.

AWS CLI

Um Bildrezepte aufzulisten

Das folgende `list-image-recipes` Beispiel listet alle Ihre Bildrezepte auf.

```
aws imagebuilder list-image-recipes
```

Ausgabe:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageRecipeSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/mybasicrecipe/2019.12.03",
      "name": "MyBasicRecipe",
      "platform": "Windows",
      "owner": "123456789012",
      "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-english-full-base-x86/2019.x.x",
      "dateCreated": "2020-02-19T18:54:25.975Z",
      "tags": {
        "KeyName": "KeyValue"
      }
    },
  ],
}

```

```

    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/recipe-
name-a1b2c3d45678/1.0.0",
      "name": "recipe-name-a1b2c3d45678",
      "platform": "Linux",
      "owner": "123456789012",
      "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/amazon-linux-2-
x86/2019.11.21",
      "dateCreated": "2019-12-16T18:19:00.120Z",
      "tags": {
        "KeyName": "KeyValue"
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListImageRecipes](#) unter AWS CLI Befehlsreferenz.

list-images

Das folgende Codebeispiel zeigt die Verwendung `list-images`.

AWS CLI

Um Bilder aufzulisten

Das folgende `list-images` Beispiel listet alle semantischen Versionen auf, auf die Sie Zugriff haben.

```
aws imagebuilder list-images
```

Ausgabe:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03",
      "name": "MyBasicRecipe",

```

```
        "version": "2019.12.03",
        "platform": "Windows",
        "owner": "123456789012",
        "dateCreated": "2020-02-14T21:29:18.810Z"
    }
]
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListImages](#) unter AWS CLI Befehlsreferenz.

list-infrastructure-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-infrastructure-configurations`.

AWS CLI

Um Infrastrukturkonfigurationen aufzulisten

Das folgende `list-infrastructure-configurations` Beispiel listet alle Ihre Infrastrukturkonfigurationen auf.

```
aws imagebuilder list-infrastructure-configurations
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "infrastructureConfigurationSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
      "name": "MyExampleInfrastructure",
      "description": "An example that will retain instances of failed builds",
      "dateCreated": "2020-02-19T19:11:51.858Z",
      "tags": {}
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/infrastructureconfiguration-name-a1b2c3d45678",
```

```

        "name": "infrastructureConfiguration-name-a1b2c3d45678",
        "dateCreated": "2019-12-16T18:19:01.038Z",
        "tags": {
            "KeyName": "KeyValue"
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2](#) Image Builder Builder-Benutzerhandbuch.

- API-Einheiten finden Sie [ListInfrastructureConfigurations](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine bestimmte Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für eine bestimmte Ressource auf.

```

aws imagebuilder list-tags-for-resource \
  --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline

```

Ausgabe:

```

{
  "tags": {
    "KeyName": "KeyValue"
  }
}

```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2](#) Image Builder Builder-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

put-component-policy

Das folgende Codebeispiel zeigt die Verwendung `put-component-policy`.

AWS CLI

Um eine Ressourcenrichtlinie auf eine Komponente anzuwenden

Der folgende `put-component-policy` Befehl wendet eine Ressourcenrichtlinie auf eine Build-Komponente an, um die kontoübergreifende gemeinsame Nutzung von Build-Komponenten zu ermöglichen. Wir empfehlen Ihnen, den RAM CLI Befehl `create-resource-share` zu verwenden. Wenn Sie den EC2 Image Builder CLI Builder-Befehl `put-component-policy` verwenden, müssen Sie den RAM CLI Befehl `promote-resource-share-create-from-policy` auch verwenden, damit die Ressource für alle Prinzipale sichtbar ist, mit denen die Ressource gemeinsam genutzt wird.

```
aws imagebuilder put-component-policy \  
  --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/  
examplecomponent/2019.12.02/1 \  
  --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":  
"Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":  
[ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ],  
"Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:component/  
examplecomponent/2019.12.02/1" ] } ] }'
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/  
examplecomponent/2019.12.02/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [PutComponentPolicy](#) unter AWS CLI Befehlsreferenz.

put-image-policy

Das folgende Codebeispiel zeigt die Verwendung `put-image-policy`.

AWS CLI

Um eine Ressourcenrichtlinie auf ein Bild anzuwenden

Der folgende `put-image-policy` Befehl wendet eine Ressourcenrichtlinie auf ein Bild an, um die kontoübergreifende gemeinsame Nutzung von Bildern zu ermöglichen. Wir empfehlen Ihnen, den RAM CLI Befehl `create-resource-share` zu verwenden. Wenn Sie den EC2 Image Builder CLI Builder-Befehl `put-image-policy` verwenden, müssen Sie auch den RAM CLI Befehl `promote-resource-share-create -from-policy` verwenden, damit die Ressource für alle Prinzipale sichtbar ist, mit denen die Ressource gemeinsam genutzt wird.

```
aws imagebuilder put-image-policy \  
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/example-  
image/2019.12.02/1 \  
  --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",  
"Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage",  
"imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-  
west-2:123456789012:image/example-image/2019.12.02/1" ] } ] }'
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-  
image/2019.12.02/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [PutImagePolicy](#) unter AWS CLI Befehlsreferenz.

put-image-recipe-policy

Das folgende Codebeispiel zeigt die Verwendung `put-image-recipe-policy`.

AWS CLI

Um eine Ressourcenrichtlinie auf ein Bildrezept anzuwenden

Der folgende `put-image-recipe-policy` Befehl wendet eine Ressourcenrichtlinie auf ein Bildrezept an, um die kontoübergreifende gemeinsame Nutzung von Bildrezepten zu ermöglichen.

Es wird empfohlen, den RAM CLI Befehl `create-resource-share` zu verwenden. Wenn Sie den EC2 Image Builder CLI Builder-Befehl `put-image-recipe-policy`, müssen Sie den RAM CLI Befehl `promote-resource-share-create-from-policy` auch verwenden, damit die Ressource für alle Prinzipale sichtbar ist, mit denen die Ressource gemeinsam genutzt wird.

```
aws imagebuilder put-image-recipe-policy \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
example-image-recipe/2019.12.02 \  
  --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":  
  "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":  
  [ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":  
  [ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/example-image-  
recipe/2019.12.02" ] } ] }'
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
example-image-recipe/2019.12.02/1"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [PutImageRecipePolicy](#) unter AWS CLI Befehlsreferenz.

start-image-pipeline-execution

Das folgende Codebeispiel zeigt die Verwendung `start-image-pipeline-execution`.

AWS CLI

Um eine Image-Pipeline manuell zu starten

Im folgenden `start-image-pipeline-execution` Beispiel wird eine Image-Pipeline manuell gestartet.

```
aws imagebuilder start-image-pipeline-execution \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
example-image-recipe/2019.12.02 \  
  --pipeline-definition-arn arn:aws:imagebuilder:us-west-2:123456789012:image-builder:  
example-pipeline-definition/2019.12.02 \  
  --start-time 2019-12-02T12:00:00Z
```

```
--image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/1"
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [StartImagePipelineExecution](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Im folgenden `tag-resource` Beispiel wird EC2 Image Builder mithilfe einer JSON-Datei eine Ressource hinzugefügt und mit Tags versehen.

```
aws imagebuilder tag-resource \  
--cli-input-json file://tag-resource.json
```

Inhalt von `tag-resource.json`:

```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline",
  "tags": {
    "KeyName": "KeyValue"
  }
}
```


Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2](#) Image Builder Builder-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird mithilfe einer JSON-Datei ein Tag aus einer Ressource entfernt.

```
aws imagebuilder untag-resource \  
  --cli-input-json file://tag-resource.json
```

Inhalt von `untag-resource.json`:

```
{  
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline",  
  "tagKeys": [  
    "KeyName"  
  ]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2](#) Image Builder Builder-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-distribution-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-distribution-configuration`.

AWS CLI

Um eine Distributionskonfiguration zu aktualisieren

Im folgenden `update-distribution-configuration` Beispiel wird eine Verteilungskonfiguration mithilfe einer JSON Datei aktualisiert.

```
aws imagebuilder update-distribution-configuration \  
  --cli-input-json file://update-distribution-configuration.json
```

Inhalt von `update-distribution-configuration.json`:

```
{  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
  "description": "Copies AMI to eu-west-2 and exports to S3",  
  "distributions": [  
    {  
      "region": "us-west-2",  
      "amiDistributionConfiguration": {  
        "name": "Name {{imagebuilder:buildDate}}",  
        "description": "An example image name with parameter references"  
      }  
    },  
    {  
      "region": "eu-west-2",  
      "amiDistributionConfiguration": {  
        "name": "My {{imagebuilder:buildVersion}} image  
{{imagebuilder:buildDate}}"  
      }  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [UpdateDistributionConfiguration](#) unter AWS CLI Befehlsreferenz.

update-image-pipeline

Das folgende Codebeispiel zeigt die Verwendung `update-image-pipeline`.

AWS CLI

Um eine Image-Pipeline zu aktualisieren

Im folgenden `update-image-pipeline` Beispiel wird eine Image-Pipeline mithilfe einer JSON-Datei aktualisiert.

```
aws imagebuilder update-image-pipeline \  
  --cli-input-json file://update-image-pipeline.json
```

Inhalt von `update-image-pipeline.json`:

```
{  
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
  "imageTestsConfiguration": {  
    "imageTestsEnabled": true,  
    "timeoutMinutes": 120  
  },  
  "schedule": {  
    "scheduleExpression": "cron(0 0 * * MON)",  
    "pipelineExecutionStartCondition":  
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"  
  },  
  "status": "DISABLED"  
}
```

Ausgabe:

```
{
```

```
"requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [UpdateImagePipeline](#) unter AWS CLI Befehlsreferenz.

update-infrastructure-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-infrastructure-configuration`.

AWS CLI

Um eine Infrastrukturkonfiguration zu aktualisieren

Im folgenden `update-infrastructure-configuration` Beispiel wird eine Infrastrukturkonfiguration mithilfe einer JSON-Datei aktualisiert.

```
aws imagebuilder update-infrastructure-configuration \  
  --cli-input-json file:/update-infrastructure-configuration.json
```

Inhalt von `update-infrastructure-configuration.json`:

```
{  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
  "description": "An example that will terminate instances of failed builds",  
  "instanceTypes": [  
    "m5.large", "m5.2xlarge"  
  ],  
  "instanceProfileName": "EC2InstanceProfileForImageFactory",  
  "securityGroupIds": [  
    "sg-a48c95ef"  
  ],  
  "subnetId": "subnet-a48c95ef",  
  "logging": {  
    "s3Logs": {  
      "s3BucketName": "bucket-name",  
      "s3KeyPrefix": "bucket-path"  
    }  
  },  
  "terminateInstanceOnFailure": true,  
}
```

```
"snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name"
}
```

Ausgabe:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Weitere Informationen finden Sie unter [Einrichten und Verwalten einer EC2 Image Builder Builder-Image-Pipeline mithilfe von AWS CLI im EC2 Image Builder Builder-Benutzerhandbuch](#).

- API-Einheiten finden Sie [UpdateInfrastructureConfiguration](#) unter AWS CLI Befehlsreferenz.

Beispiele für Incident Manager mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Incident Manager Aktionen ausführen und allgemeine Szenarien implementieren können.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-replication-set

Das folgende Codebeispiel zeigt die Verwendung `create-replication-set`.

AWS CLI

Um den Replikationssatz zu erstellen

Im folgenden `create-replication-set` Beispiel wird der Replikationssatz erstellt, den Incident Manager verwendet, um Daten in Ihrem Amazon Web Services Services-Konto

zu replizieren und zu verschlüsseln. In diesem Beispiel werden bei der Erstellung des Replikationssatzes die Regionen us-east-1 und us-east-2 verwendet.

```
aws ssm-incidents create-replication-set \
  --regions '{"us-east-1": {"sseKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}, "us-east-2":
{"sseKmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}}'
```

Ausgabe:

```
{
  "replicationSetArns": [
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
  ]
}
```

Weitere Informationen finden Sie unter [Verwenden des Incident Manager-Replikationssatzes](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateReplicationSet](#) unter AWS CLI Befehlsreferenz.

create-response-plan

Das folgende Codebeispiel zeigt die Verwendung `create-response-plan`.

AWS CLI

Um einen Reaktionsplan zu erstellen

Im folgenden `create-response-plan` Beispiel wird ein Reaktionsplan mit den angegebenen Details erstellt.

```
aws ssm-incidents create-response-plan \
  --chat-channel '{"chatbotSns": [{"arn:aws:sns:us-
east-1:111122223333:Standard_User"}]}' \
  --display-name "Example response plan" \
  --incident-template '{"impact": 5, "title": "example-incident"}' \
  --name "example-response" \
  --actions '[{"ssmAutomation": {"documentName": "AWSIncidents-
CriticalIncidentRunbookTemplate", "documentVersion": "$DEFAULT",
```

```
"roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager", "targetAccount":
"RESPONSE_PLAN_OWNER_ACCOUNT"]}]' \
--engagements '["arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"]'
```

Ausgabe:

```
{
  "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}
```

Weitere Informationen finden Sie unter [Vorbereitung auf Vorfälle](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateResponsePlan](#) in der AWS CLI Befehlsreferenz.

create-timeline-event

Das folgende Codebeispiel zeigt die Verwendung `create-timeline-event`.

AWS CLI

Beispiel 1: Um ein benutzerdefiniertes Timeline-Ereignis zu erstellen

Im folgenden `create-timeline-event` Beispiel wird ein benutzerdefiniertes Zeitplanereignis zur angegebenen Zeit des angegebenen Vorfalles erstellt.

```
aws ssm-incidents create-timeline-event \
--event-data "\"example timeline event\"" \
--event-time 2022-10-01T20:30:00.000 \
--event-type "Custom Event" \
--incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
```

Ausgabe:

```
{
  "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

Beispiel 2: So erstellen Sie ein Timeline-Ereignis mit einer Vorfalldatensatz

Im folgenden `create-timeline-event` Beispiel wird ein Zeitleistenereignis erstellt, das im Bereich „Vorfalldatensätze“ aufgeführt ist.

```
aws ssm-incidents create-timeline-event \  
  --event-data "\"New Note\"" \  
  --event-type "Note" \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE" \  
  --event-time 2023-06-20T12:06:00.000 \  
  --event-references '["resource": "arn:aws:ssm-incidents::111122223333:incident-  
record/Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE"]'
```

Ausgabe:

```
{  
  "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",  
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateTimelineEvent](#) in der AWS CLI Befehlsreferenz.

delete-incident-record

Das folgende Codebeispiel zeigt die Verwendung `delete-incident-record`.

AWS CLI

Um einen Vorfalldatensatz zu löschen

Im folgenden `delete-incident-record` Beispiel wird der angegebene Vorfalldatensatz gelöscht.

```
aws ssm-incidents delete-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-  
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Incident Tracking](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteIncidentRecord](#) in der AWS CLI Befehlsreferenz.

delete-replication-set

Das folgende Codebeispiel zeigt die Verwendung `delete-replication-set`.

AWS CLI

Um den Replikationssatz zu löschen

Im folgenden `delete-replication-set` Beispiel wird der Replikationssatz aus Ihrem Amazon Web Services Services-Konto gelöscht. Durch das Löschen des Replikationssatzes werden auch alle Incident Manager-Daten gelöscht. Dies kann nicht rückgängig gemacht werden.

```
aws ssm-incidents delete-replication-set \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden des Incident Manager-Replikationssatzes](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteReplicationSet](#) unter AWS CLI Befehlsreferenz.

delete-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-policy`.

AWS CLI

Um eine Ressourcenrichtlinie zu löschen

Im folgenden `delete-resource-policy` Beispiel wird eine Ressourcenrichtlinie aus einem Reaktionsplan gelöscht. Dadurch wird dem Schulleiter oder der Organisation, mit der der Reaktionsplan geteilt wurde, der Zugriff entzogen.

```
aws ssm-incidents delete-resource-policy \  
  --policy-id "be8b57191f0371f1c6827341aa3f0a03" \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"
```

```
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit gemeinsam genutzten Kontakten und Reaktionsplänen](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteResourcePolicy](#) unter AWS CLI Befehlsreferenz.

delete-response-plan

Das folgende Codebeispiel zeigt die Verwendung `delete-response-plan`.

AWS CLI

Um einen Reaktionsplan zu löschen

Im folgenden `delete-response-plan` Beispiel wird der angegebene Reaktionsplan gelöscht.

```
aws ssm-incidents delete-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Vorbereitung auf Vorfälle](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteResponsePlan](#) in der AWS CLI Befehlsreferenz.

delete-timeline-event

Das folgende Codebeispiel zeigt die Verwendung `delete-timeline-event`.

AWS CLI

Um ein Timeline-Ereignis zu löschen

Im folgenden `delete-timeline-event` Beispiel wird ein benutzerdefiniertes Timeline-Ereignis aus dem angegebenen Incident-Datensatz gelöscht.

```
aws ssm-incidents delete-timeline-event \  
  --event-id "c0bcc885-a41d-eb01-b4ab-9d2de193643c" \  
  --incident-id "arn:aws:ssm-incidents::111122223333:incident/Example-Incident"
```

```
--incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteTimelineEvent](#) in der AWS CLI Befehlsreferenz.

get-incident-record

Das folgende Codebeispiel zeigt die Verwendung `get-incident-record`.

AWS CLI

Um einen Vorfalldatensatz zu erhalten

Im folgenden `get-incident-record` Beispiel werden Details zum angegebenen Vorfalldatensatz abgerufen.

```
aws ssm-incidents get-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Ausgabe:

```
{  
  "incidentRecord": {  
    "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",  
    "automationExecutions": [],  
    "creationTime": "2021-05-21T18:16:57.579000+00:00",  
    "dedupeString": "c4bcc812-85e7-938d-2b78-17181176ee1a",  
    "impact": 5,  
    "incidentRecordSource": {  
      "createdBy": "arn:aws:iam::111122223333:user/draliatp",  
      "invokedBy": "arn:aws:iam::111122223333:user/draliatp",  
      "source": "aws.ssm-incidents.custom"  
    },  
    "lastModifiedBy": "arn:aws:iam::111122223333:user/draliatp",  
    "lastModifiedTime": "2021-05-21T18:16:59.149000+00:00",  
    "notificationTargets": [],  
    "status": "OPEN",
```

```

    "title": "Example-Incident"
  }
}

```

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetIncidentRecord](#) in der AWS CLI Befehlsreferenz.

get-replication-set

Das folgende Codebeispiel zeigt die Verwendung `get-replication-set`.

AWS CLI

Um den Replikationssatz abzurufen

Im folgenden `get-replication-set` Beispiel werden die Details des Replikationssatzes abgerufen, den Incident Manager verwendet, um Daten in Ihrem Amazon Web Services Services-Konto zu replizieren und zu verschlüsseln.

```

aws ssm-incidents get-replication-set \
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"

```

Ausgabe:

```

{
  "replicationSet": {
    "createdBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
    "createdTime": "2021-05-14T17:57:22.010000+00:00",
    "deletionProtected": false,
    "lastModifiedBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
    "lastModifiedTime": "2021-05-14T17:57:22.010000+00:00",
    "regionMap": {
      "us-east-1": {
        "sseKmsKeyId": "DefaultKey",
        "status": "ACTIVE"
      },
      "us-east-2": {
        "sseKmsKeyId": "DefaultKey",
        "status": "ACTIVE",
        "statusMessage": "Tagging inaccessible"
      }
    }
  }
}

```

```

    }
  },
  "status": "ACTIVE"
}
}

```

Weitere Informationen finden Sie unter [Verwenden des Incident Manager-Replikationssets](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetReplicationSet](#) unter AWS CLI Befehlsreferenz.

get-resource-policies

Das folgende Codebeispiel zeigt die Verwendung `get-resource-policies`.

AWS CLI

Um Ressourcenrichtlinien für einen Reaktionsplan aufzulisten

Das folgende `command-name` Beispiel listet die Ressourcenrichtlinien auf, die dem angegebenen Reaktionsplan zugeordnet sind.

```

aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"

```

Ausgabe:

```

{
  "resourcePolicies": [
    {
      "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"d901b37a-dbb0-458a-8842-75575c464219-external-principals\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":{\"arn:aws:iam::222233334444:root\"}},\"Action\":[\"ssm-incidents:GetResponsePlan\",\"ssm-incidents:StartIncident\",\"ssm-incidents:UpdateIncidentRecord\",\"ssm-incidents:GetIncidentRecord\",\"ssm-incidents:CreateTimelineEvent\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-incidents:GetTimelineEvent\",\"ssm-incidents:ListTimelineEvents\",\"ssm-incidents:UpdateRelatedItems\",\"ssm-incidents:ListRelatedItems\"],\"Resource\":[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\",\"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*\"]}]}",

```

```

        "policyId": "be8b57191f0371f1c6827341aa3f0a03",
        "ramResourceShareRegion": "us-east-1"
    }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit gemeinsamen Kontakten und Reaktionsplänen](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetResourcePolicies](#) unter AWS CLI Befehlsreferenz.

get-response-plan

Das folgende Codebeispiel zeigt die Verwendung `get-response-plan`.

AWS CLI

Um Einzelheiten zu einem Reaktionsplan zu erhalten

Im folgenden `command-name` Beispiel werden Details zu einem bestimmten Reaktionsplan in Ihrem AWS Konto abgerufen.

```

aws ssm-incidents get-response-plan \
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"

```

Ausgabe:

```

{
  "actions": [
    {
      "ssmAutomation": {
        "documentName": "AWSIncidents-CriticalIncidentRunbookTemplate",
        "documentVersion": "$DEFAULT",
        "roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-incidents.amazonaws.com/AWSServiceRoleForIncidentManager",
        "targetAccount": "RESPONSE_PLAN_OWNER_ACCOUNT"
      }
    }
  ],
  "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan",
  "chatChannel": {

```

```

    "chatbotSns": [
      "arn:aws:sns:us-east-1:111122223333:Standard_User"
    ],
    "displayName": "Example response plan",
    "engagements": [
      "arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"
    ],
    "incidentTemplate": {
      "impact": 5,
      "title": "Example-Incident"
    },
    "name": "Example-Response-Plan"
  }

```

Weitere Informationen finden Sie unter [Vorbereitung auf Vorfälle](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetResponsePlan](#) in der AWS CLI Befehlsreferenz.

get-timeline-event

Das folgende Codebeispiel zeigt die Verwendung `get-timeline-event`.

AWS CLI

Um Details zu einem Timeline-Ereignis abzurufen

Das folgende `get-timeline-event` Beispiel gibt Details zum angegebenen Timeline-Ereignis zurück.

```

aws ssm-incidents get-timeline-event \
  --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
  Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"

```

Ausgabe:

```

{
  "event": {
    "eventData": "\"Incident Started\"",
    "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",

```

```
    "eventTime": "2021-05-21T18:16:57+00:00",
    "eventType": "Custom Event",
    "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
  }
}
```

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetTimelineEvent](#) in der AWS CLI Befehlsreferenz.

list-incident-records

Das folgende Codebeispiel zeigt die Verwendung `list-incident-records`.

AWS CLI

Um Vorfallaufzeichnungen aufzulisten

Das folgende `command-name` Beispiel listet die Vorfallaufzeichnungen in Ihrem Amazon Web Services Services-Konto auf.

```
aws ssm-incidents list-incident-records
```

Ausgabe:

```
{
  "incidentRecordSummaries": [
    {
      "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-
Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",
      "creationTime": "2021-05-21T18:16:57.579000+00:00",
      "impact": 5,
      "incidentRecordSource": {
        "createdBy": "arn:aws:iam::111122223333:user/drალიატპ",
        "invokedBy": "arn:aws:iam::111122223333:user/drალიატპ",
        "source": "aws.ssm-incidents.custom"
      },
      "status": "OPEN",
      "title": "Example-Incident"
    }
  ]
}
```



```
]
}
```

Weitere Informationen finden Sie in der [Liste der Vorfälle](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListIncidentRecords](#) in der AWS CLI Befehlsreferenz.

list-related-items

Das folgende Codebeispiel zeigt die Verwendung `list-related-items`.

AWS CLI

Um verwandte Artikel aufzulisten

Das folgende `list-related-items` Beispiel listet die verwandten Elemente des angegebenen Vorfalls auf.

```
aws ssm-incidents list-related-items \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
  Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Ausgabe:

```
{
  "relatedItems": [
    {
      "identifier": {
        "type": "OTHER",
        "value": {
          "url": "https://console.aws.amazon.com/systems-manager/opsitems/
oi-8ef82158e190/workbench?region=us-east-1"
        }
      },
      "title": "Example related item"
    },
    {
      "identifier": {
        "type": "PARENT",
        "value": {
          "arn": "arn:aws:ssm:us-east-1:111122223333:opsitem/
oi-8084126392ac"
        }
      }
    }
  ]
}
```

```
    }
    },
    "title": "parentItem"
  }
]
}
```

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRelatedItems](#) in der AWS CLI Befehlsreferenz.

list-replication-sets

Das folgende Codebeispiel zeigt die Verwendung `list-replication-sets`.

AWS CLI

Um den Replikationssatz aufzulisten

Im folgenden `list-replication-set` Beispiel wird der Replikationssatz aufgeführt, den Incident Manager verwendet, um Daten in Ihrem AWS Konto zu replizieren und zu verschlüsseln.

```
aws ssm-incidents list-replication-sets
```

Ausgabe:

```
{
  "replicationSetArns": [
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
  ]
}
```

Weitere Informationen finden Sie unter [Verwenden des Incident Manager-Replikationssatzes](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListReplicationSets](#) unter AWS CLI Befehlsreferenz.

list-response-plans

Das folgende Codebeispiel zeigt die Verwendung `list-response-plans`.

AWS CLI

Um die verfügbaren Reaktionspläne aufzulisten

Das folgende `list-response-plans` Beispiel listet die verfügbaren Reaktionspläne in Ihrem Amazon Web Services Services-Konto auf.

```
aws ssm-incidents list-response-plans
```

Ausgabe:

```
{
  "responsePlanSummaries": [
    {
      "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan",
      "displayName": "Example response plan",
      "name": "Example-Response-Plan"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Vorbereitung auf Vorfälle](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListResponsePlans](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für einen Reaktionsplan aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags auf, die dem angegebenen Reaktionsplan zugeordnet sind.

```
aws ssm-incidents list-tags-for-resource \
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Ausgabe:

```
{
  "tags": {
    "group1": "1"
  }
}
```

Weitere Informationen finden Sie unter [Tagging](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-timeline-events

Das folgende Codebeispiel zeigt die Verwendung `list-timeline-events`.

AWS CLI

Um die Ereignisse eines Vorfalls in der Zeitleiste aufzulisten

Das folgende `command-name` Beispiel listet die Ereignisse auf der Zeitleiste des angegebenen Vorfalls auf.

```
aws ssm-incidents list-timeline-events \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
  Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Ausgabe:

```
{
  "eventSummaries": [
    {
      "eventId": "8cbcc889-35e1-a42d-2429-d6f100799915",
      "eventTime": "2021-05-21T22:36:13.766000+00:00",
      "eventType": "SSM Incident Record Update",
      "eventUpdatedTime": "2021-05-21T22:36:13.766000+00:00",
      "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
    },
    {
      "eventId": "a2bcc825-aab5-1787-c605-f9bb2640d85b",
```

```

    "eventTime": "2021-05-21T18:58:46.443000+00:00",
    "eventType": "SSM Incident Record Update",
    "eventUpdatedTime": "2021-05-21T18:58:46.443000+00:00",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
  },
  {
    "eventId": "5abcc812-89c0-b0a8-9437-1c74223d4685",
    "eventTime": "2021-05-21T18:16:59.149000+00:00",
    "eventType": "SSM Incident Record Update",
    "eventUpdatedTime": "2021-05-21T18:16:59.149000+00:00",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
  },
  {
    "eventId": "06bcc812-8820-405e-4065-8d2b14d29b92",
    "eventTime": "2021-05-21T18:16:58+00:00",
    "eventType": "SSM Automation Execution Start Failure for Incident",
    "eventUpdatedTime": "2021-05-21T18:16:58.689000+00:00",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
  },
  {
    "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",
    "eventTime": "2021-05-21T18:16:57+00:00",
    "eventType": "Custom Event",
    "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
  },
  {
    "eventId": "c0bcc885-a41d-eb01-b4ab-9d2de193643c",
    "eventTime": "2020-10-01T20:30:00+00:00",
    "eventType": "Custom Event",
    "eventUpdatedTime": "2021-05-21T22:28:26.299000+00:00",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
  }
]
}

```

Weitere Informationen finden Sie unter [Incident-Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTimelineEvents](#) in der AWS CLI Befehlsreferenz.

put-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `put-resource-policy`.

AWS CLI

Um einen Reaktionsplan und Vorfälle zu teilen

Im folgenden `command-name` Beispiel wird eine Ressourcenrichtlinie hinzugefügt `Example-Response-Plan`, die den Reaktionsplan und die zugehörigen Vorfälle mit dem angegebenen Prinzipal teilt.

```
aws ssm-incidents put-resource-policy \
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \
  --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\": \"ExampleResourcePolciy\", \"Effect\":\"Allow\", \"Principal\":{\"AWS\": \"arn:aws:iam::222233334444:root\"}, \"Action\":[\"ssm-incidents:GetResponsePlan\", \"ssm-incidents:StartIncident\", \"ssm-incidents:UpdateIncidentRecord\", \"ssm-incidents:GetIncidentRecord\", \"ssm-incidents:CreateTimelineEvent\", \"ssm-incidents:UpdateTimelineEvent\", \"ssm-incidents:GetTimelineEvent\", \"ssm-incidents:ListTimelineEvents\", \"ssm-incidents:UpdateRelatedItems\", \"ssm-incidents:ListRelatedItems\"], \"Resource\":[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\", \"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*\"]}]}"
```

Ausgabe:

```
{
  "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

Weitere Informationen finden Sie im Incident Manager-Benutzerhandbuch unter [Arbeiten mit gemeinsamen Kontakten und Reaktionsplänen](#).

- API-Einheiten finden Sie [PutResourcePolicy](#) unter AWS CLI Befehlsreferenz.

start-incident

Das folgende Codebeispiel zeigt die Verwendung `start-incident`.

AWS CLI

Um einen Vorfall zu starten

Im folgenden `start-incident` Beispiel wird ein Vorfall mit dem angegebenen Reaktionsplan gestartet.

```
aws ssm-incidents start-incident \  
  --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Ausgabe:

```
{  
  "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
}
```

Weitere Informationen finden Sie unter [Incident Creation](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [StartIncident](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einen Reaktionsplan zu taggen

Im folgenden `tag-resource` Beispiel wird ein bestimmter Antwortplan mit dem angegebenen Tag-Schlüssel-Wert-Paar gekennzeichnet.

```
aws ssm-incidents tag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \  
  --tags '{"group1":"1"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einem Reaktionsplan zu entfernen

Im folgenden `untag-resource` Beispiel werden die angegebenen Tags aus dem Reaktionsplan entfernt.

```
aws ssm-incidents untag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan" \  
  --tag-keys ["group1"]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-deletion-protection

Das folgende Codebeispiel zeigt die Verwendung `update-deletion-protection`.

AWS CLI

Um den Löschschutz für Replikationssätze zu aktualisieren

Im folgenden `update-deletion-protection` Beispiel wird der Löschschutz in Ihrem Konto aktualisiert, um Sie davor zu schützen, die letzte Region in Ihrem Replikationssatz zu löschen.

```
aws ssm-incidents update-deletion-protection \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/  
a2bcc5c9-0f53-8047-7fef-c20749989b40" \  
  --deletion-protected
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden des Incident Manager-Replikationssatzes](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDeletionProtection](#) unter AWS CLI Befehlsreferenz.

update-incident-record

Das folgende Codebeispiel zeigt die Verwendung `update-incident-record`.

AWS CLI

Um einen Vorfalldatensatz zu aktualisieren

Das folgende `command-name` Beispiel behebt den angegebenen Vorfall.

```
aws ssm-incidents update-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --status "RESOLVED"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateIncidentRecord](#) in der AWS CLI Befehlsreferenz.

update-related-items

Das folgende Codebeispiel zeigt die Verwendung `update-related-items`.

AWS CLI

Um ein Element zu aktualisieren, das sich auf Vorfälle bezieht

Im folgenden `update-related-item` Beispiel wird ein verwandtes Element aus dem angegebenen Vorfalldatensatz entfernt.

```
aws ssm-incidents update-related-items \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --related-items-update '{"itemToRemove": {"type": "OTHER", "value": {"url": "https://console.aws.amazon.com/systems-manager/opsitems/oi-8ef82158e190/workbench?region=us-east-1"}}}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRelatedItems](#) in der AWS CLI Befehlsreferenz.

update-replication-set

Das folgende Codebeispiel zeigt die Verwendung `update-replication-set`.

AWS CLI

Um einen Replikationssatz zu aktualisieren

Im folgenden `command-name` Beispiel wird die Region `us-east-2` aus dem Replikationssatz gelöscht.

```
aws ssm-incidents update-replication-set \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/  
a2bcc5c9-0f53-8047-7fef-c20749989b40" \  
  --actions '[{"deleteRegionAction": {"regionName": "us-east-2"}}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden des Incident Manager-Replikationssatzes](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateReplicationSet](#) unter AWS CLI Befehlsreferenz.

update-response-plan

Das folgende Codebeispiel zeigt die Verwendung `update-response-plan`.

AWS CLI

Um einen Reaktionsplan zu aktualisieren

Im folgenden `update-response-plan` Beispiel wird ein Chat-Kanal aus dem angegebenen Antwortplan entfernt.

```
aws ssm-incidents update-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"  
 \  
  --chat-channel '{"empty":{}}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Vorbereitung auf Vorfälle](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateResponsePlan](#) in der AWS CLI Befehlsreferenz.

update-timeline-event

Das folgende Codebeispiel zeigt die Verwendung `update-timeline-event`.

AWS CLI

Um ein Timeline-Ereignis zu aktualisieren

Im folgenden `update-timeline-event` Beispiel wird die Uhrzeit aktualisiert, zu der das Ereignis eingetreten ist.

```
aws ssm-incidents update-timeline-event \  
  --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --event-time "2021-05-21T18:10:57+00:00"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Incident Details](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateTimelineEvent](#) in der AWS CLI Befehlsreferenz.

Beispiele für Incident Manager-Kontakte mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Incident Manager-Kontakten Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-page

Das folgende Codebeispiel zeigt die Verwendung `accept-page`.

AWS CLI

Um eine Seite während eines Engagements zu akzeptieren

Im folgenden `accept-page` Beispiel wird ein Akzeptanzcode verwendet, der an den Kontaktkanal gesendet wird, um eine Seite zu akzeptieren.

```
aws ssm-contacts accept-page \  
  --page-id "arn:aws:ssm-contacts:us-east-2:682428703967:page/  
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3" \  
  --accept-type READ \  
  --accept-code 425440
```

Dieser Befehl erzeugt keine Ausgabe

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [AcceptPage](#) in der AWS CLI Befehlsreferenz.

activate-contact-channel

Das folgende Codebeispiel zeigt die Verwendung `activate-contact-channel`.

AWS CLI

Aktivieren Sie den Kontaktkanal eines Kontakts

Das folgende `activate-contact-channel` Beispiel aktiviert einen Kontaktkanal und macht ihn als Teil eines Vorfalls nutzbar.

```
aws ssm-contacts activate-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d" \  
  --accept-code 425440
```

```
--activation-code "466136"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ActivateContactChannel](#) in der AWS CLI Befehlsreferenz.

command-name

Das folgende Codebeispiel zeigt die Verwendung `command-name`.

AWS CLI

Um einen Kontakt zu löschen

Im folgenden `command-name` Beispiel wird ein Kontakt gelöscht. Der Kontakt wird über keinen Eskalationsplan, der sich auf ihn bezieht, mehr erreichbar sein.

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CommandName](#) in der AWS CLI Befehlsreferenz.

create-contact-channel

Das folgende Codebeispiel zeigt die Verwendung `create-contact-channel`.

AWS CLI

Um einen Kontaktkanal zu erstellen

Erstellt einen Kontaktkanal vom Typ SMS für den Kontakt Akua Mansa. Kontaktkanäle können vom Typ SMSEMAIL, oder erstellt werden. VOICE

```
aws ssm-contacts create-contact-channel \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --name "akuas sms-test" \  
  --type SMS \  
  \
```

```
--delivery-address '{"SimpleAddress": "+15005550199"}'
```

Ausgabe:

```
{
  "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateContactChannel](#) in der AWS CLI Befehlsreferenz.

create-contact

Das folgende Codebeispiel zeigt die Verwendung `create-contact`.

AWS CLI

Um einen Kontakt zu erstellen

Im folgenden `create-contact` Beispiel wird ein Kontakt in Ihrer Umgebung mit einem leeren Plan erstellt. Der Plan kann nach dem Erstellen von Kontaktkanälen aktualisiert werden. Verwenden Sie den `create-contact-channel` Befehl mit ARN der Ausgabe dieses Befehls. Nachdem Sie Kontaktkanäle für diesen Kontakt erstellt haben, verwenden Sie `update-contact`, um den Plan zu aktualisieren.

```
aws ssm-contacts create-contact \
  --alias "akuam" \
  --display-name "Akua Mansa" \
  --type PERSONAL \
  --plan '{"Stages": []}'
```

Ausgabe:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateContact](#) in der AWS CLI Befehlsreferenz.

deactivate-contact-channel

Das folgende Codebeispiel zeigt die Verwendung `deactivate-contact-channel`.

AWS CLI

Um einen Kontaktkanal zu deaktivieren

Das folgende `deactivate-contact-channel` Beispiel deaktiviert einen Kontaktkanal. Die Deaktivierung eines Kontaktkanals bedeutet, dass der Kontaktkanal während eines Vorfalls nicht mehr per Paging verbunden wird. Sie können einen Kontaktkanal auch jederzeit mit dem Befehl `activate-contact-channel` reaktivieren.

```
aws ssm-contacts deactivate-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeactivateContactChannel](#) in der AWS CLI Befehlsreferenz.

delete-contact-channel

Das folgende Codebeispiel zeigt die Verwendung `delete-contact-channel`.

AWS CLI

Um einen Kontaktkanal zu löschen

Im folgenden `delete-contact-channel` Beispiel wird ein Kontaktkanal gelöscht. Durch das Löschen eines Kontaktkanals wird sichergestellt, dass der Kontaktkanal während eines Vorfalls nicht per Paging aufgerufen wird.

```
aws ssm-contacts delete-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-  
channel/akuam/13149bad-52ee-45ea-ae1e-45857f78f9b2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteContactChannel](#) in der AWS CLI Befehlsreferenz.

delete-contact

Das folgende Codebeispiel zeigt die Verwendung `delete-contact`.

AWS CLI

Um einen Kontakt zu löschen

Im folgenden `delete-contact` Beispiel wird ein Kontakt gelöscht. Der Kontakt wird über keinen Eskalationsplan, der sich auf ihn bezieht, mehr erreichbar sein.

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteContact](#) in der AWS CLI Befehlsreferenz.

describe-engagement

Das folgende Codebeispiel zeigt die Verwendung `describe-engagement`.

AWS CLI

Um die Details eines Engagements zu beschreiben

Im folgenden `describe-engagement` Beispiel sind die Details einer Zusammenarbeit mit einem Kontakt- oder Eskalationsplan aufgeführt. Der Betreff und der Inhalt werden an die Kontaktkanäle gesendet.

```
aws ssm-contacts describe-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
  example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Ausgabe:

```
{
```



```

    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbc7fc53356",
    "Sender": "cli",
    "Subject": "cli-test",
    "Content": "Testing engagements via CLI",
    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "StartTime": "2021-05-18T18:25:41.151000+00:00"
}

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEngagement](#) in der AWS CLI Befehlsreferenz.

describe-page

Das folgende Codebeispiel zeigt die Verwendung `describe-page`.

AWS CLI

Um einem Kontaktkanal die Details einer Seite aufzulisten

Das folgende `describe-page` Beispiel listet Details einer Seite für einen Kontaktkanal auf. Die Seite wird den Betreff und den bereitgestellten Inhalt enthalten.

```

aws ssm-contacts describe-page \
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93"

```

Ausgabe:

```

{
  "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93",
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0",
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
  "Sender": "cli",
  "Subject": "cli-test",
  "Content": "Testing engagements via CLI",
}

```

```

    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "SentTime": "2021-05-18T18:43:29.301000+00:00",
    "ReadTime": "2021-05-18T18:43:55.708000+00:00",
    "DeliveryTime": "2021-05-18T18:43:55.265000+00:00"
  }

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribePage](#) in der AWS CLI Befehlsreferenz.

get-contact-channel

Das folgende Codebeispiel zeigt die Verwendung `get-contact-channel`.

AWS CLI

Um die Details eines Kontaktkanals aufzulisten

Das folgende `get-contact-channel` Beispiel listet die Details eines Kontaktkanals auf.

```

aws ssm-contacts get-contact-channel \
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"

```

Ausgabe:

```

{
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
  "ContactChannelArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
  "Name": "akuas sms",
  "Type": "SMS",
  "DeliveryAddress": {
    "SimpleAddress": "+15005550199"
  },
  "ActivationStatus": "ACTIVATED"
}

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetContactChannel](#) in der AWS CLI Befehlsreferenz.

get-contact-policy

Das folgende Codebeispiel zeigt die Verwendung `get-contact-policy`.

AWS CLI

Um die Ressourcenrichtlinien eines Kontakts aufzulisten

Das folgende `get-contact-policy` Beispiel listet die Ressourcenrichtlinien auf, die dem angegebenen Kontakt zugeordnet sind.

```
aws ssm-contacts get-contact-policy \
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Ausgabe:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",
  "Policy": "{\n\"Version\": \"2012-10-17\", \"Statement\": [{\n\"Sid\": \n\"SharePolicyForDocumentationDralia\", \"Effect\": \"Allow\", \"Principal\": \n{\n\"AWS\": \"222233334444\"}, \"Action\": [\n\"ssm-contacts:GetContact\", \"ssm-contacts:StartEngagement\", \"ssm-contacts:DescribeEngagement\", \"ssm-contacts:ListPagesByEngagement\", \"ssm-contacts:StopEngagement\"], \"Resource\": [\n\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\", \"arn:aws:ssm-contacts:*:111122223333:engagement/akuam/*\"]\n}]]}"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit gemeinsamen Kontakten und Reaktionsplänen](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetContactPolicy](#) unter AWS CLI Befehlsreferenz.

get-contact

Das folgende Codebeispiel zeigt die Verwendung `get-contact`.

AWS CLI

Beispiel 1: Um einen Kontaktplan zu beschreiben

Das folgende `get-contact` Beispiel beschreibt einen Kontakt.

```
aws ssm-contacts get-contact \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Ausgabe:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
  "Alias": "akuam",
  "DisplayName": "Akua Mansa",
  "Type": "PERSONAL",
  "Plan": {
    "Stages": [
      {
        "DurationInMinutes": 5,
        "Targets": [
          {
            "ChannelTargetInfo": {
              "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65",
              "RetryIntervalInMinutes": 1
            }
          }
        ]
      },
      {
        "DurationInMinutes": 5,
        "Targets": [
          {
            "ChannelTargetInfo": {
              "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad",
              "RetryIntervalInMinutes": 1
            }
          }
        ]
      },
      {
        "DurationInMinutes": 5,
        "Targets": [
          {
            "ChannelTargetInfo": {
              "ContactChannelId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/77d4f447-f619-4954-afff-85551e369c2a",
```

```

    "RetryIntervalInMinutes": 1
  }
}
]
}
]
}
}

```

Beispiel 2: Um einen Eskalationsplan zu beschreiben

Das folgende `get-contact` Beispiel beschreibt einen Eskalationsplan.

```

aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"

```

Ausgabe:

```

{
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
  "Alias": "example_escalation",
  "DisplayName": "Example Escalation",
  "Type": "ESCALATION",
  "Plan": {
    "Stages": [
      {
        "DurationInMinutes": 5,
        "Targets": [
          {
            "ContactTargetInfo": {
              "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/akuam",
              "IsEssential": true
            }
          }
        ]
      },
      {
        "DurationInMinutes": 5,
        "Targets": [
          {

```

```

        "ContactTargetInfo": {
            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/alejr",
            "IsEssential": false
        }
    ],
    {
        "DurationInMinutes": 0,
        "Targets": [
            {
                "ContactTargetInfo": {
                    "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/anasi",
                    "IsEssential": false
                }
            }
        ]
    }
]
}
}

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetContact](#) in der AWS CLI Befehlsreferenz.

list-contact-channels

Das folgende Codebeispiel zeigt die Verwendung `list-contact-channels`.

AWS CLI

Um die Kontaktkanäle eines Kontakts aufzulisten

Das folgende `list-contact-channels` Beispiel listet die verfügbaren Kontaktkanäle des angegebenen Kontakts auf.

```

aws ssm-contacts list-contact-channels \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"

```

Ausgabe:

```
{
  [
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
      "Name": "akuas email",
      "Type": "EMAIL",
      "DeliveryAddress": {
        "SimpleAddress": "akuam@example.com"
      },
      "ActivationStatus": "NOT_ACTIVATED"
    },
    {
      "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
      "Name": "akuas sms",
      "Type": "SMS",
      "DeliveryAddress": {
        "SimpleAddress": "+15005550100"
      },
      "ActivationStatus": "ACTIVATED"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListContactChannels](#) in der AWS CLI Befehlsreferenz.

list-contacts

Das folgende Codebeispiel zeigt die Verwendung `list-contacts`.

AWS CLI

Um alle Eskalationspläne und Kontakte aufzulisten

Das folgende `list-contacts` Beispiel listet die Kontakte und Eskalationspläne in Ihrem Konto auf.

```
aws ssm-contacts list-contacts
```

Ausgabe:

```
{
  "Contacts": [
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
      "Alias": "akuam",
      "DisplayName": "Akua Mansa",
      "Type": "PERSONAL"
    },
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
alejr",
      "Alias": "alejr",
      "DisplayName": "Alejandro Rosalez",
      "Type": "PERSONAL"
    },
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
anasi",
      "Alias": "anasi",
      "DisplayName": "Ana Carolina Silva",
      "Type": "PERSONAL"
    },
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
      "Alias": "example_escalation",
      "DisplayName": "Example Escalation",
      "Type": "ESCALATION"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListContacts](#) in der AWS CLI Befehlsreferenz.

list-engagements

Das folgende Codebeispiel zeigt die Verwendung `list-engagements`.

AWS CLI

Um alle Engagements aufzulisten

Das folgende `list-engagements` Beispiel listet Interaktionen mit Eskalationsplänen und Kontakten auf. Sie können auch Engagements für einen einzelnen Vorfall auflisten.

```
aws ssm-contacts list-engagements
```

Ausgabe:

```
{
  "Engagements": [
    {
      "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/akuam/91792571-0b53-4821-9f73-d25d13d9e529",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
      "Sender": "cli",
      "StartTime": "2021-05-18T20:37:50.300000+00:00"
    },
    {
      "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
      "Sender": "cli",
      "StartTime": "2021-05-18T18:40:26.666000+00:00"
    },
    {
      "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/example_escalation",
      "Sender": "cli",
      "StartTime": "2021-05-18T18:25:41.151000+00:00"
    },
    {
```

```

    "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f",
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
    "Sender": "cli",
    "StartTime": "2021-05-18T18:20:58.093000+00:00"
  }
]
}

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListEngagements](#) in der AWS CLI Befehlsreferenz.

list-page-receipts

Das folgende Codebeispiel zeigt die Verwendung `list-page-receipts`.

AWS CLI

Um Seitenbelege aufzulisten

Im folgenden `command-name` Beispiel wird aufgeführt, ob eine Seite von einem Kontakt empfangen wurde oder nicht.

```

aws ssm-contacts list-page-receipts \
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"

```

Ausgabe:

```

{
  "Receipts": [
    {
      "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
      "ReceiptType": "DELIVERED",
      "ReceiptInfo": "425440",
      "ReceiptTime": "2021-05-18T20:42:57.485000+00:00"
    },
    {
      "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",

```

```

        "ReceiptType": "READ",
        "ReceiptInfo": "425440",
        "ReceiptTime": "2021-05-18T20:42:57.907000+00:00"
    },
    {
        "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
        "ReceiptType": "SENT",
        "ReceiptInfo": "SM6656c19132f1465f9c9c1123a5dde7c9",
        "ReceiptTime": "2021-05-18T20:40:52.962000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPageReceipts](#) in der AWS CLI Befehlsreferenz.

list-pages-by-contact

Das folgende Codebeispiel zeigt die Verwendung `list-pages-by-contact`.

AWS CLI

Um Seiten nach Kontakt aufzulisten

Im folgenden `list-pages-by-contact` Beispiel werden alle Seiten des angegebenen Kontakts aufgelistet.

```

aws ssm-contacts list-pages-by-contact \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"

```

Ausgabe:

```

{
  "Pages": [
    {
      "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
      "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",

```

```

        "Sender": "cli",
        "SentTime": "2021-05-18T18:43:29.301000+00:00",
        "DeliveryTime": "2021-05-18T18:43:55.265000+00:00",
        "ReadTime": "2021-05-18T18:43:55.708000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPagesByContactin](#) der AWS CLI Befehlsreferenz.

list-pages-by-engagement

Das folgende Codebeispiel zeigt die Verwendung `list-pages-by-engagement`.

AWS CLI

Um Seiten mit Kontaktkanälen aufzulisten, die mit einem Engagement gestartet wurden.

Das folgende `list-pages-by-engagement` Beispiel listet die Seiten auf, die während der Interaktion mit dem definierten Engagement-Plan entstanden sind.

```

aws ssm-contacts list-pages-by-engagement \
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
  akuam/78a29753-3674-4ac5-9f83-0468563567f0"

```

Ausgabe:

```

{
  "Pages": [
    {
      "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
      ad0052bd-e606-498a-861b-25726292eb93",
      "EngagementArn": "arn:aws:ssm-contacts:us-
      east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
      akuam",
      "Sender": "cli",
      "SentTime": "2021-05-18T18:40:27.245000+00:00"
    }
  ]
}

```

```
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPagesByEngagement](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für einen Kontakt aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags des angegebenen Kontakts auf.

```
aws ssm-contacts list-tags-for-resource \
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "group1",
      "Value": "1"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-contact-policy

Das folgende Codebeispiel zeigt die Verwendung `put-contact-policy`.

AWS CLI

Um einen Kontakt und Interaktionen zu teilen

Im folgenden `put-contact-policy` Beispiel wird dem Kontakt Akua eine Ressourcenrichtlinie hinzugefügt, die den Kontakt und die damit verbundenen Interaktionen mit dem Principal teilt.

```
aws ssm-contacts put-contact-policy \  
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":  
  \"ExampleResourcePolicy\",\"Action\":[\"ssm-contacts:GetContact\",\"ssm-  
  contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-  
  contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],  
  \"Principal\":{\"AWS\":\"222233334444\"},\"Effect\":\"Allow\",\"Resource  
  \":[\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\",\"arn:aws:ssm-  
  contacts:*:111122223333:engagement/akuam/*\"]}]}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit gemeinsamen Kontakten und Reaktionsplänen](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [PutContactPolicy](#) unter AWS CLI Befehlsreferenz.

send-activation-code

Das folgende Codebeispiel zeigt die Verwendung `send-activation-code`.

AWS CLI

Um einen Aktivierungscode zu senden

Im folgenden `send-activation-code` Beispiel werden ein Aktivierungscode und eine Nachricht an den angegebenen Kontaktkanal gesendet.

```
aws ssm-contacts send-activation-code \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-  
  channel/akuam/8ddae2d1-12c8-4e45-b852-c8587266c400"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [SendActivationCode](#) in der AWS CLI Befehlsreferenz.

start-engagement

Das folgende Codebeispiel zeigt die Verwendung `start-engagement`.

AWS CLI

Beispiel 1: Zur Seite mit den Kontaktkanälen eines Kontakts

Auf den folgenden `start-engagement` Seiten finden Sie die Kontaktkanäle eines Kontakts. Absender, Betreff, öffentlicher Betreff und öffentlicher Inhalt sind alle frei von Feldern. Incident Manager sendet den Betreff und den Inhalt an die bereitgestellten VOICE Kanäle oder Kontaktkanäle. EMAIL Incident Manager sendet den öffentlichen Betreff und die öffentlichen Inhalte an die bereitgestellten Kontaktkanäle. SMS Der Sender wird verwendet, um nachzuverfolgen, wer das Engagement gestartet hat.

```
aws ssm-contacts start-engagement \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \  
  --sender "cli" \  
  --subject "cli-test" \  
  --content "Testing engagements via CLI" \  
  --public-subject "cli-test" \  
  --public-content "Testing engagements va CLI"
```

Ausgabe:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"  
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

Beispiel 2: Um einen Kontakt im bereitgestellten Eskalationsplan zu platzieren.

Im Folgenden werden `start-engagement` die Kontakte über einen Eskalationsplan eingebunden. Jeder Kontakt wird entsprechend seinem Engagementplan weitergeleitet.

```
aws ssm-contacts start-engagement \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/  
example_escalation" \  

```

```
--sender "cli" \  
--subject "cli-test" \  
--content "Testing engagements via CLI" \  
--public-subject "cli-test" \  
--public-content "Testing engagements va CLI"
```

Ausgabe:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"  
}
```

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- APIEinzelheiten finden Sie [StartEngagement](#) in der AWS CLI Befehlsreferenz.

stop-engagement

Das folgende Codebeispiel zeigt die Verwendung stop-engagement.

AWS CLI

Um ein Engagement zu beenden

Das folgende stop-engagement Beispiel verhindert, dass ein Engagement weitere Kontakte und Kontaktkanäle weiterleitet.

```
aws ssm-contacts stop-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- APIEinzelheiten finden Sie [StopEngagement](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung tag-resource.

AWS CLI

Um einen Kontakt zu taggen

Das folgende `tag-resource` Beispiel kennzeichnet einen angegebenen Kontakt mit dem angegebenen Tag-Schlüssel-Wert-Paar.

```
aws ssm-contacts tag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tags '[{"Key":"group1","Value":"1"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags von einem Kontakt zu entfernen

Im folgenden `untag-resource` Beispiel wird das `group1`-Tag aus dem angegebenen Kontakt entfernt.

```
aws ssm-contacts untag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tag-keys "group1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-contact-channel

Das folgende Codebeispiel zeigt die Verwendung `update-contact-channel`.

AWS CLI

Um einen Kontaktkanal zu aktualisieren

Das folgende `update-contact-channel` Beispiel aktualisiert den Namen und die Lieferadresse eines Kontaktkanals.

```
aws ssm-contacts update-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad" \  
  --name "akuas voice channel" \  
  --delivery-address '{"SimpleAddress": "+15005550198"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateContactChannel](#) in der AWS CLI Befehlsreferenz.

update-contact

Das folgende Codebeispiel zeigt die Verwendung `update-contact`.

AWS CLI

Um den Engagementplan des Kontakts zu aktualisieren

Im folgenden `update-contact` Beispiel wird der Engagementplan des Kontakts Akua aktualisiert, sodass er die drei Arten von Kontaktkanälen umfasst. Dies erfolgt nach der Erstellung von Kontaktkanälen für Akua.

```
aws ssm-contacts update-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \  
  --plan '{"Stages": [{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":  
{"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65", "RetryIntervalInMinutes":  
1 }]}], {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":  
{"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/  
akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad", "RetryIntervalInMinutes": 1 }]}],  
{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo": {"ContactChannelId":  
"arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/77d4f447-  
f619-4954-afff-85551e369c2a", "RetryIntervalInMinutes": 1 }]}]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kontakte](#) im Incident Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateContact](#) in der AWS CLI Befehlsreferenz.

Amazon Inspector Inspector-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Inspector Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-attributes-to-findings

Das folgende Codebeispiel zeigt die Verwendung `add-attributes-to-findings`.

AWS CLI

Um den Ergebnissen Attribute hinzuzufügen

Der folgende `add-attribute-to-finding` Befehl weist dem Ergebnis mit dem Wert `example` ein Attribut mit dem Schlüssel `Example` und dem Wert `ARN` von `examplearn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU` zu:

```
aws inspector add-attributes-to-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attributes key=Example,value=example
```

Ausgabe:

```
{
  "failedItems": {}
}
```

Weitere Informationen finden Sie unter Amazon Inspector Findings im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [AddAttributesToFindings](#) in der AWS CLI Befehlsreferenz.

create-assessment-target

Das folgende Codebeispiel zeigt die Verwendung `create-assessment-target`.

AWS CLI

Um ein Bewertungsziel zu erstellen

Mit dem folgenden `create-assessment-target` Befehl wird ein Bewertungsziel erstellt, das `ExampleAssessmentTarget` anhand der Ressourcengruppe mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv` benannt wird.

```
aws inspector create-assessment-target --assessment-target-name ExampleAssessmentTarget --resource-group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv
```

Ausgabe:

```
{
  "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX"
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [CreateAssessmentTarget](#) in der AWS CLI Befehlsreferenz.

create-assessment-template

Das folgende Codebeispiel zeigt die Verwendung `create-assessment-template`.

AWS CLI

So erstellen Sie eine Bewertungsvorlage

Mit dem folgenden `create-assessment-template` Befehl wird eine Bewertungsvorlage erstellt, die `ExampleAssessmentTemplate` für das Bewertungsziel mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX` aufgerufen wird `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX`:

```
aws inspector create-assessment-template --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-template-name ExampleAssessmentTemplate --duration-in-seconds 180 --rules-package-arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p --user-attributes-for-findings key=ExampleTag,value=examplevalue
```

Ausgabe:

```
{
  "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T"
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [CreateAssessmentTemplate](#) in der AWS CLI Befehlsreferenz.

create-filter

Das folgende Codebeispiel zeigt die Verwendung `create-filter`.

AWS CLI

Um einen Filter zu erstellen

Im folgenden `create-filter` Beispiel wird eine Unterdrückungsregel erstellt, die Ergebnisse vom ECR Instanztyp auslöst.

```
aws inspector2 create-filter \  
  --name "ExampleSuppressionRuleECR" \  
  --description "This suppression rule omits ECR instance type findings" \  
  --action SUPPRESS \  
  --
```

```
--filter-criteria 'resourceType=[{comparison="EQUALS",  
value="AWS_ECR_INSTANCE"}]'
```

Ausgabe:

```
{  
  "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/  
EXAMPLE4444444444"  
}
```

Weitere Informationen finden Sie unter [Filtern von Amazon Inspector Inspector-Ergebnissen](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateFilter](#) in der AWS CLI Befehlsreferenz.

create-findings-report

Das folgende Codebeispiel zeigt die Verwendung `create-findings-report`.

AWS CLI

Um einen Ergebnisbericht zu erstellen

Im folgenden `create-findings-report` Beispiel wird ein Ergebnisbericht erstellt.

```
aws inspector2 create-findings-report \  
  --report-format CSV \  
  --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-  
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE33333 \  
  --filter-criteria '{"ecrImageRepositoryName":  
[{"comparison":"EQUALS","value":"debian"}]}'
```

Ausgabe:

```
{  
  "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
}
```

Weitere Informationen finden Sie unter [Ergebnisse in Amazon Inspector verwalten im Amazon Inspector](#) Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateFindingsReport](#) in der AWS CLI Befehlsreferenz.

create-resource-group

Das folgende Codebeispiel zeigt die Verwendung `create-resource-group`.

AWS CLI

Um eine Ressourcengruppe zu erstellen

Der folgende `create-resource-group` Befehl erstellt eine Ressourcengruppe mit dem Tag-Schlüssel `Name` und dem Wert `vonexample`:

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

Ausgabe:

```
{
  "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv"
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [CreateResourceGroup](#) in der AWS CLI Befehlsreferenz.

create-sbom-export

Das folgende Codebeispiel zeigt die Verwendung `create-sbom-export`.

AWS CLI

Um einen Software-Stücklistenbericht (SBOM) zu erstellen

Im folgenden `create-sbom-export` Beispiel wird ein Software-Stücklistenbericht (SBOM) erstellt.

```
aws inspector2 create-sbom-export \
  --report-format SPDX_2_3 \
  --resource-filter-criteria
  'ecrRepositoryName=[{comparison="EQUALS",value="debian"}]' \
```

```
--s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-  
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE33333
```

Ausgabe:

```
{  
  "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
}
```

Weitere Informationen finden Sie unter [Exportieren SBOMs mit Amazon Inspector](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSbomExport](#) in der AWS CLI Befehlsreferenz.

delete-assessment-run

Das folgende Codebeispiel zeigt die Verwendung `delete-assessment-run`.

AWS CLI

Um einen Testlauf zu löschen

Mit dem folgenden `delete-assessment-run` Befehl wird der Bewertungslauf mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe` gelöscht:

```
aws inspector delete-assessment-run --assessment-run-arn arn:aws:inspector:us-  
west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DeleteAssessmentRun](#) in der AWS CLI Befehlsreferenz.

delete-assessment-target

Das folgende Codebeispiel zeigt die Verwendung `delete-assessment-target`.

AWS CLI

So löschen Sie ein Bewertungsziel

Der folgende `delete-assessment-target` Befehl löscht das Bewertungsziel mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`:

```
aws inspector delete-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [DeleteAssessmentTarget](#) in der AWS CLI Befehlsreferenz.

delete-assessment-template

Das folgende Codebeispiel zeigt die Verwendung `delete-assessment-template`.

AWS CLI

So löschen Sie eine Bewertungsvorlage

Mit dem folgenden `delete-assessment-template` Befehl wird die Bewertungsvorlage mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T` gelöscht:

```
aws inspector delete-assessment-template --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DeleteAssessmentTemplate](#) in der AWS CLI Befehlsreferenz.

delete-filter

Das folgende Codebeispiel zeigt die Verwendung `delete-filter`.

AWS CLI

Um einen Filter zu löschen

Im folgenden `delete-filter` Beispiel wird ein Filter gelöscht.

```
aws inspector2 delete-filter \
```

```
--arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444"
```

Ausgabe:

```
{
  "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444"
}
```

Weitere Informationen finden Sie unter [Filtern von Amazon Inspector Inspector-Ergebnissen](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteFilter](#) in der AWS CLI Befehlsreferenz.

describe-assessment-runs

Das folgende Codebeispiel zeigt die Verwendung `describe-assessment-runs`.

AWS CLI

Zur Beschreibung von Bewertungsläufen

Der folgende `describe-assessment-run` Befehl beschreibt einen Bewertungslauf mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE`:

```
aws inspector describe-assessment-runs --assessment-run-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

Ausgabe:

```
{
  "assessmentRuns": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
      "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw",
      "completedAt": 1458680301.4,
      "createdAt": 1458680170.035,
      "dataCollected": true,
    }
  ]
}
```

```
"durationInSeconds": 3600,
"name": "Run 1 for ExampleAssessmentTemplate",
"notifications": [],
"rulesPackageArns": [
  "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
],
"startedAt": 1458680170.161,
"state": "COMPLETED",
"stateChangedAt": 1458680301.4,
"stateChanges": [
  {
    "state": "CREATED",
    "stateChangedAt": 1458680170.035
  },
  {
    "state": "START_DATA_COLLECTION_PENDING",
    "stateChangedAt": 1458680170.065
  },
  {
    "state": "START_DATA_COLLECTION_IN_PROGRESS",
    "stateChangedAt": 1458680170.096
  },
  {
    "state": "COLLECTING_DATA",
    "stateChangedAt": 1458680170.161
  },
  {
    "state": "STOP_DATA_COLLECTION_PENDING",
    "stateChangedAt": 1458680239.883
  },
  {
    "state": "DATA_COLLECTED",
    "stateChangedAt": 1458680299.847
  },
  {
    "state": "EVALUATING_RULES",
    "stateChangedAt": 1458680300.099
  },
  {
    "state": "COMPLETED",
    "stateChangedAt": 1458680301.4
  }
],
"userAttributesForFindings": []
```

```
    }
  ],
  "failedItems": {}
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DescribeAssessmentRuns](#) in der AWS CLI Befehlsreferenz.

describe-assessment-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-assessment-targets`.

AWS CLI

Um die Bewertungsziele zu beschreiben

Der folgende `describe-assessment-targets` Befehl beschreibt das Bewertungsziel mit dem ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`:

```
aws inspector describe-assessment-targets --assessment-target-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

Ausgabe:

```
{
  "assessmentTargets": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq",
      "createdAt": 1458074191.459,
      "name": "ExampleAssessmentTarget",
      "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI",
      "updatedAt": 1458074191.459
    }
  ],
  "failedItems": {}
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [DescribeAssessmentTargets](#) in der AWS CLI Befehlsreferenz.

describe-assessment-templates

Das folgende Codebeispiel zeigt die Verwendung von `describe-assessment-templates`.

AWS CLI

Um Bewertungsvorlagen zu beschreiben

Der folgende `describe-assessment-templates` Befehl beschreibt die Bewertungsvorlage mit dem ARN `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw`:

```
aws inspector describe-assessment-templates --assessment-template-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw
```

Ausgabe:

```
{
  "assessmentTemplates": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw",
      "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq",
      "createdAt": 1458074191.844,
      "durationInSeconds": 3600,
      "name": "ExampleAssessmentTemplate",
      "rulesPackageArns": [
        "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
      ],
      "userAttributesForFindings": []
    }
  ],
  "failedItems": {}
}
```

Weitere Informationen finden Sie unter [Amazon Inspector Assessment Templates and Assessment Runs](#) im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DescribeAssessmentTemplates](#) in der AWS CLI Befehlsreferenz.

describe-cross-account-access-role

Das folgende Codebeispiel zeigt die Verwendung `describe-cross-account-access-role`.

AWS CLI

Um die Rolle für den kontoübergreifenden Zugriff zu beschreiben

Der folgende `describe-cross-account-access-role` Befehl beschreibt die IAM Rolle, die Amazon Inspector den Zugriff auf Ihr AWS Konto ermöglicht:

```
aws inspector describe-cross-account-access-role
```

Ausgabe:

```
{
  "registeredAt": 1458069182.826,
  "roleArn": "arn:aws:iam::123456789012:role/inspector",
  "valid": true
}
```

Weitere Informationen finden Sie unter Amazon Inspector einrichten im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DescribeCrossAccountAccessRole](#) in der AWS CLI Befehlsreferenz.

describe-findings

Das folgende Codebeispiel zeigt die Verwendung `describe-findings`.

AWS CLI

Um Ergebnisse zu beschreiben

Der folgende `describe-findings` Befehl beschreibt das Ergebnis mit dem ARN `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4`:

```
aws inspector describe-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4
```

Ausgabe:

```
{
  "failedItems": {},
  "findings": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
      "assetAttributes": {
        "ipv4Addresses": [],
        "schemaVersion": 1
      },
      "assetType": "ec2-instance",
      "attributes": [],
      "confidence": 10,
      "createdAt": 1458680301.37,
      "description": "Amazon Inspector did not find any potential security
issues during this assessment.",
      "indicatorOfCompromise": false,
      "numericSeverity": 0,
      "recommendation": "No remediation needed.",
      "schemaVersion": 1,
      "service": "Inspector",
      "serviceAttributes": {
        "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
        "rulesPackageArn": "arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-X1KXtawP",
        "schemaVersion": 1
      },
      "severity": "Informational",
      "title": "No potential security issues found",
      "updatedAt": 1458680301.37,
      "userAttributes": []
    }
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Findings im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DescribeFindings](#) in der AWS CLI Befehlsreferenz.

describe-resource-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-resource-groups`.

AWS CLI

Um Ressourcengruppen zu beschreiben

Der folgende `describe-resource-groups` Befehl beschreibt die Ressourcengruppe mit dem ARN `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI`:

```
aws inspector describe-resource-groups --resource-group-arns arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI
```

Ausgabe:

```
{
  "failedItems": {},
  "resourceGroups": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI",
      "createdAt": 1458074191.098,
      "tags": [
        {
          "key": "Name",
          "value": "example"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [DescribeResourceGroups](#) in der AWS CLI Befehlsreferenz.

describe-rules-packages

Das folgende Codebeispiel zeigt die Verwendung `describe-rules-packages`.

AWS CLI

Um Regelpakete zu beschreiben

Der folgende `describe-rules-packages` Befehl beschreibt das Regelpaket mit dem ARN `arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p`:

```
aws inspector describe-rules-packages --rules-package-arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p
```

Ausgabe:

```
{
  "failedItems": {},
  "rulesPackages": [
    {
      "arn": "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
      "description": "The rules in this package help verify whether the EC2 instances in your application are exposed to Common Vulnerabilities and Exposures (CVEs). Attacks can exploit unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of your service or data. The CVE system provides a reference for publicly known information security vulnerabilities and exposures. For more information, see [https://cve.mitre.org/](https://cve.mitre.org/). If a particular CVE appears in one of the produced Findings at the end of a completed Inspector assessment, you can search [https://cve.mitre.org/](https://cve.mitre.org/) using the CVE's ID (for example, \"CVE-2009-0021\") to find detailed information about this CVE, its severity, and how to mitigate it. ",
      "name": "Common Vulnerabilities and Exposures",
      "provider": "Amazon Web Services, Inc.",
      "version": "1.1"
    }
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Rules Packages and Rules im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [DescribeRulesPackages](#) in der AWS CLI Befehlsreferenz.

get-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-configuration`.

AWS CLI

Um die Einstellungskonfiguration für Inspector-Scans abzurufen

Im folgenden `get-configuration` Beispiel wird die Einstellungskonfiguration für Inspector-Scans abgerufen.

```
aws inspector2 get-configuration
```

Ausgabe:

```
{
  "ec2Configuration": {
    "scanModeState": {
      "scanMode": "EC2_HYBRID",
      "scanModeStatus": "SUCCESS"
    }
  },
  "ecrConfiguration": {
    "rescanDurationState": {
      "pullDateRescanDuration": "DAYS_90",
      "rescanDuration": "DAYS_30",
      "status": "SUCCESS",
      "updatedAt": "2024-05-14T21:16:20.237000+00:00"
    }
  }
}
```

Weitere Informationen finden Sie unter [Automatisiertes Scannen von Ressourcen mit Amazon Inspector](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [GetConfiguration](#) in der AWS CLI Befehlsreferenz.

get-telemetry-metadata

Das folgende Codebeispiel zeigt die Verwendung `get-telemetry-metadata`.

AWS CLI

Um die Telemetrie-Metadaten abzurufen

Der folgende `get-telemetry-metadata` Befehl generiert Informationen zu den Daten, die für den Bewertungslauf mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE` gesammelt wurden:

```
aws inspector get-telemetry-metadata --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

Ausgabe:

```
{
  "telemetryMetadata": [
    {
      "count": 2,
      "dataSize": 345,
      "messageType": "InspectorDuplicateProcess"
    },
    {
      "count": 3,
      "dataSize": 255,
      "messageType": "InspectorTimeEventMsg"
    },
    {
      "count": 4,
      "dataSize": 1082,
      "messageType": "InspectorNetworkInterface"
    },
    {
      "count": 2,
      "dataSize": 349,
      "messageType": "InspectorDnsEntry"
    },
    {
      "count": 11,
      "dataSize": 2514,
      "messageType": "InspectorDirectoryInfoMsg"
    },
    {
      "count": 1,
```

```
    "dataSize": 179,  
    "messageType": "InspectorTcpV6ListeningPort"  
  },  
  {  
    "count": 101,  
    "dataSize": 10949,  
    "messageType": "InspectorTerminal"  
  },  
  {  
    "count": 26,  
    "dataSize": 5916,  
    "messageType": "InspectorUser"  
  },  
  {  
    "count": 282,  
    "dataSize": 32148,  
    "messageType": "InspectorDynamicallyLoadedCodeModule"  
  },  
  {  
    "count": 18,  
    "dataSize": 10172,  
    "messageType": "InspectorCreateProcess"  
  },  
  {  
    "count": 3,  
    "dataSize": 8001,  
    "messageType": "InspectorProcessPerformance"  
  },  
  {  
    "count": 1,  
    "dataSize": 360,  
    "messageType": "InspectorOperatingSystem"  
  },  
  {  
    "count": 6,  
    "dataSize": 546,  
    "messageType": "InspectorStopProcess"  
  },  
  {  
    "count": 1,  
    "dataSize": 1553,  
    "messageType": "InspectorInstanceMetaData"  
  },  
  {
```

```
    "count": 2,  
    "dataSize": 434,  
    "messageType": "InspectorTcpV4Connection"  
  },  
  {  
    "count": 474,  
    "dataSize": 2960322,  
    "messageType": "InspectorPackageInfo"  
  },  
  {  
    "count": 3,  
    "dataSize": 2235,  
    "messageType": "InspectorSystemPerformance"  
  },  
  {  
    "count": 105,  
    "dataSize": 46048,  
    "messageType": "InspectorCodeModule"  
  },  
  {  
    "count": 1,  
    "dataSize": 182,  
    "messageType": "InspectorUdpV6ListeningPort"  
  },  
  {  
    "count": 2,  
    "dataSize": 371,  
    "messageType": "InspectorUdpV4ListeningPort"  
  },  
  {  
    "count": 18,  
    "dataSize": 8362,  
    "messageType": "InspectorKernelModule"  
  },  
  {  
    "count": 29,  
    "dataSize": 48788,  
    "messageType": "InspectorConfigurationInfo"  
  },  
  {  
    "count": 1,  
    "dataSize": 79,  
    "messageType": "InspectorMonitoringStart"  
  },  
}
```

```
{
  {
    "count": 5,
    "dataSize": 0,
    "messageType": "InspectorSplitMsgBegin"
  },
  {
    "count": 51,
    "dataSize": 4593,
    "messageType": "InspectorGroup"
  },
  {
    "count": 1,
    "dataSize": 184,
    "messageType": "InspectorTcpV4ListeningPort"
  },
  {
    "count": 1159,
    "dataSize": 3146579,
    "messageType": "Total"
  },
  {
    "count": 5,
    "dataSize": 0,
    "messageType": "InspectorSplitMsgEnd"
  },
  {
    "count": 1,
    "dataSize": 612,
    "messageType": "InspectorLoadImageInProgress"
  }
]
}
```

- API-Einheiten finden Sie [GetTelemetryMetadata](#) unter AWS CLI Befehlsreferenz.

list-account-permissions

Das folgende Codebeispiel zeigt die Verwendung `list-account-permissions`.

AWS CLI

Um Kontoberechtigungen aufzulisten

Das folgende `list-account-permissions` Beispiel listet Ihre Kontoberechtigungen auf.

```
aws inspector2 list-account-permissions
```

Ausgabe:

```
{
  "permissions": [
    {
      "operation": "ENABLE_SCANNING",
      "service": "ECR"
    },
    {
      "operation": "DISABLE_SCANNING",
      "service": "ECR"
    },
    {
      "operation": "ENABLE_REPOSITORY",
      "service": "ECR"
    },
    {
      "operation": "DISABLE_REPOSITORY",
      "service": "ECR"
    },
    {
      "operation": "ENABLE_SCANNING",
      "service": "EC2"
    },
    {
      "operation": "DISABLE_SCANNING",
      "service": "EC2"
    },
    {
      "operation": "ENABLE_SCANNING",
      "service": "LAMBDA"
    },
    {
      "operation": "DISABLE_SCANNING",
      "service": "LAMBDA"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Identity and Access Management für Amazon Inspector](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAccountPermissions](#) unter AWS CLI Befehlsreferenz.

list-assessment-run-agents

Das folgende Codebeispiel zeigt die Verwendung `list-assessment-run-agents`.

AWS CLI

Um die Bewertung aufzulisten, führen Sie Agenten aus.

Der folgende `list-assessment-run-agents` Befehl listet die Agenten des Testlaufs mit den angegebenen auf ARN.

```
aws inspector list-assessment-run-agents \
  --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
  template/0-4r1V2mAw/run/0-MKkpXXPE
```

Ausgabe:

```
{
  "assessmentRunAgents": [
    {
      "agentHealth": "HEALTHY",
      "agentHealthCode": "HEALTHY",
      "agentId": "i-49113b93",
      "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
      "telemetryMetadata": [
        {
          "count": 2,
          "dataSize": 345,
          "messageType": "InspectorDuplicateProcess"
        },
        {
          "count": 3,
          "dataSize": 255,
          "messageType": "InspectorTimeEventMsg"
        },
        {
          "count": 4,
```



```
        "dataSize": 1082,
        "messageType": "InspectorNetworkInterface"
    },
    {
        "count": 2,
        "dataSize": 349,
        "messageType": "InspectorDnsEntry"
    },
    {
        "count": 11,
        "dataSize": 2514,
        "messageType": "InspectorDirectoryInfoMsg"
    },
    {
        "count": 1,
        "dataSize": 179,
        "messageType": "InspectorTcpV6ListeningPort"
    },
    {
        "count": 101,
        "dataSize": 10949,
        "messageType": "InspectorTerminal"
    },
    {
        "count": 26,
        "dataSize": 5916,
        "messageType": "InspectorUser"
    },
    {
        "count": 282,
        "dataSize": 32148,
        "messageType": "InspectorDynamicallyLoadedCodeModule"
    },
    {
        "count": 18,
        "dataSize": 10172,
        "messageType": "InspectorCreateProcess"
    },
    {
        "count": 3,
        "dataSize": 8001,
        "messageType": "InspectorProcessPerformance"
    },
    {
```



```
{
  "count": 18,
  "dataSize": 8362,
  "messageType": "InspectorKernelModule"
},
{
  "count": 29,
  "dataSize": 48788,
  "messageType": "InspectorConfigurationInfo"
},
{
  "count": 1,
  "dataSize": 79,
  "messageType": "InspectorMonitoringStart"
},
{
  "count": 5,
  "dataSize": 0,
  "messageType": "InspectorSplitMsgBegin"
},
{
  "count": 51,
  "dataSize": 4593,
  "messageType": "InspectorGroup"
},
{
  "count": 1,
  "dataSize": 184,
  "messageType": "InspectorTcpV4ListeningPort"
},
{
  "count": 1159,
  "dataSize": 3146579,
  "messageType": "Total"
},
{
  "count": 5,
  "dataSize": 0,
  "messageType": "InspectorSplitMsgEnd"
},
{
  "count": 1,
  "dataSize": 612,
  "messageType": "InspectorLoadImageInProgress"
}
```

```
}
  ]
}
]
```

Weitere Informationen finden Sie unter [AWS Agents](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAssessmentRunAgents](#) in der AWS CLI Befehlsreferenz.

list-assessment-runs

Das folgende Codebeispiel zeigt die Verwendung `list-assessment-runs`.

AWS CLI

Um Bewertungsläufe aufzulisten

Der folgende `list-assessment-runs` Befehl listet alle vorhandenen Bewertungsläufe auf.

```
aws inspector list-assessment-runs
```

Ausgabe:

```
{
  "assessmentRunArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE",
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v"
  ]
}
```

Weitere Informationen finden Sie unter [Amazon Inspector Assessment Templates and Assessment Runs](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAssessmentRuns](#) in der AWS CLI Befehlsreferenz.

list-assessment-targets

Das folgende Codebeispiel zeigt die Verwendung `list-assessment-targets`.

AWS CLI

Um die Bewertungsziele aufzulisten

Der folgende `list-assessment-targets` Befehl listet alle vorhandenen Bewertungsziele auf:

```
aws inspector list-assessment-targets
```

Ausgabe:

```
{
  "assessmentTargetArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [ListAssessmentTargets](#) in der AWS CLI Befehlsreferenz.

list-assessment-templates

Das folgende Codebeispiel zeigt die Verwendung `list-assessment-templates`.

AWS CLI

Um Bewertungsvorlagen aufzulisten

Der folgende `list-assessment-templates` Befehl listet alle vorhandenen Bewertungsvorlagen auf:

```
aws inspector list-assessment-templates
```

Ausgabe:

```
{
  "assessmentTemplateArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
    template/0-4r1V2mAw",
  ]
}
```

```
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-
    Uza6ihLh"
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [ListAssessmentTemplates](#) in der AWS CLI Befehlsreferenz.

list-coverage-statistics

Das folgende Codebeispiel zeigt die Verwendung `list-coverage-statistics`.

AWS CLI

Beispiel 1: Um Deckungsstatistiken nach Gruppen aufzulisten

Das folgende `list-coverage-statistics` Beispiel listet die Abdeckungsstatistiken Ihrer AWS Umgebung nach Gruppen auf.

```
aws inspector2 list-coverage-statistics \
  --group-by RESOURCE_TYPE
```

Ausgabe:

```
{
  "countsByGroup": [
    {
      "count": 56,
      "groupKey": "AWS_LAMBDA_FUNCTION"
    },
    {
      "count": 27,
      "groupKey": "AWS_ECR_REPOSITORY"
    },
    {
      "count": 18,
      "groupKey": "AWS_EC2_INSTANCE"
    },
    {
      "count": 3,
```

```

        "groupKey": "AWS_ECR_CONTAINER_IMAGE"
      },
      {
        "count": 1,
        "groupKey": "AWS_ACCOUNT"
      }
    ],
    "totalCounts": 105
  }

```

Weitere Informationen finden Sie unter [Bewertung der Amazon Inspector Inspector-Abdeckung Ihrer AWS Umgebung](#) im Amazon Inspector Inspector-Benutzerhandbuch.

Beispiel 2: Um Statistiken zur Netzabdeckung nach Ressourcentyp aufzulisten

Das folgende `list-coverage-statistics` Beispiel listet die Deckungsstatistiken Ihrer AWS Umgebung nach Ressourcentyp auf.

```

aws inspector2 list-coverage-statistics
  --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_ECR_REPOSITORY"}]}'
  --group-by SCAN_STATUS_REASON

```

Ausgabe:

```

{
  "countsByGroup": [
    {
      "count": 27,
      "groupKey": "SUCCESSFUL"
    }
  ],
  "totalCounts": 27
}

```

Weitere Informationen finden Sie unter [Bewertung der Amazon Inspector Inspector-Abdeckung Ihrer AWS Umgebung](#) im Amazon Inspector Inspector-Benutzerhandbuch.

Beispiel 3: Um Deckungsstatistiken nach ECR Repository-Namen aufzulisten

Das folgende `list-coverage-statistics` Beispiel listet die Deckungsstatistiken Ihrer AWS Umgebung nach ECR Repository-Namen auf.

```
aws inspector2 list-coverage-statistics
  --filter-criteria '{"ecrRepositoryName":
[{"comparison":"EQUALS","value":"debian"}]}'
  --group-by SCAN_STATUS_REASON
```

Ausgabe:

```
{
  "countsByGroup": [
    {
      "count": 3,
      "groupKey": "SUCCESSFUL"
    }
  ],
  "totalCounts": 3
}
```

Weitere Informationen finden Sie unter [Bewertung der Amazon Inspector Inspector-Abdeckung Ihrer AWS Umgebung](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [ListCoverageStatistics](#) in der AWS CLI Befehlsreferenz.

list-coverage

Das folgende Codebeispiel zeigt die Verwendung `list-coverage`.

AWS CLI

Beispiel 1: Um Details zur Abdeckung Ihrer Umgebung aufzulisten

Das folgende `list-coverage` Beispiel listet die Deckungsdetails Ihrer Umgebung auf.

```
aws inspector2 list-coverage
```

Ausgabe:

```
{
  "coveredResources": [
    {
      "accountId": "123456789012",
```



```

    "lastScannedAt": "2024-05-20T16:23:20-07:00",
    "resourceId": "i-EXAMPLE5555555555",
    "resourceMetadata": {
      "ec2": {
        "amiId": "ami-EXAMPLE6666666666",
        "platform": "LINUX"
      }
    },
    "resourceType": "AWS_EC2_INSTANCE",
    "scanStatus": {
      "reason": "SUCCESSFUL",
      "statusCode": "ACTIVE"
    },
    "scanType": "PACKAGE"
  }
]
}

```

Beispiel 2: Um Deckungsdetails zum Ressourcentyp der Lambda-Funktion aufzulisten

Im folgenden `list-coverage` Beispiel werden die Details zum Ressourcentyp Ihrer Lambda-Funktion aufgeführt.

```

aws inspector2 list-coverage
  --filter-criteria '{"resourceType":
[{"comparison": "EQUALS", "value": "AWS_LAMBDA_FUNCTION"}]}'

```

Ausgabe:

```

{
  "coveredResources": [
    {
      "accountId": "123456789012",
      "resourceId": "arn:aws:lambda:us-west-2:123456789012:function:Eval-
container-scan-results:$LATEST",
      "resourceMetadata": {
        "lambdaFunction": {
          "functionName": "Eval-container-scan-results",
          "functionTags": {},
          "layers": [],
          "runtime": "PYTHON_3_7"
        }
      }
    }
  ]
}

```

```

    },
    "resourceType": "AWS_LAMBDA_FUNCTION",
    "scanStatus": {
      "reason": "SUCCESSFUL",
      "statusCode": "ACTIVE"
    },
    "scanType": "CODE"
  }
]
}

```

- API-Einheiten finden Sie [ListCoverage](#) in der AWS CLI Befehlsreferenz.

list-delegated-admin-accounts

Das folgende Codebeispiel zeigt die Verwendung `list-delegated-admin-accounts`.

AWS CLI

Um Informationen über das delegierte Administratorkonto Ihrer Organisation aufzulisten

Im folgenden `list-delegated-admin-accounts` Beispiel werden Informationen zum delegierten Administratorkonto Ihrer Organisation aufgeführt.

```
aws inspector2 list-delegated-admin-accounts
```

Ausgabe:

```

{
  "delegatedAdminAccounts": [
    {
      "accountId": "123456789012",
      "status": "ENABLED"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Benennen eines delegierten Administrators für Amazon Inspector im Amazon Inspector](#) Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListDelegatedAdminAccounts AWS CLI](#) Befehlsreferenz.

AWS CLI

Um Filter aufzulisten, die mit dem Konto verknüpft sind, mit dem Sie Amazon Inspector aktiviert haben

In den folgenden `list-filters` Beispielen sind Filter aufgeführt, die mit dem Konto verknüpft sind, mit dem Sie Amazon Inspector aktiviert haben.

```
aws inspector2 list-filters
```

Ausgabe:

```
{
  "filters": [
    {
      "action": "SUPPRESS",
      "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444",
      "createdAt": "2024-05-15T21:11:08.602000+00:00",
      "criteria": {
        "resourceType": [
          {
            "comparison": "EQUALS",
            "value": "AWS_EC2_INSTANCE"
          }
        ]
      },
      "description": "This suppression rule omits EC2 instance type findings",
      "name": "ExampleSuppressionRuleEC2",
      "ownerId": "o-EXAMPLE222",
      "tags": {},
      "updatedAt": "2024-05-15T21:11:08.602000+00:00"
    },
    {
      "action": "SUPPRESS",
      "arn": "arn:aws:inspector2:us-east-1:813737243517:owner/o-EXAMPLE222/filter/EXAMPLE4444444444",
      "createdAt": "2024-05-15T21:28:27.054000+00:00",
      "criteria": {
        "resourceType": [
          {
            "comparison": "EQUALS",
            "value": "AWS_ECR_INSTANCE"
          }
        ]
      }
    }
  ]
}
```

```
        }
      ]
    },
    "description": "This suppression rule omits ECR instance type findings",
    "name": "ExampleSuppressionRuleECR",
    "ownerId": "o-EXAMPLE222",
    "tags": {},
    "updatedAt": "2024-05-15T21:28:27.054000+00:00"
  }
]
}
```

Weitere Informationen finden Sie unter [Filtern von Amazon Inspector Inspector-Ergebnissen](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [ListFilters](#) in der AWS CLI Befehlsreferenz.

list-findings

Das folgende Codebeispiel zeigt die Verwendung `list-findings`.

AWS CLI

Um Ergebnisse aufzulisten

Der folgende `list-findings` Befehl listet alle generierten Ergebnisse auf:

```
aws inspector list-findings
```

Ausgabe:

```
{
  "findingArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v/finding/0-tyvmqBLy"
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Findings im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [ListFindings](#) in der AWS CLI Befehlsreferenz.

list-rules-packages

Das folgende Codebeispiel zeigt die Verwendung `list-rules-packages`.

AWS CLI

Um Regelpakete aufzulisten

Der folgende `list-rules-packages` Befehl listet alle verfügbaren Inspector-Regelpakete auf:

```
aws inspector list-rules-packages
```

Ausgabe:

```
{
  "rulesPackageArns": [
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-H5hpSawc",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-JJ0tZiqQ",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-vg5GGHSD"
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Rules Packages and Rules im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [ListRulesPackages](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für Ressourcen aufzulisten

Der folgende `list-tags-for-resource` Befehl listet alle mit der Bewertungsvorlage verknüpften Tags mit dem Wert ARN von auf auf auf auf auf auf auf auf

```
arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu:
```

```
aws inspector list-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFLiYu
```

Ausgabe:

```
{
  "tags": [
    {
      "key": "Name",
      "value": "Example"
    }
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-usage-totals

Das folgende Codebeispiel zeigt die Verwendung `list-usage-totals`.

AWS CLI

Um die Gesamtnutzungswerte der letzten 30 Tage aufzulisten

In den folgenden `list-usage-totals` Beispielen werden die Gesamtnutzungswerte der letzten 30 Tage aufgeführt.

```
aws inspector2 list-usage-totals
```

Ausgabe:

```
{
  "totals": [
    {
      "accountId": "123456789012",
      "usage": [
        {
```

```

    "currency": "USD",
    "estimatedMonthlyCost": 4.6022044647,
    "total": 1893.4784083333334,
    "type": "EC2_AGENTLESS_INSTANCE_HOURS"
  },
  {
    "currency": "USD",
    "estimatedMonthlyCost": 18.892449279,
    "total": 10882.050784722222,
    "type": "EC2_INSTANCE_HOURS"
  },
  {
    "currency": "USD",
    "estimatedMonthlyCost": 5.4525363736,
    "total": 6543.043648333333,
    "type": "LAMBDA_FUNCTION_CODE_HOURS"
  },
  {
    "currency": "USD",
    "estimatedMonthlyCost": 3.9064080309,
    "total": 9375.379274166668,
    "type": "LAMBDA_FUNCTION_HOURS"
  },
  {
    "currency": "USD",
    "estimatedMonthlyCost": 0.06,
    "total": 6.0,
    "type": "ECR_RESCAN"
  },
  {
    "currency": "USD",
    "estimatedMonthlyCost": 0.09,
    "total": 1.0,
    "type": "ECR_INITIAL_SCAN"
  }
]
}
]
}

```

Weitere Informationen finden Sie unter [Überwachung der Nutzung und der Kosten in Amazon Inspector](#) im Amazon Inspector Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [ListUsageTotals](#) in der AWS CLI Befehlsreferenz.

preview-agents

Das folgende Codebeispiel zeigt die Verwendung `preview-agents`.

AWS CLI

Um eine Vorschau der Agenten anzuzeigen

Mit dem folgenden `preview-agents` Befehl wird eine Vorschau der Agenten angezeigt, die auf den EC2 Instances installiert sind, die Teil des Bewertungsziels sind, mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`:

```
aws inspector preview-agents --preview-agents-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

Ausgabe:

```
{
  "agentPreviews": [
    {
      "agentId": "i-49113b93"
    }
  ]
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [PreviewAgents](#) in der AWS CLI Befehlsreferenz.

register-cross-account-access-role

Das folgende Codebeispiel zeigt die Verwendung `register-cross-account-access-role`.

AWS CLI

Um die Rolle für den kontoübergreifenden Zugriff zu registrieren

Der folgende `register-cross-account-access-role` Befehl registriert die IAM Rolle bei ARN `arn:aws:iam::123456789012:role/inspector`, die Amazon Inspector verwendet, um Ihre EC2 Instances zu Beginn des Bewertungslaufs aufzulisten oder wenn Sie den Befehl `preview-agents` aufrufen:

```
aws inspector register-cross-account-access-role --role-arn arn:aws:iam::123456789012:role/inspector
```

Weitere Informationen finden Sie unter Amazon Inspector einrichten im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [RegisterCrossAccountAccessRole](#) in der AWS CLI Befehlsreferenz.

remove-attributes-from-findings

Das folgende Codebeispiel zeigt die Verwendung `remove-attributes-from-findings`.

AWS CLI

Um Attribute aus Ergebnissen zu entfernen

Der folgende `remove-attributes-from-finding` Befehl entfernt das Attribut mit dem Schlüssel `Example` und dem Wert von `example` aus dem Ergebnis mit dem Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU`:

```
aws inspector remove-attributes-from-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attribute-keys key=Example,value=example
```

Ausgabe:

```
{
  "failedItems": {}
}
```

Weitere Informationen finden Sie unter Amazon Inspector Findings im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [RemoveAttributesFromFindings](#) in der AWS CLI Befehlsreferenz.

set-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `set-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource festzulegen

Mit dem folgenden `set-tags-for-resource` Befehl wird dem Tag mit dem Schlüssel `example` und dem Wert `example` der Bewertungsvorlage der Wert ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0`:

```
aws inspector set-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0 --tags key=Example,value=example
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [SetTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-assessment-run

Das folgende Codebeispiel zeigt die Verwendung `start-assessment-run`.

AWS CLI

Um einen Testlauf zu starten

Mit dem folgenden `start-assessment-run` Befehl wird der `examplerun` anhand der Bewertungsvorlage benannte Bewertungslauf mit dem Zeichen ARN von `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T`:

```
aws inspector start-assessment-run --assessment-run-name examplerun --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

Ausgabe:

```
{
  "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0o0oxyY"
}
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- APIEinzelheiten finden Sie [StartAssessmentRun](#) in der AWS CLI Befehlsreferenz.

stop-assessment-run

Das folgende Codebeispiel zeigt die Verwendung `stop-assessment-run`.

AWS CLI

Um einen Testlauf zu beenden

Mit dem folgenden `stop-assessment-run` Befehl wird der Bewertungslauf mit dem Wert ARN von `beendetarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY`:

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Templates and Assessment Runs im Amazon Inspector Inspector-Handbuch.

- APIEinzelheiten finden Sie [StopAssessmentRun](#) in der AWS CLI Befehlsreferenz.

subscribe-to-event

Das folgende Codebeispiel zeigt die Verwendung `subscribe-to-event`.

AWS CLI

Um eine Veranstaltung zu abonnieren

Das folgende Beispiel ermöglicht das Senden von SNS Amazon-Benachrichtigungen über das `ASSESSMENT_RUN_COMPLETED` Ereignis an das Thema mit ARN dem `arn:aws:sns:us-west-2:123456789012:exampletopic`

```
aws inspector subscribe-to-event \  
  --event ASSESSMENT_RUN_COMPLETED \  
  --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0 \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:exampletopic
```

```
--topic-arn arn:aws:sns:us-west-2:123456789012:exampletopic
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon Inspector Assessment Templates and Assessment Runs](#) im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [SubscribeToEvent](#) in der AWS CLI Befehlsreferenz.

unsubscribe-from-event

Das folgende Codebeispiel zeigt die Verwendung `unsubscribe-from-event`.

AWS CLI

Um sich von einer Veranstaltung abzumelden

Der folgende `unsubscribe-from-event` Befehl deaktiviert das Senden von SNS Amazon-Benachrichtigungen über das `ASSESSMENT_RUN_COMPLETED` Ereignis an das Thema mit dem ARN `arn:aws:sns:us-west-2:123456789012:exampletopic`:

```
aws inspector unsubscribe-from-event --event ASSESSMENT_RUN_COMPLETED --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0  
--topic arn:aws:sns:us-west-2:123456789012:exampletopic
```

Weitere Informationen finden Sie unter [Amazon Inspector Assessment Templates and Assessment Runs](#) im Amazon Inspector Inspector-Handbuch.

- API-Einheiten finden Sie [UnsubscribeFromEvent](#) in der AWS CLI Befehlsreferenz.

update-assessment-target

Das folgende Codebeispiel zeigt die Verwendung `update-assessment-target`.

AWS CLI

Um ein Bewertungsziel zu aktualisieren

Der folgende `update-assessment-target` Befehl aktualisiert das Bewertungsziel mit dem ARN Namen von `Example` und die Ressourcengruppe mit dem ARN `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt:arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX`

```
aws inspector update-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-target-name Example --resource-group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt
```

Weitere Informationen finden Sie unter Amazon Inspector Assessment Targets im Amazon Inspector Inspector-Leitfaden.

- API-Einheiten finden Sie [UpdateAssessmentTarget](#) in der AWS CLI Befehlsreferenz.

update-filter

Das folgende Codebeispiel zeigt die Verwendung `update-filter`.

AWS CLI

Um einen Filter zu aktualisieren

Im folgenden `update-filter` Beispiel wird ein Filter aktualisiert, sodass Lambda-Ergebnisse anstelle von ECR Instanzergebnissen weggelassen werden.

```
aws inspector2 update-filter \
  --filter-arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444" \
  --name "ExampleSuppressionRuleLambda" \
  --description "This suppression rule omits Lambda instance findings" \
  --reason "Updating filter to omit Lambda instance findings instead of ECR instance findings"
```

Ausgabe:

```
{
  "filters": [
    {
      "action": "SUPPRESS",
      "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444",
      "createdAt": "2024-05-15T21:28:27.054000+00:00",
      "criteria": {
        "resourceType": [
          {
            "comparison": "EQUALS",
```

```
        "value": "AWS_ECR_INSTANCE"
      }
    ]
  },
  "description": "This suppression rule omits Lambda instance findings",
  "name": "ExampleSuppressionRuleLambda",
  "ownerId": "o-EXAMPLE222",
  "reason": "Updating filter to omit Lambda instance findings instead of
ECR instance findings",
  "tags": {},
  "updatedAt": "2024-05-15T22:23:13.665000+00:00"
}
]
```

Weitere Informationen finden Sie unter [Ergebnisse in Amazon Inspector verwalten im Amazon Inspector](#) Inspector-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateFilter](#) in der AWS CLI Befehlsreferenz.

AWS IoT Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-certificate-transfer

Das folgende Codebeispiel zeigt die Verwendung `accept-certificate-transfer`.

AWS CLI

Um ein Gerätezertifikat zu akzeptieren, das von einem anderen AWS Konto übertragen wurde

Das folgende `accept-certificate-transfer` Beispiel akzeptiert ein Gerätezertifikat, das von einem anderen AWS Konto übertragen wurde. Das Zertifikat wird anhand seiner ID identifiziert.

```
aws iot accept-certificate-transfer \  
  --certificate-  
id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Übertragen eines Zertifikats auf ein anderes Konto](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [AcceptCertificateTransfer](#) in der AWS CLI Befehlsreferenz.

add-thing-to-billing-group

Das folgende Codebeispiel zeigt die Verwendung `add-thing-to-billing-group`.

AWS CLI

Beispiel 1: Um einer Abrechnungsgruppe eine Sache nach Namen hinzuzufügen

Im folgenden `add-thing-to-billing-group` Beispiel wird das Objekt mit `MyLightBulb` dem Namen der angegebenen Abrechnungsgruppe hinzugefügt `GroupOne`.

```
aws iot add-thing-to-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyLightBulb
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einer Abrechnungsgruppe ein Ding von ARN hinzuzufügen

Im folgenden `add-thing-to-billing-group` Beispiel wird einer Abrechnungsgruppe mit ARN dem angegebenen Wert eine Sache mit einem bestimmten Wert hinzugefügt `ARN`. Die Angabe von ARN ist hilfreich, wenn Sie mit mehreren AWS Regionen oder Konten arbeiten. Auf diese Weise können Sie sicherstellen, dass Sie zur richtigen Region und zum richtigen Konto hinzufügen.


```
aws iot add-thing-to-thing-group \  
  --billing-group-arn "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne" \  
  --thing-arn "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AddThingToBillingGroup](#) in der AWS CLI Befehlsreferenz.

add-thing-to-thing-group

Das folgende Codebeispiel zeigt die Verwendung `add-thing-to-thing-group`.

AWS CLI

Um einer Gruppe etwas hinzuzufügen

Das folgende `add-thing-to-thing-group` Beispiel fügt das angegebene Ding der angegebenen Dinggruppe hinzu.

```
aws iot add-thing-to-thing-group \  
  --thing-name MyLightBulb \  
  --thing-group-name LightBulbs
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AddThingToThingGroup](#) in der AWS CLI Befehlsreferenz.

associate-targets-with-job

Das folgende Codebeispiel zeigt die Verwendung `associate-targets-with-job`.

AWS CLI

Um eine Dinggruppe einem fortlaufenden Job zuzuordnen

Im folgenden `associate-targets-with-job` Beispiel wird die angegebene Dinggruppe dem angegebenen kontinuierlichen Auftrag zugeordnet.

```
aws iot associate-targets-with-job \  
  --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --job-id "example-job-04"
```

Ausgabe:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [AssociateTargetsWithJob](#) in der AWS CLI Befehlsreferenz.

attach-policy

Das folgende Codebeispiel zeigt die Verwendung `attach-policy`.

AWS CLI

Beispiel 1: Um eine Richtlinie an eine Dinggruppe anzuhängen

Im folgenden `attach-policy` Beispiel wird die angegebene Richtlinie an eine durch ihre ARN identifizierte Dinggruppe angehängt.

```
aws iot attach-policy \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --policy-name "UpdateDeviceCertPolicy"
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

Beispiel 2: So hängen Sie eine Richtlinie an ein Zertifikat an

Im folgenden `attach-policy` Beispiel wird die Richtlinie `UpdateDeviceCertPolicy` an den durch ein Zertifikat angegebenen Prinzipal angehängt.

```
aws iot attach-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --target "arn:aws:iot:us-west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Anhängen einer AWS IoT-Richtlinie an ein Gerätezertifikat](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AttachPolicy](#) in der AWS CLI Befehlsreferenz.

attach-security-profile

Das folgende Codebeispiel zeigt die Verwendung `attach-security-profile`.

AWS CLI

Um allen nicht registrierten Geräten ein Sicherheitsprofil zuzuordnen

Im folgenden `attach-security-profile` Beispiel wird das AWS angegebene IoT Device Defender-Sicherheitsprofil `Testprofile` allen nicht registrierten Geräten in der `us-west-2` Region für dieses AWS Konto zugeordnet.

```
aws iot attach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [AttachSecurityProfile](#) in der AWS CLI Befehlsreferenz.

attach-thing-principal

Das folgende Codebeispiel zeigt die Verwendung `attach-thing-principal`.

AWS CLI

Um ein Zertifikat an dein Ding anzuhängen

Im folgenden `attach-thing-principal` Beispiel wird ein Zertifikat an das `MyTemperatureSensor` Ding angehängt. Das Zertifikat wird durch einen ARN identifiziert. Sie finden das ARN für ein Zertifikat in der AWS IoT-Konsole.

```
aws iot attach-thing-principal \  
  --thing-name MyTemperatureSensor \  
  --principal arn:aws:iot:us-  
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [How to Manage Things with the Registry](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AttachThingPrincipal](#) in der AWS CLI Befehlsreferenz.

cancel-audit-mitigation-actions-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-audit-mitigation-actions-task`.

AWS CLI

Um eine Aufgabe für Überwachungsmaßnahmen abzubrechen

Im folgenden `cancel-audit-mitigation-actions-task` Beispiel wird die Anwendung von Minderungsmaßnahmen für die angegebene Aufgabe abgebrochen. Sie können Aufgaben, die bereits abgeschlossen sind, nicht stornieren.

```
aws iot cancel-audit-mitigation-actions-task  
  --task-id "myActionsTaskId"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [CancelAuditMitigationActionsTask \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CancelAuditMitigationActionsTask](#) in der AWS CLI Befehlsreferenz.

cancel-audit-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-audit-task`.

AWS CLI

Um eine Audit-Aufgabe abubrechen

Im folgenden `cancel-audit-task` Beispiel wird eine Überwachungsaufgabe mit der angegebenen Aufgaben-ID storniert. Sie können eine abgeschlossene Aufgabe nicht stornieren.

```
aws iot cancel-audit-task \  
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CancelAuditTask](#) in der AWS CLI Befehlsreferenz.

`cancel-certificate-transfer`

Das folgende Codebeispiel zeigt die Verwendung `cancel-certificate-transfer`.

AWS CLI

Um die Übertragung eines Zertifikats auf ein anderes AWS Konto abubrechen

Im folgenden `cancel-certificate-transfer` Beispiel wird die Übertragung der angegebenen Zertifikatsübertragung abgebrochen. Das Zertifikat wird durch eine Zertifikat-ID identifiziert. Sie finden die ID für ein Zertifikat in der AWS IoT-Konsole.

```
aws iot cancel-certificate-transfer \  
  --certificate-  
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78add5e605d630e05c7fc8
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Übertragen eines Zertifikats auf ein anderes Konto](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [CancelCertificateTransfer](#) in der AWS CLI Befehlsreferenz.

`cancel-job-execution`

Das folgende Codebeispiel zeigt die Verwendung `cancel-job-execution`.

AWS CLI

Um die Ausführung eines Jobs auf einem Gerät abubrechen

Im folgenden `cancel-job-execution` Beispiel wird die Ausführung des angegebenen Jobs auf einem Gerät abgebrochen. Wenn sich der Job nicht im `QUEUED` Status befindet, müssen Sie den `--force` Parameter hinzufügen.

```
aws iot cancel-job-execution \  
  --job-id "example-job-03" \  
  --thing-name "MyRPI"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CancelJobExecution](#) in der AWS CLI Befehlsreferenz.

cancel-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-job`.

AWS CLI

Um einen Job zu stornieren

Im folgenden `cancel-job` Beispiel wird der angegebene Auftrag storniert.

```
aws iot cancel-job \  
  --job-id "example-job-03"
```

Ausgabe:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",  
  "jobId": "example-job-03",  
  "description": "example job test"  
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CancelJob](#) in der AWS CLI Befehlsreferenz.

clear-default-authorizer

Das folgende Codebeispiel zeigt die Verwendung `clear-default-authorizer`.

AWS CLI

Um den Standard-Autorisierer zu löschen

Im folgenden `clear-default-authorizer` Beispiel wird der aktuell konfigurierte benutzerdefinierte Standardautorisierer gelöscht. Nachdem Sie diesen Befehl ausgeführt haben, gibt es keinen Standardautorisierer. Wenn Sie einen benutzerdefinierten Autorisierer verwenden, müssen Sie ihn in den HTTP Anforderungsheadern namentlich angeben.

```
aws iot clear-default-authorizer
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [ClearDefaultAuthorizer](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ClearDefaultAuthorizer](#) unter AWS CLI Befehlsreferenz.

confirm-topic-rule-destination

Das folgende Codebeispiel zeigt die Verwendung `confirm-topic-rule-destination`.

AWS CLI

Um das Ziel einer Themenregel zu bestätigen

Im folgenden `confirm-topic-rule-destination` Beispiel wird das Ziel einer Themenregel mit einem Bestätigungstoken bestätigt, das an einem HTTP Endpunkt empfangen wurde.

```
aws iot confirm-topic-rule-destination \  
  --confirmation-token "AYADeIcmtq-  
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREFyY1E0Um1GeDg0V21BZWZ1VjZtZWFRVUJJUkt  
aywpPqg8YEsa1LD4B40aJ2s1wEHKMybiF1Ro0ZzYisI0IvslzQY5UmCkq3tV-3f7-  
nKfosgIAAAAADAAAEEAAAAAAAAAAAAAAAAAAAAA9RMgy-  
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gT06n50NLJVpzyQR1UmPIj5sShqXEQGcC  
iufgrzTeP18RZY0Wr006Aj9DiVzJZx-1iD6Pu-
```

```
G6PUw1ka07Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseR01AnPIetJ803Z4I1I1F8xXlcdPGP-PV1d0XFemyL8g"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Bestätigen eines Ziels für Themenregeln](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ConfirmTopicRuleDestination](#) in der AWS CLI Befehlsreferenz.

create-audit-suppression

Das folgende Codebeispiel zeigt die Verwendung `create-audit-suppression`.

AWS CLI

Um ein Audit zu erstellen, das die Unterdrückung feststellt

Im folgenden `create-audit-suppression` Beispiel wird für eine Richtlinie mit dem Namen "virtualMachinePolicy", die als zu freizügig gekennzeichnet wurde, eine Unterdrückung der Prüfungsergebnisse festgestellt.

```
aws iot create-audit-suppression \
  --check-name IOT_POLICY_OVERLY_PERMISSIVE_CHECK \
  --resource-identifier
  policyVersionIdentifier={"policyName"="virtualMachinePolicy","policyVersionId"="1"}
  \
  --no-suppress-indefinitely \
  --expiration-date 2020-10-20
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit finding suppressions](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateAuditSuppression](#) in der AWS CLI Befehlsreferenz.

create-authorizer

Das folgende Codebeispiel zeigt die Verwendung `create-authorizer`.

AWS CLI

Um einen benutzerdefinierten Authorizer zu erstellen

Im folgenden `create-authorizer` Beispiel wird ein benutzerdefinierter Autorisierer erstellt, der die angegebene Lambda-Funktion als Teil eines benutzerdefinierten Authentifizierungsdienstes verwendet.

```
aws iot create-authorizer \
  --authorizer-name "CustomAuthorizer" \
  --authorizer-function-arn "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction" \
  --token-key-name "MyAuthToken" \
  --status ACTIVE \
  --token-signing-public-keys FIRST_KEY="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAAQCAQ8AMIIBCgKCAQEA1uJ0B4lQPgG/lM6ZfIwo
Z+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTagTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
zwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfPOHDJQgID0XnZwAlNnZc0hCwIx
50g2LW20y9R/dmqtDmJiVP97Z4GykxPvw1YHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
lQIDAQAB
-----END PUBLIC KEY-----"
```

Ausgabe:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

Weitere Informationen finden Sie [CreateAuthorizer](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [CreateAuthorizer](#) unter AWS CLI Befehlsreferenz.

create-billing-group

Das folgende Codebeispiel zeigt die Verwendung `create-billing-group`.

AWS CLI

Um eine Abrechnungsgruppe zu erstellen

Im folgenden `create-billing-group` Beispiel wird eine einfache Abrechnungsgruppe mit dem Namen `GroupOne` erstellt.

```
aws iot create-billing-group \  
  --billing-group-name GroupOne
```

Ausgabe:

```
{  
  "billingGroupName": "GroupOne",  
  "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",  
  "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"  
}
```

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateBillingGroup](#) in der AWS CLI Befehlsreferenz.

create-certificate-from-csr

Das folgende Codebeispiel zeigt die Verwendung `create-certificate-from-csr`.

AWS CLI

Um ein Gerätezertifikat aus einer Zertifikatsignieranforderung zu erstellen (CSR)

Im folgenden `create-certificate-from-csr` Beispiel wird ein Gerätezertifikat aus einem erstellten CSR erstellt. Sie können den `openssl` Befehl verwenden, um eine zu erstellen.

```
aws iot create-certificate-from-csr \  
  --certificate-signing-request=file://certificate.csr
```

Ausgabe:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/  
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificateId":  
  "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificatePem": "<certificate-text>"  
}
```

Weitere Informationen finden Sie [CreateCertificateFromCSR](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [CreateCertificateFromCsr](#) unter AWS CLI Befehlsreferenz.

create-custom-metric

Das folgende Codebeispiel zeigt die Verwendung `create-custom-metric`.

AWS CLI

Um eine benutzerdefinierte Metrik zu erstellen, die von Ihren Geräten auf Device Defender veröffentlicht wird

Im folgenden `create-custom-metric` Beispiel wird eine benutzerdefinierte Metrik erstellt, die den Akkustand in Prozent misst.

```
aws iot create-custom-metric \
  --metric-name "batteryPercentage" \
  --metric-type "number" \
  --display-name "Remaining battery percentage." \
  --region us-east-1 \
  --client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0"
```

Ausgabe:

```
{
  "metricName": "batteryPercentage",
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage"
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Metriken](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [CreateCustomMetric](#) in der AWS CLI Befehlsreferenz.

create-dimension

Das folgende Codebeispiel zeigt die Verwendung `create-dimension`.

AWS CLI

Um eine Dimension zu erstellen

Im Folgenden `create-dimension` wird eine Dimension mit einem einzelnen Themenfilter namens `TopicFilterForAuthMessages` erstellt.

```
aws iot create-dimension \  
  --name TopicFilterForAuthMessages \  
  --type TOPIC_FILTER \  
  --string-values device/+/auth
```

Ausgabe:

```
{  
  "name": "TopicFilterForAuthMessages",  
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"  
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateDimension](#) in der AWS CLI Befehlsreferenz.

create-domain-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-domain-configuration`.

AWS CLI

Um eine Domänenkonfiguration zu erstellen

Im folgenden `create-domain-configuration` Beispiel wird eine AWS-verwaltete Domänenkonfiguration mit dem Dienstyp `DATA` erstellt.

```
aws iot create-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --service-type "DATA"
```

Ausgabe:

```
{  
  "domainConfigurationName": "additionalDataDomain",  
  "domainConfigurationArn": "arn:aws:iot:us-west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
```

```
}
```

Weitere Informationen finden Sie unter [Configurable Endpoints](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateDomainConfiguration](#) in der AWS CLI Befehlsreferenz.

create-dynamic-thing-group

Das folgende Codebeispiel zeigt die Verwendung `create-dynamic-thing-group`.

AWS CLI

Um eine dynamische Dinggruppe zu erstellen

Im folgenden `create-dynamic-thing-group` Beispiel wird eine dynamische Dinggruppe erstellt, die alles enthält, dessen Temperaturattribut größer als 60 Grad ist. Sie müssen die AWS IoT-Flottenindizierung aktivieren, bevor Sie dynamische Dinggruppen verwenden können.

```
aws iot create-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm" \  
  --query-string "attributes.temperature>60"
```

Ausgabe:

```
{  
  "thingGroupName": "RoomTooWarm",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",  
  "thingGroupId": "9d52492a-fc87-43f4-b6e2-e571d2ffcad1",  
  "indexName": "AWS_Things",  
  "queryString": "attributes.temperature>60",  
  "queryVersion": "2017-09-30"  
}
```

Weitere Informationen finden Sie unter [Dynamische Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateDynamicThingGroup](#) in der AWS CLI Befehlsreferenz.

create-job

Das folgende Codebeispiel zeigt die Verwendung `create-job`.

AWS CLI

Beispiel 1: Um einen Job zu erstellen

Das folgende `create-job` Beispiel erstellt einen einfachen AWS IoT-Job, der ein JSON Dokument an das MyRaspberryPi Gerät sendet.

```
aws iot create-job \  
  --job-id "example-job-01" \  
  --targets "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi" \  
  --document file://example-job.json \  
  --description "example job test" \  
  --target-selection SNAPSHOT
```

Ausgabe:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",  
  "jobId": "example-job-01",  
  "description": "example job test"  
}
```

Beispiel 2: Um einen kontinuierlichen Job zu erstellen

Im folgenden `create-job` Beispiel wird ein Job erstellt, der weiter ausgeführt wird, nachdem die als Ziele angegebenen Dinge den Job abgeschlossen haben. In diesem Beispiel ist das Ziel eine Dinggruppe. Wenn der Gruppe also neue Geräte hinzugefügt werden, wird der kontinuierliche Job für diese neuen Dinge ausgeführt.

```
aws iot create-job --job-id „example-job-04“ --targets „arn:aws:iot:us-west- 2:123456789012:  
-job.json --description „Beispiel für einen kontinuierlichen Job“ --target-selection thinggroup/  
DeadBulbs“ --document file://example CONTINUOUS
```

Ausgabe:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateJob](#) in der AWS CLI Befehlsreferenz.

create-keys-and-certificate

Das folgende Codebeispiel zeigt die Verwendung `create-keys-and-certificate`.

AWS CLI

Um ein RSA key pair zu erstellen und ein X.509-Zertifikat auszustellen

Im Folgenden `create-keys-and-certificate` wird ein RSA 2048-Bit-Schlüsselpaar erstellt und ein X.509-Zertifikat unter Verwendung des ausgegebenen öffentlichen Schlüssels ausgestellt. Da dies das einzige Mal ist, dass AWS IoT den privaten Schlüssel für dieses Zertifikat bereitstellt, sollten Sie es an einem sicheren Ort aufbewahren.

```
aws iot create-keys-and-certificate \
  --certificate-pem-outfile "myTest.cert.pem" \
  --public-key-outfile "myTest.public.key" \
  --private-key-outfile "myTest.private.key"
```

Ausgabe:

```
{
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificateId":
    "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAstC01BTSEXAMPLE2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
BgqhkiG9w0BCQEWEG5vb251QGftYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAdgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAAEXAMPLEsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgqhkiG9w0BCQEXAMPLE251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvYsWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZEXAMPLELG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE"
```

```
Ibb30hjZnzcvcQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEEXAMPLEBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiEXAMPEQEFAA0CAQ8AMIIBCgKCAQEAEEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEuuN/
dMAS3fyce8DW/4+EXAMPLEYjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqX0gHh/
xJTtwo
+sGpWEXAMPLEDz18x0d2ka4tCzuWEXAMPLEeahJbYkCPUBSU8opVkr7qkEXAMPLE1DR6sx2Hocli00Ltu6Fkw91swQWEX
\nGB3ZPrNh0PzQYvjUSTzeccyNCx2EXAMPLEvp9mQ0UXP6p1fgxwKRX2fEXAMPLEDa
\nhJLXkX3rHU2xbxJSq7D+XEXAMPLEcw+LyFhI5mgFR188eGdsAEXAMPLE1nI9EesG\nFQIDAQAB\n-----
END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security
reasons\n-----END RSA PRIVATE KEY-----\n"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen und Registrieren eines AWS IoT-Gerätecertifikats im AWS IoT Developer Guide](#).

- API-Einheiten finden Sie [CreateKeysAndCertificate](#) in der AWS CLI Befehlsreferenz.

create-mitigation-action

Das folgende Codebeispiel zeigt die Verwendung `create-mitigation-action`.

AWS CLI

Um eine Minderungsmaßnahme zu erstellen

Im folgenden `create-mitigation-action` Beispiel wird eine Minderungsaktion mit dem Namen definiert, `AddThingsToQuarantineGroup1Action` die, wenn sie angewendet wird, Dinge in die angegebene Dinggruppe verschiebt. `QuarantineGroup1` Diese Aktion überschreibt dynamische Dinggruppen.

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

Inhalt von `params.json`:

```
{
```



```

    "actionName": "AddThingsToQuarantineGroup1Action",
    "actionParams": {
      "addThingsToThingGroupParams": {
        "thingGroupNames": [
          "QuarantineGroup1"
        ],
        "overrideDynamicGroups": true
      }
    },
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole"
  }
}

```

Ausgabe:

```

{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroup1Action",
  "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
}

```

Weitere Informationen finden Sie unter [CreateMitigationAction \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateMitigationAction](#) in der AWS CLI Befehlsreferenz.

create-ota-update

Das folgende Codebeispiel zeigt die Verwendung `create-ota-update`.

AWS CLI

Um ein OTA Update zur Verwendung mit Amazon Free zu erstellen RTOS

Das folgende `create-ota-update` Beispiel erstellt ein AWS IoT OTAUpdate für eine Zielgruppe von Dingen oder Gruppen. Dies ist Teil eines kostenlosen RTOS over-the-air Amazon-Updates, mit dem Sie neue Firmware-Images auf einem einzelnen Gerät oder einer Gruppe von Geräten bereitstellen können.

```

aws iot create-ota-update \
  --cli-input-json file://create-ota-update.json

```

Inhalt von create-ota-update.json:

```
{
  "otaUpdateId": "ota12345",
  "description": "A critical update needed right away.",
  "targets": [
    "device1",
    "device2",
    "device3",
    "device4"
  ],
  "targetSelection": "SNAPSHOT",
  "awsJobExecutionsRolloutConfig": {
    "maximumPerMinute": 10
  },
  "files": [
    {
      "fileName": "firmware.bin",
      "fileLocation": {
        "stream": {
          "streamId": "004",
          "fileId": 123
        }
      },
      "codeSigning": {
        "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
      }
    }
  ]
  "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
}
```

Ausgabe:

```
{
  "otaUpdateId": "ota12345",
  "awsIotJobId": "job54321",
  "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
  "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/itsajob",
  "otaUpdateStatus": "CREATE_IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [C reateOTAUpdate](#) in der AWS APIIoT-Referenz.

- API-Einheiten finden Sie [CreateOtaUpdate](#) unter AWS CLI Befehlsreferenz.

create-policy-version

Das folgende Codebeispiel zeigt die Verwendung `create-policy-version`.

AWS CLI

Um eine Richtlinie mit einer neuen Version zu aktualisieren

Im folgenden `create-policy-version` Beispiel wird eine Richtliniendefinition aktualisiert und eine neue Richtlinienversion erstellt. In diesem Beispiel wird die neue Version auch zur Standardversion.

```
aws iot create-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-document file://policy.json \  
  --set-as-default
```

Inhalt von `policy.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iot:UpdateCertificate",  
      "Resource": "*"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",  
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
  \"Allow\", \"Action\": \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",  
  "policyVersionId": "2",  
  "isDefaultVersion": true  
}
```

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreatePolicyVersion](#) in der AWS CLI Befehlsreferenz.

create-policy

Das folgende Codebeispiel zeigt die Verwendung `create-policy`.

AWS CLI

Um eine AWS IoT-Richtlinie zu erstellen

Im folgenden `create-policy` Beispiel wird eine AWS IoT-Richtlinie mit dem Namen `TemperatureSensorPolicy` erstellt. Die `policy.json` Datei enthält Anweisungen, die AWS IoT-Richtlinienaktionen zulassen.

```
aws iot create-policy \  
  --policy-name TemperatureSensorPolicy \  
  --policy-document file://policy.json
```

Inhalt von `policy.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Publish",  
        "iot:Receive"  
      ],  
      "Resource": [  
        "arn:aws:iot:us-west-2:123456789012:topic/topic_1",  
        "arn:aws:iot:us-west-2:123456789012:topic/topic_2"  
      ]  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Subscribe"  
      ],  
      "Resource": [  

```

```

        "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1",
        "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:Connect"
    ],
    "Resource": [
      "arn:aws:iot:us-west-2:123456789012:client/basicPubSub"
    ]
  }
]
}

```

Ausgabe:

```

{
  "policyName": "TemperatureSensorPolicy",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
  "policyDocument": "{
    \"Version\": \"2012-10-17\",
    \"Statement\": [
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Publish\",
          \"iot:Receive\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Subscribe\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1\",

```

```

        \ "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2\"
      ]
    },
    {
      \ "Effect\" : \ "Allow\" ,
      \ "Action\" : [
        \ "iot:Connect\"
      ],
      \ "Resource\" : [
        \ "arn:aws:iot:us-west-2:123456789012:client/basicPubSub\"
      ]
    }
  ]
},
"policyVersionId": "1"
}

```

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreatePolicy](#) in der AWS CLI Befehlsreferenz.

create-provisioning-claim

Das folgende Codebeispiel zeigt die Verwendung `create-provisioning-claim`.

AWS CLI

Um einen Bereitstellungsanspruch zu erstellen

Im folgenden `create-provisioning-claim` Beispiel wird ein Bereitstellungsanspruch aus einer Bereitstellungsvorlage erstellt.

```

aws iot create-provisioning-claim \
  --template-name MyTestProvisioningTemplate

```

Ausgabe:

```

{
  "certificateId":
    "78de02184b2ce80cf8fb709bda59e62b19fb83513590483eb0434589476ab09f",
  "certificatePem": "-----BEGIN CERTIFICATE-----\nMIIDdzCCA1
+gAwIBAgIUXSZhEBLztMLZ2fHG

```



```

NlimB7E9X8d3PcWYZVz/3ly4b9NPPRqdFlcPT24Sn68ZYiieaurC+Z
\n1VGLB6uoZUD2MLAE3jrwKd7tGp0c4E8i
9H402D8ip0lWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABAOIBAEAybn
QUtx9T2/nK\ntZT2pA4iugecxI4dz+DmT0XVXs5VJmrx/
nBSq6ejXExEpSIM04RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1kr0IPuxWebN0rRoPZr+1RSer+ww2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9Fd0UoQxdjHBHaM5PDmJ
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nnou0+91eTOP+/QfP7P8Zj0Ik02Xiv1RcVDyN/
E4QXPkuIkM/8vS8VK+
E9pATQ0MtB\n2lw8R/YU5AJd6j1EXAMPLEGU2UzRzInNWiLtkPPPqgqXXhx0f+mxByjcMa1VJk0L
\nh0G2R0UCgY
EA+R0cHNHy/XbsP7Fih0hEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5S1MBp7S1idxak
bIn28fKfPn62DaemGCIoyDgLf+eUxBx
\nngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAxuwT\npzdWwmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjsks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDxuxDe+V369Bianx8aZfyIsckA70wXW1w1h
\nngRC5rQ4X0gp3+Jmw7eA08LRYDjaN846+
Qbt02KcCgYAWS0UL51bijQR0ZwI0dz27\nnFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
BRtASW0Dab\naHXsDhVm2Jye+ESLohMaClOyCkT3118yqXIcEDStM07f01Ryag164EiJvSIrMfny\nnNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2x0NZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXgh0eoTpC8VXkzcnzouTiQ
\neFQQSGfnp7Ioe6UIz23715pKduszSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nnPRtIZQKBgA8MASxtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJcKAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nnvd1c5cFQewWb4/
zqAtPdinkPLN94ileI
79XQdc7R1J0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
    },
    "expiration": 1595955066.0
}

```

Weitere Informationen finden Sie unter [Bereitstellung durch einen vertrauenswürdigen Benutzer](#) im AWS IoT Core Developers Guide.

- API-Einheiten finden Sie [CreateProvisioningClaim](#) in der AWS CLI Befehlsreferenz.

create-provisioning-template-version

Das folgende Codebeispiel zeigt die Verwendung `create-provisioning-template-version`.

AWS CLI

Um eine Provisioning-Vorlagenversion zu erstellen

Im folgenden Beispiel wird eine Version für die angegebene Provisioningvorlage erstellt. Der Hauptteil der neuen Version ist in der Datei `template.json` enthalten.

```
aws iot create-provisioning-template-version \  
  --template-name widget-template \  
  --template-body file://template.json
```

Inhalt von `template.json`:

```
{  
  "Parameters" : {  
    "DeviceLocation": {  
      "Type": "String"  
    }  
  },  
  "Mappings": {  
    "LocationTable": {  
      "Seattle": {  
        "LocationUrl": "https://example.aws"  
      }  
    }  
  },  
  "Resources" : {  
    "thing" : {  
      "Type" : "AWS::IoT::Thing",  
      "Properties" : {  
        "AttributePayload" : {  
          "version" : "v1",  
          "serialNumber" : "serialNumber"  
        },  
        "ThingName" : {"Fn::Join":["",["ThingPrefix_",  
{"Ref":"SerialNumber"}]]},  
        "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",  
{"Ref":"SerialNumber"}]]},  
        "ThingGroups" : ["widgets", "WA"],
```

```

        "BillingGroup": "BillingGroup"
    },
    "OverrideSettings" : {
        "AttributePayload" : "MERGE",
        "ThingTypeName" : "REPLACE",
        "ThingGroups" : "DO_NOTHING"
    }
},
"certificate" : {
    "Type" : "AWS::IoT::Certificate",
    "Properties" : {
        "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
        "Status" : "Active"
    }
},
"policy" : {
    "Type" : "AWS::IoT::Policy",
    "Properties" : {
        "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement": [{
                "Effect": "Allow",
                "Action":["iot:Publish"],
                "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/foo/
bar"]
            }]
        }
    }
},
"DeviceConfiguration": {
    "FallbackUrl": "https://www.example.com/test-site",
    "LocationUrl": {
        "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
"LocationUrl"]}
    }
}
}

```

Ausgabe:

```
{
```

```
"templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-  
template",  
"templateName": "widget-template",  
"versionId": 2,  
"isDefaultVersion": false  
}
```

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [CreateProvisioningTemplateVersion](#) in der AWS CLI Befehlsreferenz.

create-provisioning-template

Das folgende Codebeispiel zeigt die Verwendung `create-provisioning-template`.

AWS CLI

Um eine Bereitstellungsvorlage zu erstellen

Im folgenden `create-provisioning-template` Beispiel wird eine in der Datei definierte Bereitstellungsvorlage erstellt. `template.json`

```
aws iot create-provisioning-template \  
  --template-name widget-template \  
  --description "A provisioning template for widgets" \  
  --provisioning-role-arn arn:aws:iam::123456789012:role/Provision_role \  
  --template-body file://template.json
```

Inhalt von `template.json`:

```
{  
  "Parameters" : {  
    "DeviceLocation": {  
      "Type": "String"  
    }  
  },  
  "Mappings": {  
    "LocationTable": {  
      "Seattle": {  
        "LocationUrl": "https://example.aws"  
      }  
    }  
  }  
}
```

```

    },
    "Resources" : {
      "thing" : {
        "Type" : "AWS::IoT::Thing",
        "Properties" : {
          "AttributePayload" : {
            "version" : "v1",
            "serialNumber" : "serialNumber"
          },
          "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
          "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
          "ThingGroups" : ["widgets", "WA"],
          "BillingGroup": "BillingGroup"
        },
        "OverrideSettings" : {
          "AttributePayload" : "MERGE",
          "ThingTypeName" : "REPLACE",
          "ThingGroups" : "DO_NOTHING"
        }
      },
      "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
          "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
          "Status" : "Active"
        }
      },
      "policy" : {
        "Type" : "AWS::IoT::Policy",
        "Properties" : {
          "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement": [{
              "Effect": "Allow",
              "Action":["iot:Publish"],
              "Resource": ["arn:aws:iot:us-east-1:504350838278:topic/foo/
bar"]
            }]
          }
        }
      }
    }
  },
}

```

```
"DeviceConfiguration": {
  "FallbackUrl": "https://www.example.com/test-site",
  "LocationUrl": {
    "Fn::FindInMap": ["LocationTable", {"Ref": "DeviceLocation"}],
    "LocationUrl": []
  }
}
```

Ausgabe:

```
{
  "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-template",
  "templateName": "widget-template",
  "defaultVersionId": 1
}
```

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [CreateProvisioningTemplate](#) in der AWS CLI Befehlsreferenz.

create-role-alias

Das folgende Codebeispiel zeigt die Verwendung `create-role-alias`.

AWS CLI

Um einen Rollenalias zu erstellen

Im folgenden `create-role-alias` Beispiel wird ein Rollenalias erstellt, der `LightBulbRole` für die angegebene Rolle aufgerufen wird.

```
aws iot create-role-alias \
  --role-alias LightBulbRole \
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Ausgabe:

```
{
```

```
"roleAlias": "LightBulbRole",
"roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

Weitere Informationen finden Sie [CreateRoleAlias](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [CreateRoleAlias](#) unter AWS CLI Befehlsreferenz.

create-scheduled-audit

Das folgende Codebeispiel zeigt die Verwendung `create-scheduled-audit`.

AWS CLI

Um ein geplantes Audit zu erstellen

Im folgenden `create-scheduled-audit` Beispiel wird ein geplantes Audit erstellt, das wöchentlich, am Mittwoch, ausgeführt wird, um zu überprüfen, ob CA-Zertifikate oder Gerätezertifikate ablaufen.

```
aws iot create-scheduled-audit \
  --scheduled-audit-name WednesdayCertCheck \
  --frequency WEEKLY \
  --day-of-week WED \
  --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK
```

Ausgabe:

```
{
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateScheduledAudit](#) in der AWS CLI Befehlsreferenz.

create-security-profile

Das folgende Codebeispiel zeigt die Verwendung `create-security-profile`.

AWS CLI

Um ein Sicherheitsprofil zu erstellen

Im folgenden `create-security-profile` Beispiel wird ein Sicherheitsprofil erstellt, das überprüft, ob die Mobilfunkbandbreite einen Schwellenwert überschreitet oder ob innerhalb von fünf Minuten mehr als 10 Autorisierungsfehler auftreten.

```
aws iot create-security-profile \
  --security-profile-name PossibleIssue \
  --security-profile-description "Check to see if authorization fails 10 times in 5 minutes or if cellular bandwidth exceeds 128" \
  --behaviors "[{\\"name\\":\\"CellularBandwidth\\",\\"metric\\":\\"aws:message-byte-size\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":128},\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}},{\\"name\\":\\"Authorization\\",\\"metric\\":\\"aws:num-authorization-failures\\",\\"criteria\\":{\\"comparisonOperator\\":\\"less-than\\",\\"value\\":{\\"count\\":10},\\"durationSeconds\\":300,\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}}]"
```

Ausgabe:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/PossibleIssue"
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateSecurityProfile](#) in der AWS CLI Befehlsreferenz.

`create-stream`

Das folgende Codebeispiel zeigt die Verwendung `create-stream`.

AWS CLI

Um einen Stream für die Bereitstellung einer oder mehrerer großer Dateien in Blöcken zu erstellen

MQTT

Im folgenden `create-stream` Beispiel wird ein Stream für die Bereitstellung einer oder mehrerer großer Dateien in Chunks Over erstellt. MQTT Ein Stream transportiert Datenbytes in Blöcken

oder Blöcken, die als MQTT Nachrichten aus einer Quelle wie S3 verpackt sind. Es können eine oder mehrere Dateien mit einem Stream verbunden sein.

```
aws iot create-stream \  
  --cli-input-json file://create-stream.json
```

Inhalt von `create-stream.json`:

```
{  
  "streamId": "stream12345",  
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
  "files": [  
    {  
      "fileId": 123,  
      "s3Location": {  
        "bucket": "codesign-ota-bucket",  
        "key": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"  
      }  
    }  
  ],  
  "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"  
}
```

Ausgabe:

```
{  
  "streamId": "stream12345",  
  "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",  
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
  "streamVersion": "1"  
}
```

Weitere Informationen finden Sie [CreateStream](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [CreateStream](#) unter AWS CLI Befehlsreferenz.

create-thing-group

Das folgende Codebeispiel zeigt die Verwendung `create-thing-group`.

AWS CLI

Beispiel 1: Um eine Dinggruppe zu erstellen

Im folgenden `create-thing-group` Beispiel wird eine Dinggruppe `LightBulbs` mit einer Beschreibung und zwei Attributen erstellt.

```
aws iot create-thing-group \  
  --thing-group-name LightBulbs \  
  --thing-group-properties "thingGroupDescription=\"Generic bulb group\",  
  attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

Ausgabe:

```
{  
  "thingGroupName": "LightBulbs",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",  
  "thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"  
}
```

Beispiel 2: Um eine Dinggruppe zu erstellen, die Teil einer übergeordneten Gruppe ist

Im Folgenden `create-thing-group` wird eine Dinggruppe mit `HalogenBulbs` dem Namen einer übergeordneten Dinggruppe erstellt `LightBulbs`.

```
aws iot create-thing-group \  
  --thing-group-name HalogenBulbs \  
  --parent-group-name LightBulbs
```

Ausgabe:

```
{  
  "thingGroupName": "HalogenBulbs",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",  
  "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"  
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateThingGroup](#) in der AWS CLI Befehlsreferenz.

create-thing-type

Das folgende Codebeispiel zeigt die Verwendung `create-thing-type`.

AWS CLI

Um einen Dingtyp zu definieren

Das folgende `create-thing-type` Beispiel definiert einen Dingtyp und die zugehörigen Attribute.

```
aws iot create-thing-type \  
  --thing-type-name "LightBulb" \  
  --thing-type-properties "thingTypeDescription=light bulb type,  
  searchableAttributes=wattage,model"
```

Ausgabe:

```
{  
  "thingTypeName": "LightBulb",  
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",  
  "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"  
}
```

Weitere Informationen finden Sie unter [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateThingType](#) in der AWS CLI Befehlsreferenz.

create-thing

Das folgende Codebeispiel zeigt die Verwendung `create-thing`.

AWS CLI

Beispiel 1: Um einen Ding-Datensatz in der Registrierung zu erstellen

Das folgende `create-thing` Beispiel erstellt einen Eintrag für ein Gerät in der AWS IoT-Dingregistrierung.

```
aws iot create-thing \  
  --thing-name "LightBulb"
```

```
--thing-name SampleIoTThing
```

Ausgabe:

```
{
  "thingName": "SampleIoTThing",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/SampleIoTThing",
  "thingId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "
}
```

Beispiel 2: Um ein Ding zu definieren, das einem Dingtyp zugeordnet ist

Im folgenden `create-thing` Beispiel wird ein Ding mit dem angegebenen Dingtyp und seinen Attributen erstellt.

```
aws iot create-thing \
  --thing-name "MyLightBulb" \
  --thing-type-name "LightBulb" \
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

Ausgabe:

```
{
  "thingName": "MyLightBulb",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
  "thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

Weitere Informationen finden Sie unter [How to Manage Things with the Registry](#) and [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateThing](#) in der AWS CLI Befehlsreferenz.

create-topic-rule-destination

Das folgende Codebeispiel zeigt die Verwendung `create-topic-rule-destination`.

AWS CLI

Um ein Ziel für eine Themenregel zu erstellen

Im folgenden `create-topic-rule-destination` Beispiel wird ein Ziel für Themenregeln für einen HTTP Endpunkt erstellt.

```
aws iot create-topic-rule-destination \  
  --destination-configuration httpUrlConfiguration={confirmationUrl=https://  
example.com}
```

Ausgabe:

```
{  
  "topicRuleDestination": {  
    "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "status": "IN_PROGRESS",  
    "statusReason": "Awaiting confirmation. Confirmation message sent on  
2020-07-09T22:47:54.154Z; no response received from the endpoint.",  
    "httpUrlProperties": {  
      "confirmationUrl": "https://example.com"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Ziels für Themenregeln](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [CreateTopicRuleDestination](#) in der AWS CLI Befehlsreferenz.

create-topic-rule

Das folgende Codebeispiel zeigt die Verwendung `create-topic-rule`.

AWS CLI

Um eine Regel zu erstellen, die eine SNS Amazon-Warnung sendet

Im folgenden `create-topic-rule` Beispiel wird eine Regel erstellt, die eine SNS Amazon-Nachricht sendet, wenn die Bodenfeuchte, wie sie in einem Geräteschatten gefunden wurde, niedrig ist.

```
aws iot create-topic-rule \  
  --rule-name "LowMoistureRule" \  
  --topic-rule-destination-arn arn:aws:iot:us-west-2:123456789012:ruledestination/http/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

```
--topic-rule-payload file://plant-rule.json
```

Für das Beispiel muss der folgende JSON Code in einer Datei mit dem Namen gespeichert werden `plant-rule.json`:

```
{
  "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
state.reported.moisture = 'low'\n",
  "description": "Sends an alert whenever soil moisture level readings are too
low.",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
    "sns": {
      "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
      "messageFormat": "RAW"
    }
  ]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen einer AWS IoT-Regel](#) im AWS IoT-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateTopicRule](#) in der AWS CLI Befehlsreferenz.

delete-account-audit-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-account-audit-configuration`.

AWS CLI

Um alle Audit-Checks für Ihr AWS Konto zu deaktivieren

Im folgenden `delete-account-audit-configuration` Beispiel werden die Standardeinstellungen für AWS IoT Device Defender für dieses Konto wiederhergestellt, wobei alle Auditprüfungen deaktiviert und die Konfigurationsdaten gelöscht werden. Außerdem werden alle geplanten Audits für dieses Konto gelöscht. Verwenden Sie diesen Befehl mit Vorsicht.

```
aws iot delete-account-audit-configuration \
  --delete-scheduled-audits
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteAccountAuditConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-audit-suppression

Das folgende Codebeispiel zeigt die Verwendung `delete-audit-suppression`.

AWS CLI

Um ein Audit zu löschen, bei dem eine Unterdrückung festgestellt wurde

Im folgenden `delete-audit-suppression` Beispiel wird die Unterdrückung eines Prüfungsergebnisses für `DEVICE __ CERTIFICATE EXPIRING _ CHECK` gelöscht.

```
aws iot delete-audit-suppression \
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
  --resource-identifier deviceCertificateId="c7691e<shortened>"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit finding suppressions](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteAuditSuppression](#) in der AWS CLI Befehlsreferenz.

delete-authorizer

Das folgende Codebeispiel zeigt die Verwendung `delete-authorizer`.

AWS CLI

Um einen benutzerdefinierten Autorisierer zu löschen

Im folgenden `delete-authorizer` Beispiel wird der angegebene Autorisierer gelöscht.

`CustomAuthorizer` Ein benutzerdefinierter Autorisierer muss sich in diesem `INACTIVE` Status befinden, bevor Sie ihn löschen können.

```
aws iot delete-authorizer \  
  --authorizer-name CustomAuthorizer
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteAuthorizer](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteAuthorizer](#) in der AWS CLI Befehlsreferenz.

delete-billing-group

Das folgende Codebeispiel zeigt die Verwendung `delete-billing-group`.

AWS CLI

Um eine Abrechnungsgruppe zu löschen

Im folgenden `delete-billing-group` Beispiel wird die angegebene Abrechnungsgruppe gelöscht. Sie können eine Abrechnungsgruppe löschen, auch wenn sie ein oder mehrere Dinge enthält.

```
aws iot delete-billing-group \  
  --billing-group-name BillingGroupTwo
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteBillingGroup](#) in der AWS CLI Befehlsreferenz.

delete-ca-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-ca-certificate`.

AWS CLI

Um ein CA-Zertifikat zu löschen

Im folgenden `delete-ca-certificate` Beispiel wird das CA-Zertifikat mit der angegebenen Zertifikat-ID gelöscht.

```
aws iot delete-ca-certificate \  
  --certificate-id
```

```
--certificate-  
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [DeleteCACertificate](#) in der AWS APIIoT-Referenz.

- APIEinzelheiten finden Sie [DeleteCaCertificate](#) unter AWS CLI Befehlsreferenz.

delete-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-certificate`.

AWS CLI

Um ein Gerätezertifikat zu löschen

Im folgenden `delete-certificate` Beispiel wird das Gerätezertifikat mit der angegebenen ID gelöscht.

```
aws iot delete-certificate \  
  --certificate-  
  id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteCertificate](#) in der AWS APIIoT-Referenz.

- APIEinzelheiten finden Sie [DeleteCertificate](#) unter AWS CLI Befehlsreferenz.

delete-custom-metric

Das folgende Codebeispiel zeigt die Verwendung `delete-custom-metric`.

AWS CLI

Um eine benutzerdefinierte Metrik zu löschen

Im folgenden `delete-custom-metric` Beispiel wird eine benutzerdefinierte Metrik gelöscht.

```
aws iot delete-custom-metric \  
  --metric-name batteryPercentage \  
  --metric-id metricId
```



```
--region us-east-1
```

Ausgabe:

```
HTTP 200
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Metriken](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [DeleteCustomMetric](#) in der AWS CLI Befehlsreferenz.

delete-dimension

Das folgende Codebeispiel zeigt die Verwendung `delete-dimension`.

AWS CLI

Um eine Dimension zu löschen

Im folgenden `delete-dimension` Beispiel wird eine Dimension namens `TopicFilterForAuthMessages` gelöscht.

```
aws iot delete-dimension \  
  --name TopicFilterForAuthMessages
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteDimension](#) in der AWS CLI Befehlsreferenz.

delete-domain-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-domain-configuration`.

AWS CLI

Um eine Domänenkonfiguration zu löschen

Im folgenden `delete-domain-configuration` Beispiel wird eine `additionalDataDomain` aus Ihrem AWS Konto benannte Domänenkonfiguration gelöscht.

```
aws iot delete-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --domain-configuration-status "OK"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Configurable Endpoints](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteDomainConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-dynamic-thing-group

Das folgende Codebeispiel zeigt die Verwendung `delete-dynamic-thing-group`.

AWS CLI

Um eine dynamische Dinggruppe zu löschen

Im folgenden `delete-dynamic-thing-group` Beispiel wird die angegebene dynamische Dinggruppe gelöscht.

```
aws iot delete-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dynamische Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteDynamicThingGroup](#) in der AWS CLI Befehlsreferenz.

delete-job-execution

Das folgende Codebeispiel zeigt die Verwendung `delete-job-execution`.

AWS CLI

Um eine Jobausführung zu löschen

Im folgenden `delete-job-execution` Beispiel wird die Auftragsausführung des angegebenen Jobs auf einem Gerät gelöscht. Wird verwendet `describe-job-execution`, um die Ausführungsnummer abzurufen.

```
aws iot delete-job-execution
  --job-id "example-job-02"
  --thing-name "MyRaspberryPi"
  --execution-number 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteJobExecution](#) in der AWS CLI Befehlsreferenz.

delete-job

Das folgende Codebeispiel zeigt die Verwendung `delete-job`.

AWS CLI

Einen Auftrag löschen

Im folgenden `delete-job` Beispiel wird der angegebene Job gelöscht. Durch Angabe der `--force` Option wird der Job gelöscht, auch wenn der Status lautet `IN_PROGRESS`.

```
aws iot delete-job \
  --job-id "example-job-04" \
  --force
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteJob](#) in der AWS CLI Befehlsreferenz.

delete-mitigation-action

Das folgende Codebeispiel zeigt die Verwendung `delete-mitigation-action`.

AWS CLI

Um eine Minderungsaktion zu löschen

Im folgenden `delete-mitigation-action` Beispiel wird die angegebene Abhilfemaßnahme gelöscht.

```
aws iot delete-mitigation-action \  
  --action-name AddThingsToQuarantineGroup1Action
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [DeleteMitigationAction \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteMitigationAction](#) in der AWS CLI Befehlsreferenz.

delete-ota-update

Das folgende Codebeispiel zeigt die Verwendung `delete-ota-update`.

AWS CLI

Um ein OTA Update zu löschen

Im folgenden `delete-ota-update` Beispiel wird das angegebene OTA Update gelöscht.

```
aws iot delete-ota-update \  
  --ota-update-id ota12345 \  
  --delete-stream \  
  --force-delete-aws-job
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [DeleteOTAUpdate](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DeleteOtaUpdate](#) unter AWS CLI Befehlsreferenz.

delete-policy-version

Das folgende Codebeispiel zeigt die Verwendung `delete-policy-version`.

AWS CLI

Um eine Version der Richtlinie zu löschen

Im folgenden `delete-policy-version` Beispiel wird Version 2 der angegebenen Richtlinie aus Ihrem AWS Konto gelöscht.

```
aws iot delete-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeletePolicyVersion](#) in der AWS CLI Befehlsreferenz.

delete-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-policy`.

AWS CLI

Um eine Richtlinie zu löschen

Im folgenden `delete-policy` Beispiel wird die angegebene Richtlinie aus Ihrem AWS Konto gelöscht.

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeletePolicy](#) in der AWS CLI Befehlsreferenz.

delete-provisioning-template-version

Das folgende Codebeispiel zeigt die Verwendung `delete-provisioning-template-version`.

AWS CLI

Um eine Provisioning-Vorlagenversion zu löschen

Im folgenden `delete-provisioning-template-version` Beispiel wird Version 2 der angegebenen Provisioning-Vorlage gelöscht.

```
aws iot delete-provisioning-template-version \  
  --version-id 2 \  
  --template-name "widget-template"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [DeleteProvisioningTemplateVersion](#) in der AWS CLI Befehlsreferenz.

delete-provisioning-template

Das folgende Codebeispiel zeigt die Verwendung `delete-provisioning-template`.

AWS CLI

Um eine Bereitstellungsvorlage zu löschen

Im folgenden `delete-provisioning-template` Beispiel wird die angegebene Bereitstellungsvorlage gelöscht.

```
aws iot delete-provisioning-template \  
  --template-name widget-template
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [DeleteProvisioningTemplate](#) in der AWS CLI Befehlsreferenz.

delete-registration-code

Das folgende Codebeispiel zeigt die Verwendung `delete-registration-code`.

AWS CLI

Um Ihren Registrierungscode zu löschen

Im folgenden `delete-registration-code` Beispiel wird ein für das AWS IoT-Konto spezifischer Registrierungscode gelöscht.

```
aws iot delete-registration-code
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden Sie Ihr eigenes Zertifikat](#) im AWS IoT-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteRegistrationCode](#) in der AWS CLI Befehlsreferenz.

delete-role-alias

Das folgende Codebeispiel zeigt die Verwendung `delete-role-alias`.

AWS CLI

So löschen Sie einen AWS IoT-Rollenalias

Im folgenden `delete-role-alias` Beispiel wird ein AWS IoT-Rollenalias mit dem Namen `LightBulbRole` gelöscht.

```
aws iot delete-role-alias \  
  --role-alias LightBulbRole
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Autorizing Direct Calls to AWS Services](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteRoleAlias](#) in der AWS CLI Befehlsreferenz.

delete-scheduled-audit

Das folgende Codebeispiel zeigt die Verwendung `delete-scheduled-audit`.

AWS CLI

Um ein geplantes Audit zu löschen

Im folgenden `delete-scheduled-audit` Beispiel wird das geplante AWS IoT Device Defender Defender-Audit mit dem Namen `AWSIoTDeviceDefenderDailyAudit` gelöscht.

```
aws iot delete-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteScheduledAudit](#) in der AWS CLI Befehlsreferenz.

delete-security-profile

Das folgende Codebeispiel zeigt die Verwendung `delete-security-profile`.

AWS CLI

Um ein Sicherheitsprofil zu löschen

Im folgenden `delete-security-profile` Beispiel wird ein Sicherheitsprofil mit dem Namen `PossibleIssue` gelöscht.

```
aws iot delete-security-profile \  
  --security-profile-name PossibleIssue
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteSecurityProfile](#) in der AWS CLI Befehlsreferenz.

delete-stream

Das folgende Codebeispiel zeigt die Verwendung `delete-stream`.

AWS CLI

Um einen Stream zu löschen

Im folgenden `delete-stream` Beispiel wird der angegebene Stream gelöscht.

```
aws iot delete-stream \  
  --stream-name
```



```
--stream-id stream12345
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteStream](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DeleteStream](#) unter AWS CLI Befehlsreferenz.

delete-thing-group

Das folgende Codebeispiel zeigt die Verwendung `delete-thing-group`.

AWS CLI

Um eine Dinggruppe zu löschen

Im folgenden `delete-thing-group` Beispiel wird die angegebene Dinggruppe gelöscht. Sie können eine Dinggruppe nicht löschen, wenn sie untergeordnete Dinggruppen enthält.

```
aws iot delete-thing-group \  
  --thing-group-name DefectiveBulbs
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteThingGroup](#) in der AWS CLI Befehlsreferenz.

delete-thing-type

Das folgende Codebeispiel zeigt die Verwendung `delete-thing-type`.

AWS CLI

Beispiel 1: Um einen Dingtyp zu löschen

Im folgenden `delete-thing-type` Beispiel wird ein veralteter Dingtyp gelöscht.

```
aws iot delete-thing-type \  
  --thing-type-name "obsoleteThingType"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteThingType](#) in der AWS CLI Befehlsreferenz.

delete-thing

Das folgende Codebeispiel zeigt die Verwendung `delete-thing`.

AWS CLI

Um detaillierte Informationen zu einer Sache anzuzeigen

Das folgende `delete-thing` Beispiel löscht eine Sache aus der AWS IoT-Registrierung für Ihr AWS Konto.

```
aws iot delete-thing --thing-name "FourthBulb"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [How to Manage Things with the Registry](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteThing](#) in der AWS CLI Befehlsreferenz.

delete-topic-rule-destination

Das folgende Codebeispiel zeigt die Verwendung `delete-topic-rule-destination`.

AWS CLI

Um ein Ziel für eine Themenregel zu löschen

Im folgenden `delete-topic-rule-destination` Beispiel wird das angegebene Ziel für Themenregeln gelöscht.

```
aws iot delete-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-1111EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Ziels für Themenregeln](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DeleteTopicRuleDestination](#) in der AWS CLI Befehlsreferenz.

delete-topic-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-topic-rule`.

AWS CLI

So löschen Sie eine Regel

Im folgenden `delete-topic-rule` Beispiel wird die angegebene Regel gelöscht.

```
aws iot delete-topic-rule \  
  --rule-name "LowMoistureRule"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Regel](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteTopicRule](#) in der AWS CLI Befehlsreferenz.

delete-v2-logging-level

Das folgende Codebeispiel zeigt die Verwendung `delete-v2-logging-level`.

AWS CLI

Um die Protokollierungsebene für eine Dinggruppe zu löschen

Im folgenden `delete-v2-logging-level` Beispiel wird die Protokollierungsebene für die angegebene Dinggruppe gelöscht.

```
aws iot delete-v2-logging-level \  
  --target-type THING_GROUP \  
  --target-name LightBulbs
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteV2 LoggingLevel](#) in der Befehlsreferenz.AWS CLI

deprecate-thing-type

Das folgende Codebeispiel zeigt die Verwendung. `deprecate-thing-type`

AWS CLI

Beispiel 1: Um einen Dingtyp als veraltet zu kennzeichnen

Im folgenden `deprecate-thing-type` Beispiel wird ein Dingtyp als veraltet eingestuft, sodass Benutzer ihm keine neuen Dinge zuordnen können.

```
aws iot deprecate-thing-type \  
  --thing-type-name "obsoleteThingType"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die Ablehnung eines Dingtyps rückgängig zu machen

Im folgenden `deprecate-thing-type` Beispiel wird die Ablehnung eines Dingtyps rückgängig gemacht, sodass Benutzer ihm wieder neue Dinge zuordnen können.

```
aws iot deprecate-thing-type \  
  --thing-type-name "obsoleteThingType" \  
  --undo-deprecate
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeprecateThingType](#) in der AWS CLI Befehlsreferenz.

describe-account-audit-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-account-audit-configuration`.

AWS CLI

Um die aktuellen Audit-Konfigurationseinstellungen anzuzeigen

Das folgende `describe-account-audit-configuration` Beispiel listet die aktuellen Einstellungen für Ihre AWS IoT Device Defender Defender-Auditkonfiguration auf.

```
aws iot describe-account-audit-configuration
```

Ausgabe:

```
{
  "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTDeviceDefenderAudit_1551201085996",
  "auditNotificationTargetConfigurations": {
    "SNS": {
      "targetArn": "arn:aws:sns:us-west-2:123456789012:ddaudits",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/
AWSIoTDeviceDefenderAudit",
      "enabled": true
    }
  },
  "auditCheckConfigurations": {
    "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
      "enabled": true
    },
    "CA_CERTIFICATE_EXPIRING_CHECK": {
      "enabled": true
    },
    "CONFLICTING_CLIENT_IDS_CHECK": {
      "enabled": true
    },
    "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
      "enabled": true
    },
    "DEVICE_CERTIFICATE_SHARED_CHECK": {
      "enabled": true
    },
    "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
      "enabled": true
    },
    "LOGGING_DISABLED_CHECK": {
      "enabled": true
    },
    "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
      "enabled": true
    },
  },
}
```

```

    "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
      "enabled": true
    },
    "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
      "enabled": true
    }
  }
}

```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeAccountAuditConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-audit-finding

Das folgende Codebeispiel zeigt die Verwendung `describe-audit-finding`.

AWS CLI

Um Details zu einem Prüfungsergebnis aufzulisten

Das folgende `describe-audit-finding` Beispiel listet die Details für das angegebene AWS IoT Device Defender Defender-Audit-Ergebnis auf. Ein Audit kann zu mehreren Ergebnissen führen. Verwenden Sie den `list-audit-findings` Befehl, um eine Liste der Ergebnisse eines Audits abzurufen `findingId`.

```

aws iot describe-audit-finding \
  --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"

```

Ausgabe:

```

{
  "finding": {
    "findingId": "ef4826b8-e55a-44b9-b460-5c485355371b",
    "taskId": "873ed69c74a9ec8fa9b8e88e9abc4661",
    "checkName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
    "taskStartTime": 1576012045.745,
    "findingTime": 1576012046.168,
    "severity": "CRITICAL",
    "nonCompliantResource": {
      "resourceType": "IOT_POLICY",
      "resourceIdentifier": {

```

```

        "policyVersionIdentifier": {
            "policyName": "smp-ggrass-group_Core-policy",
            "policyVersionId": "1"
        }
    },
    "reasonForNonCompliance": "Policy allows broad access to IoT data plane
actions: [iot:Subscribe, iot:Connect, iot:GetThingShadow, iot>DeleteThingShadow,
iot:UpdateThingShadow, iot:Publish].",
    "reasonForNonComplianceCode":
"ALLAWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS"
}
}

```

Weitere Informationen finden [Sie unter Prüfergebnisse überprüfen \(Auditbefehle\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeAuditFinding](#) in der AWS CLI Befehlsreferenz.

describe-audit-mitigation-actions-task

Das folgende Codebeispiel zeigt die Verwendung `describe-audit-mitigation-actions-task`.

AWS CLI

Um die Details einer Aufgabe zur Prüfung von Minderungsmaßnahmen anzuzeigen

Das folgende `describe-audit-mitigation-actions-task` Beispiel zeigt die Details für die angegebene Aufgabe, bei der die auf ein Ergebnis angewendete `ResetPolicyVersionAction` wurde. Zu den Ergebnissen gehören, wann die Aufgabe gestartet und beendet wurde, wie viele Ergebnisse angestrebt wurden (und welches Ergebnis), und die Definition der Aktion, die im Rahmen dieser Aufgabe angewendet wird.

```

aws iot describe-audit-mitigation-actions-task \
  --task-id ResetPolicyTask01

```

Ausgabe:

```

{
  "taskStatus": "COMPLETED",
  "startTime": "2019-12-10T15:13:19.457000-08:00",

```

```

"endTime": "2019-12-10T15:13:19.947000-08:00",
"taskStatistics": {
  "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
    "totalFindingsCount": 1,
    "failedFindingsCount": 0,
    "succeededFindingsCount": 1,
    "skippedFindingsCount": 0,
    "canceledFindingsCount": 0
  }
},
"target": {
  "findingIds": [
    "ef4826b8-e55a-44b9-b460-5c485355371b"
  ]
},
"auditCheckToActionsMapping": {
  "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": [
    "ResetPolicyVersionAction"
  ]
},
"actionsDefinition": [
  {
    "name": "ResetPolicyVersionAction",
    "id": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
ReplacePolicyVersionRole",
    "actionParams": {
      "replaceDefaultPolicyVersionParams": {
        "templateName": "BLANK_POLICY"
      }
    }
  }
]
}

```

Weitere Informationen finden Sie unter [DescribeAuditMitigationActionsTask \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeAuditMitigationActionsTask](#) in der AWS CLI Befehlsreferenz.

describe-audit-suppression

Das folgende Codebeispiel zeigt die Verwendung `describe-audit-suppression`.

AWS CLI

Um Einzelheiten zu einem Audit zu erhalten, bei dem eine Unterdrückung festgestellt wurde

Im folgenden `describe-audit-suppression` Beispiel werden Details zur Unterdrückung eines Prüfungsergebnisses aufgeführt.

```
aws iot describe-audit-task \  
  --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

Ausgabe:

```
{  
  "taskStatus": "COMPLETED",  
  "taskType": "SCHEDULED_AUDIT_TASK",  
  "taskStartTime": 1596168096.157,  
  "taskStatistics": {  
    "totalChecks": 1,  
    "inProgressChecks": 0,  
    "waitingForDataCollectionChecks": 0,  
    "compliantChecks": 0,  
    "nonCompliantChecks": 1,  
    "failedChecks": 0,  
    "canceledChecks": 0  
  },  
  "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",  
  "auditDetails": {  
    "DEVICE_CERTIFICATE_EXPIRING_CHECK": {  
      "checkRunStatus": "COMPLETED_NON_COMPLIANT",  
      "checkCompliant": false,  
      "totalResourcesCount": 195,  
      "nonCompliantResourcesCount": 2  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Audit finding suppressions](#) im AWS IoT Developers Guide.

- API Einzelheiten finden Sie [DescribeAuditSuppression](#) in der AWS CLI Befehlsreferenz.

describe-audit-task

Das folgende Codebeispiel zeigt die Verwendung `describe-audit-task`.

AWS CLI

Um Informationen über eine Audit-Instanz abzurufen

Im folgenden `describe-audit-task` Beispiel werden Informationen zu einer Instanz eines AWS IoT Device Defender Defender-Audits abgerufen. Wenn das Audit abgeschlossen ist, sind zusammenfassende Statistiken für den Lauf in den Ergebnissen enthalten.

```
aws iot describe-audit-task \  
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Ausgabe:

```
{  
  "taskStatus": "COMPLETED",  
  "taskType": "ON_DEMAND_AUDIT_TASK",  
  "taskStartTime": 1560356923.434,  
  "taskStatistics": {  
    "totalChecks": 3,  
    "inProgressChecks": 0,  
    "waitingForDataCollectionChecks": 0,  
    "compliantChecks": 3,  
    "nonCompliantChecks": 0,  
    "failedChecks": 0,  
    "canceledChecks": 0  
  },  
  "auditDetails": {  
    "CA_CERTIFICATE_EXPIRING_CHECK": {  
      "checkRunStatus": "COMPLETED_COMPLIANT",  
      "checkCompliant": true,  
      "totalResourcesCount": 0,  
      "nonCompliantResourcesCount": 0  
    },  
    "DEVICE_CERTIFICATE_EXPIRING_CHECK": {  
      "checkRunStatus": "COMPLETED_COMPLIANT",  
      "checkCompliant": true,  
      "totalResourcesCount": 6,  
      "nonCompliantResourcesCount": 0  
    },  
    "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {  
      "checkRunStatus": "COMPLETED_COMPLIANT",  
      "checkCompliant": true,  
      "totalResourcesCount": 0,  
    }  
  }  
}
```

```

        "nonCompliantResourcesCount": 0
    }
}
}

```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeAuditTask](#) in der AWS CLI Befehlsreferenz.

describe-authorizer

Das folgende Codebeispiel zeigt die Verwendung `describe-authorizer`.

AWS CLI

Um Informationen über einen benutzerdefinierten Autorisierer zu erhalten

Im folgenden `describe-authorizer` Beispiel werden Details für den angegebenen benutzerdefinierten Autorisierer angezeigt.

```

aws iot describe-authorizer \
  --authorizer-name CustomAuthorizer

```

Ausgabe:

```

{
  "authorizerDescription": {
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer",
    "authorizerFunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:CustomAuthorizerFunction",
    "tokenKeyName": "MyAuthToken",
    "tokenSigningPublicKeys": {
      "FIRST_KEY": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAQ8AMIIBCgKCAQEAE1uJ0B4lQPgG/1M6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAgrCUgRFyggjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzw0BKPeic0asNJpqT8PkBbRaKYLEJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfP0HDJQgID0XnZwAlNnZc0hCwIx\n50g2LW20y9R/
dmqtDmJiVP97Z4GykxPvwlyHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
KEY-----"
    }
  }
}

```

```
    },
    "status": "ACTIVE",
    "creationDate": 1571245658.069,
    "lastModifiedDate": 1571245658.069
  }
}
```

Weitere Informationen finden Sie [DescribeAuthorizer](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DescribeAuthorizer](#) unter AWS CLI Befehlsreferenz.

describe-billing-group

Das folgende Codebeispiel zeigt die Verwendung `describe-billing-group`.

AWS CLI

Um Informationen über eine Abrechnungsgruppe zu erhalten

Im folgenden `describe-billing-group` Beispiel werden Informationen für die angegebene Abrechnungsgruppe abgerufen.

```
aws iot describe-billing-group --billing-group-name GroupOne
```

Ausgabe:

```
{
  "billingGroupName": "GroupOne",
  "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562",
  "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
  "version": 1,
  "billingGroupProperties": {},
  "billingGroupMetadata": {
    "creationDate": 1560199355.378
  }
}
```

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DescribeBillingGroup](#) in der AWS CLI Befehlsreferenz.

describe-ca-certificate

Das folgende Codebeispiel zeigt die Verwendung `describe-ca-certificate`.

AWS CLI

Um Details zu einem CA-Zertifikat zu erhalten

Im folgenden `describe-ca-certificate` Beispiel werden die Details für das angegebene CA-Zertifikat angezeigt.

```
aws iot describe-ca-certificate \
  --certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Ausgabe:

```
{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "status": "INACTIVE",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPWX18taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTA1VMTMQ8wDQYDVQQKDAZBbWF6b24wHhcNMTEwMjEzMTUwMTI0MjEzMTUwMTI0MjEz
\nMjEzMTUwMTI0MjEzMTUwMTI0MjEzMTUwMTI0MjEzMTUwMTI0MjEzMTUwMTI0MjEzMTUw
\nhkiG9w0BAQEFAAQCAQ8AMIIBCgKCAQEAZd3R3ioa1CS0MhFWfBrVGR036EK07UAF
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18Vz1FtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMaw0jhF4Lscu1KG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWwvUskzxbn01F2WCshbEJvqY8fIWtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJw0QxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+n04jd5cqaoEPtzoEPUEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJslxwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3Ijg0nd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcB1CFwH1yakwjJL1dR4yarnrfxgI
\nEqP4A0YVimAVoQ5FBwnloHe16+3qtDib1U9DeXBUctS55EcfREXAMPLEYtXdqU5C
\nU9ia4KAjV0dxW1+EFYmWx5eGeb0gDTNHBy1V6B/f0SZiQAwDYp4x3B+gAP+a/bWB
\nu1um0qtBdWe6L6/83L+JhaTByqV25iVJ4c/UZUnG8926wU1DM9zQvEXuEVvzZ7+m\n4PSNqst/
nV0vnLpoG4e0WgcJgANuB33CSwtjWSuYsbhmQQRknGhREXAMPLEZT4fm\nfo0e\n-----END
CERTIFICATE-----\n",
    "ownedBy": "123456789012",
    "creationDate": 1569365372.053,
```

```

    "autoRegistrationStatus": "DISABLE",
    "lastModifiedDate": 1569365372.053,
    "customerVersion": 1,
    "generationId": "c5c2eb95-140b-4f49-9393-6aaac85b2a90",
    "validity": {
      "notBefore": 1569360675.0,
      "notAfter": 1884720675.0
    }
  }
}

```

Weitere Informationen finden Sie unter [DescribeCACertificate](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DescribeCaCertificate](#) unter AWS CLI Befehlsreferenz.

describe-certificate

Das folgende Codebeispiel zeigt die Verwendung `describe-certificate`.

AWS CLI

Um Informationen über ein Zertifikat zu erhalten

Im folgenden `describe-certificate` Beispiel werden die Details für das angegebene Zertifikat angezeigt.

```

aws iot describe-certificate \
  --certificate-
  id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"

```

Ausgabe:

```

{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
    "certificateId":
    "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
    "status": "ACTIVE",
    "certificatePem": "-----BEGIN CERTIFICATE-----
MIICiTEXAMPLEQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQHEwdTZWF0dGx1MQ8wDQYDVQKKEwZBbWF6

```

```

b24xFDASBgNVBAwTC01BTSBDEXAMPLE1MRIwEAYDVQQDEw1UZjN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNEXAMPLEdBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BEXAMPLEz
b2xEXAMPLEYDVQQDEw1UZjN0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8EXAMPLEZIHvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLYEXAMPLEEpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7EXAMPLEGBzZswY6786m86gpE
Ibb30hjZncvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4
nUhVvxYUnEXAMPLE8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLE10ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
    "ownedBy": "123456789012",
    "creationDate": 1541022751.983,
    "lastModifiedDate": 1541022751.983,
    "customerVersion": 1,
    "transferData": {},
    "generationId": "6974fbcd-2e61-4114-bc5e-4204cc79b045",
    "validity": {
      "notBefore": 1541022631.0,
      "notAfter": 2524607999.0
    }
  }
}
}

```

Weitere Informationen finden Sie [DescribeCertificate](#) in der AWS API IoT-Referenz.

- API Einzelheiten finden Sie [DescribeCertificate](#) unter AWS CLI Befehlsreferenz.

describe-custom-metric

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-metric`.

AWS CLI

Um Informationen über eine benutzerdefinierte Device Defender-Metrik abzurufen

Im folgenden `describe-custom-metric` Beispiel werden Informationen zu einer benutzerdefinierten Metrik mit dem Namen `myCustomMetric` abgerufen.

```

aws iot describe-custom-metric \
  --metric-name myCustomMetric

```

Ausgabe:

```
{
  "metricName": "myCustomMetric",
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/myCustomMetric",
  "metricType": "number",
  "displayName": "My custom metric",
  "creationDate": 2020-11-17T23:02:12.879000-09:00,
  "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Metriken](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [DescribeCustomMetric](#) in der AWS CLI Befehlsreferenz.

describe-default-authorizer

Das folgende Codebeispiel zeigt die Verwendung `describe-default-authorizer`.

AWS CLI

Um Informationen über den standardmäßigen benutzerdefinierten Autorisierer zu erhalten

Im folgenden `describe-default-authorizer` Beispiel werden Details für den benutzerdefinierten Standardautorisierer angezeigt.

```
aws iot describe-default-authorizer
```

Ausgabe:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

Weitere Informationen finden Sie [DescribeDefaultAuthorizer](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DescribeDefaultAuthorizer](#) unter AWS CLI Befehlsreferenz.

describe-dimension

Das folgende Codebeispiel zeigt die Verwendung `describe-dimension`.

AWS CLI

Um Informationen über eine Dimension zu erhalten

Im folgenden `describe-dimension` Beispiel werden Informationen zu einer Dimension mit dem Namen `TopicFilterForAuthMessages` abgerufen.

```
aws iot describe-dimension \
  --name TopicFilterForAuthMessages
```

Ausgabe:

```
{
  "name": "TopicFilterForAuthMessages",
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
  "type": "TOPIC_FILTER",
  "stringValues": [
    "device/+/auth"
  ],
  "creationDate": 1578620223.255,
  "lastModifiedDate": 1578620223.255
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeDimension](#) in der AWS CLI Befehlsreferenz.

describe-domain-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-domain-configuration`.

AWS CLI

Um eine Domänenkonfiguration zu beschreiben

Im folgenden `describe-domain-configuration` Beispiel werden Details zur angegebenen Domänenkonfiguration angezeigt.

```
aws iot describe-domain-configuration \
  --domain-configuration-name "additionalDataDomain"
```

Ausgabe:

```
{
  "domainConfigurationName": "additionalDataDomain",
  "domainConfigurationArn": "arn:aws:iot:us-
east-1:758EXAMPLE143:domainconfiguration/additionalDataDomain/norpw",
  "domainName": "d055exampleed74y71zfd-ats.beta.us-east-1.iot.amazonaws.com",
  "serverCertificates": [],
  "domainConfigurationStatus": "ENABLED",
  "serviceType": "DATA",
  "domainType": "AWS_MANAGED",
  "lastStatusChangeDate": 1601923783.774
}
```

Weitere Informationen finden Sie unter [Configurable Endpoints](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeDomainConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-endpoint

Das folgende Codebeispiel zeigt die Verwendung des `describe-endpoint`.

AWS CLI

Beispiel 1: Um Ihren aktuellen AWS Endpunkt zu ermitteln

Im folgenden `describe-endpoint` Beispiel wird der AWS Standardendpunkt abgerufen, auf den alle Befehle angewendet werden.

```
aws iot describe-endpoint
```

Ausgabe:

```
{
  "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

Weitere Informationen finden Sie [DescribeEndpoint](#) im AWS IoT Developer Guide.

Beispiel 2: Um Ihren ATS Endpunkt zu ermitteln

Im folgenden `describe-endpoint` Beispiel wird der Amazon Trust Services (ATS) -Endpunkt abgerufen.

```
aws iot describe-endpoint \  
  --endpoint-type iot:Data-ATS
```

Ausgabe:

```
{  
  "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"  
}
```

Weitere Informationen finden Sie unter [X.509-Zertifikate und AWS IoT im AWS IoT Developer Guide](#).

- API-Einheiten finden Sie [DescribeEndpoint](#) in der AWS CLI Befehlsreferenz.

describe-event-configurations

Das folgende Codebeispiel zeigt die Verwendung `describe-event-configurations`.

AWS CLI

Um zu zeigen, welche Ereignistypen veröffentlicht werden

Im folgenden `describe-event-configurations` Beispiel wird die Konfiguration aufgeführt, die steuert, welche Ereignisse generiert werden, wenn etwas hinzugefügt, aktualisiert oder gelöscht wird.

```
aws iot describe-event-configurations
```

Ausgabe:

```
{  
  "eventConfigurations": {  
    "CA_CERTIFICATE": {  
      "Enabled": false  
    },  
    "CERTIFICATE": {  
      "Enabled": false  
    }  
  }  
}
```

```
    },
    "JOB": {
      "Enabled": false
    },
    "JOB_EXECUTION": {
      "Enabled": false
    },
    "POLICY": {
      "Enabled": false
    },
    "THING": {
      "Enabled": false
    },
    "THING_GROUP": {
      "Enabled": false
    },
    "THING_GROUP_HIERARCHY": {
      "Enabled": false
    },
    "THING_GROUP_MEMBERSHIP": {
      "Enabled": false
    },
    "THING_TYPE": {
      "Enabled": false
    },
    "THING_TYPE_ASSOCIATION": {
      "Enabled": false
    }
  }
}
```

Weitere Informationen finden Sie unter [Ereignismeldungen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeEventConfigurations](#) in der AWS CLI Befehlsreferenz.

describe-index

Das folgende Codebeispiel zeigt die Verwendung `describe-index`.

AWS CLI

Um den aktuellen Status des Dingindex abzurufen

Im folgenden `describe-index` Beispiel wird der aktuelle Status des Dingindex abgerufen.

```
aws iot describe-index \  
  --index-name "AWS_Things"
```

Ausgabe:

```
{  
  "indexName": "AWS_Things",  
  "indexStatus": "ACTIVE",  
  "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"  
}
```

Weitere Informationen finden Sie unter [Managing Thing Indexing](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeIndex](#) in der AWS CLI Befehlsreferenz.

describe-job-execution

Das folgende Codebeispiel zeigt die Verwendung `describe-job-execution`.

AWS CLI

Um Ausführungsdetails für einen Job auf einem Gerät abzurufen

Im folgenden `describe-job-execution` Beispiel werden Ausführungsdetails für den angegebenen Job abgerufen.

```
aws iot describe-job-execution \  
  --job-id "example-job-01" \  
  --thing-name "MyRaspberryPi"
```

Ausgabe:

```
{  
  "execution": {  
    "jobId": "example-job-01",  
    "status": "QUEUED",  
    "statusDetails": {},  
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi",  
    "queuedAt": 1560787023.636,  
    "lastUpdatedAt": 1560787023.636,  
    "executionNumber": 1,  
    "versionNumber": 1  
  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeJobExecution](#) in der AWS CLI Befehlsreferenz.

describe-job

Das folgende Codebeispiel zeigt die Verwendung `describe-job`.

AWS CLI

Um den detaillierten Status für einen Job abzurufen

Im folgenden `describe-job` Beispiel wird der detaillierte Status für den Job abgerufen, dessen ID lautet `example-job-01`.

```
aws iot describe-job \  
  --job-id "example-job-01"
```

Ausgabe:

```
{  
  "job": {  
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",  
    "jobId": "example-job-01",  
    "targetSelection": "SNAPSHOT",  
    "status": "IN_PROGRESS",  
    "targets": [  
      "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"  
    ],  
    "description": "example job test",  
    "presignedUrlConfig": {},  
    "jobExecutionsRolloutConfig": {},  
    "createdAt": 1560787022.733,  
    "lastUpdatedAt": 1560787026.294,  
    "jobProcessDetails": {  
      "numberOfCanceledThings": 0,  
      "numberOfSucceededThings": 0,  
      "numberOfFailedThings": 0,  
      "numberOfRejectedThings": 0,  
    }  
  }  
}
```

```
        "numberOfQueuedThings": 1,  
        "numberOfInProgressThings": 0,  
        "numberOfRemovedThings": 0,  
        "numberOfTimedOutThings": 0  
    },  
    "timeoutConfig": {}  
}  
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeJob](#) in der AWS CLI Befehlsreferenz.

describe-mitigation-action

Das folgende Codebeispiel zeigt die Verwendung `describe-mitigation-action`.

AWS CLI

Um die Details für eine definierte Schadensbegrenzungsmaßnahme anzuzeigen

Im folgenden `describe-mitigation-action` Beispiel werden Details für die angegebene Abhilfemaßnahme angezeigt.

```
aws iot describe-mitigation-action \  
  --action-name AddThingsToQuarantineGroupAction
```

Ausgabe:

```
{  
  "actionName": "AddThingsToQuarantineGroupAction",  
  "actionType": "ADD_THINGS_TO_THING_GROUP",  
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/  
AddThingsToQuarantineGroupAction",  
  "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa",  
  "roleArn": "arn:aws:iam::123456789012:role/service-role/  
MoveThingsToQuarantineGroupRole",  
  "actionParams": {  
    "addThingsToThingGroupParams": {  
      "thingGroupNames": [  
        "QuarantineGroup1"  
      ],  
    },  
  },  
}
```

```

        "overrideDynamicGroups": true
    }
},
"creationDate": "2019-12-10T11:09:35.999000-08:00",
"lastModifiedDate": "2019-12-10T11:09:35.999000-08:00"
}

```

Weitere Informationen finden Sie unter [DescribeMitigationAction \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeMitigationAction](#) in der AWS CLI Befehlsreferenz.

describe-provisioning-template-version

Das folgende Codebeispiel zeigt die Verwendung `describe-provisioning-template-version`.

AWS CLI

Um eine Provisioning-Vorlagenversion zu beschreiben

Das folgende `describe-provisioning-template-version` Beispiel beschreibt eine Provisioning-Vorlagenversion.

```

aws iot describe-provisioning-template-version \
  --template-name MyTestProvisioningTemplate \
  --version-id 1

```

Ausgabe:

```

{
  "versionId": 1,
  "creationDate": 1589308310.574,
  "templateBody": "{
    \"Parameters\":{
      \"SerialNumber\":{
        \"Type\":\"String\"
      },
      \"AWS::IoT::Certificate::Id\":{
        \"Type\":\"String\"
      }
    },
    \"Resources\":{
      \"certificate\":{

```



```

        \ "Properties\":{
            \ "CertificateId\":{
                \ "Ref\":"\ "AWS::IoT::Certificate::Id\ "
            },
            \ "Status\":"\ "Active\ "
        },
        \ "Type\":"\ "AWS::IoT::Certificate\ "
    },
    \ "policy\":{
        \ "Properties\":{
            \ "PolicyName\":"\ "MyIotPolicy\ "
        },
        \ "Type\":"\ "AWS::IoT::Policy\ "
    },
    \ "thing\":{
        \ "OverrideSettings\":{
            \ "AttributePayload\":"\ "MERGE\ ",
            \ "ThingGroups\":"\ "DO_NOTHING\ ",
            \ "ThingTypeName\":"\ "REPLACE\ "
        },
        \ "Properties\":{
            \ "AttributePayload\":{ },
            \ "ThingGroups\":[ ],
            \ "ThingName\":{
                \ "Fn::Join\":[
                    \ "\ ",
                    [
                        \ "DemoGroup_\ ",
                        { \ "Ref\":"\ "SerialNumber\ " }
                    ]
                ]
            }
        },
        \ "ThingTypeName\":"\ "VirtualThings\ "
    },
    \ "Type\":"\ "AWS::IoT::Thing\ "
}
}
}
},
"isDefaultVersion": true
}

```

Weitere Informationen finden Sie unter [Bereitstellen von Geräten ohne Gerätezertifikate mithilfe von Fleet Provisioning](#) im AWS IoT Core Developers Guide.

- API-Einheiten finden Sie [DescribeProvisioningTemplateVersion](#) in der AWS CLI Befehlsreferenz.

describe-provisioning-template

Das folgende Codebeispiel zeigt die Verwendung `describe-provisioning-template`.

AWS CLI

Um eine Bereitstellungsvorlage zu beschreiben

Das folgende `describe-provisioning-template` Beispiel beschreibt eine Bereitstellungsvorlage.

```
aws iot describe-provisioning-template \
  --template-name MyTestProvisioningTemplate
```

Ausgabe:

```
{
  "templateArn": "arn:aws:iot:us-west-2:57EXAMPLE833:provisioningtemplate/
MyTestProvisioningTemplate",
  "templateName": "MyTestProvisioningTemplate",
  "creationDate": 1589308310.574,
  "lastModifiedDate": 1589308345.539,
  "defaultVersionId": 1,
  "templateBody": "{
    \"Parameters\":{
      \"SerialNumber\":{
        \"Type\":\"String\"
      },
      \"AWS::IoT::Certificate::Id\":{
        \"Type\":\"String\"
      }
    },
    \"Resources\":{
      \"certificate\":{
        \"Properties\":{
          \"CertificateId\":{
            \"Ref\":\"AWS::IoT::Certificate::Id\"
          },
          \"Status\":\"Active\"
        }
      }
    }
  }
```

```

    },
    \"Type\": \"AWS::IoT::Certificate\"
  },
  \"policy\": {
    \"Properties\": {
      \"PolicyName\": \"MyIotPolicy\"
    },
    \"Type\": \"AWS::IoT::Policy\"
  },
  \"thing\": {
    \"OverrideSettings\": {
      \"AttributePayload\": \"MERGE\",
      \"ThingGroups\": \"DO_NOTHING\",
      \"ThingTypeName\": \"REPLACE\"
    },
    \"Properties\": {
      \"AttributePayload\": {},
      \"ThingGroups\": [],
      \"ThingName\": {
        \"Fn::Join\": [
          \"\",
          [
            \"DemoGroup_\",
            {\"Ref\": \"SerialNumber\"}
          ]
        ]
      },
      \"ThingTypeName\": \"VirtualThings\"
    },
    \"Type\": \"AWS::IoT::Thing\"
  }
}
},
\"enabled\": true,
\"provisioningRoleArn\": \"arn:aws:iam::571032923833:role/service-role/IoT_access\"
}

```

Weitere Informationen finden Sie unter [Bereitstellen von Geräten ohne Gerätezertifikate mithilfe von Fleet Provisioning](#) im AWS IoT Core Developers Guide.

- API-Einheiten finden Sie [DescribeProvisioningTemplate](#) in der AWS CLI Befehlsreferenz.

describe-role-alias

Das folgende Codebeispiel zeigt die Verwendung `describe-role-alias`.

AWS CLI

Um Informationen über einen AWS IoT-Rollenalias abzurufen

Im folgenden `describe-role-alias` Beispiel werden Details für den angegebenen Rollenalias angezeigt.

```
aws iot describe-role-alias \  
  --role-alias LightBulbRole
```

Ausgabe:

```
{  
  "roleAliasDescription": {  
    "roleAlias": "LightBulbRole",  
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/  
LightBulbRole",  
    "roleArn": "arn:aws:iam::123456789012:role/light_bulb_role_001",  
    "owner": "123456789012",  
    "credentialDurationSeconds": 3600,  
    "creationDate": 1570558643.221,  
    "lastModifiedDate": 1570558643.221  
  }  
}
```

Weitere Informationen finden Sie [DescribeRoleAlias](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DescribeRoleAlias](#) unter AWS CLI Befehlsreferenz.

describe-scheduled-audit

Das folgende Codebeispiel zeigt die Verwendung `describe-scheduled-audit`.

AWS CLI

Um Informationen über ein geplantes Audit zu erhalten

Im folgenden `describe-scheduled-audit` Beispiel werden detaillierte Informationen zu einem geplanten AWS IoT Device Defender-Audit mit dem Namen `AWSIoTDeviceDefenderDailyAudit` abgerufen.

```
aws iot describe-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Ausgabe:

```
{  
  "frequency": "DAILY",  
  "targetCheckNames": [  
    "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK",  
    "CONFLICTING_CLIENT_IDS_CHECK",  
    "DEVICE_CERTIFICATE_SHARED_CHECK",  
    "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",  
    "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK",  
    "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK"  
  ],  
  "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",  
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/  
AWSIoTDeviceDefenderDailyAudit"  
}
```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeScheduledAudit](#) in der AWS CLI Befehlsreferenz.

describe-security-profile

Das folgende Codebeispiel zeigt die Verwendung `describe-security-profile`.

AWS CLI

Um Informationen über ein Sicherheitsprofil abzurufen

Im folgenden `describe-security-profile` Beispiel werden Informationen über das AWS IoT Device Defender-Sicherheitsprofil mit dem Namen `PossibleIssue` abgerufen.

```
aws iot describe-security-profile \  
  --security-profile-name PossibleIssue
```

Ausgabe:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
  "securityProfileDescription": "check to see if authorization fails 10 times in 5
minutes or if cellular bandwidth exceeds 128",
  "behaviors": [
    {
      "name": "CellularBandwidth",
      "metric": "aws:message-byte-size",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 128
        },
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    },
    {
      "name": "Authorization",
      "metric": "aws:num-authorization-failures",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 10
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    }
  ],
  "version": 1,
  "creationDate": 1560278102.528,
  "lastModifiedDate": 1560278102.528
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeSecurityProfile](#) in der AWS CLI Befehlsreferenz.

describe-stream

Das folgende Codebeispiel zeigt die Verwendung `describe-stream`.

AWS CLI

Um Informationen über einen Stream zu erhalten

Im folgenden `describe-stream` Beispiel werden die Details zum angegebenen Stream angezeigt.

```
aws iot describe-stream \  
  --stream-id stream12345
```

Ausgabe:

```
{  
  "streamInfo": {  
    "streamId": "stream12345",  
    "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",  
    "streamVersion": 1,  
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
    "files": [  
      {  
        "fileId": "123",  
        "s3Location": {  
          "bucket": "codesign-ota-bucket",  
          "key": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"  
        }  
      }  
    ],  
    "createdAt": 1557863215.995,  
    "lastUpdatedAt": 1557863215.995,  
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"  
  }  
}
```

Weitere Informationen finden Sie [DescribeStream](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [DescribeStream](#) unter AWS CLI Befehlsreferenz.

describe-thing-group

Das folgende Codebeispiel zeigt die Verwendung `describe-thing-group`.

AWS CLI

Um Informationen über eine Dinggruppe zu erhalten

Im folgenden `describe-thing-group` Beispiel werden Informationen über die genannte Dinggruppe abgerufen `HalogenBulbs`.

```
aws iot describe-thing-group \  
  --thing-group-name HalogenBulbs
```

Ausgabe:

```
{  
  "thingGroupName": "HalogenBulbs",  
  "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",  
  "version": 1,  
  "thingGroupProperties": {},  
  "thingGroupMetadata": {  
    "parentGroupName": "LightBulbs",  
    "rootToParentThingGroups": [  
      {  
        "groupName": "LightBulbs",  
        "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/  
LightBulbs"  
      }  
    ],  
    "creationDate": 1559927609.897  
  }  
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DescribeThingGroup](#) in der AWS CLI Befehlsreferenz.

describe-thing-type

Das folgende Codebeispiel zeigt die Verwendung `describe-thing-type`.

AWS CLI

Um Informationen über einen Dingtyp zu erhalten

Im folgenden `describe-thing-type` Beispiel werden Informationen über den angegebenen Dingtyp angezeigt, der in Ihrem AWS Konto definiert ist.

```
aws iot describe-thing-type \  
  --thing-type-name "LightBulb"
```

Ausgabe:

```
{  
  "thingTypeName": "LightBulb",  
  "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190",  
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",  
  "thingTypeProperties": {  
    "thingTypeDescription": "light bulb type",  
    "searchableAttributes": [  
      "model",  
      "wattage"  
    ]  
  },  
  "thingTypeMetadata": {  
    "deprecated": false,  
    "creationDate": 1559772562.498  
  }  
}
```

Weitere Informationen finden Sie unter [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DescribeThingType](#) in der AWS CLI Befehlsreferenz.

`describe-thing`

Das folgende Codebeispiel zeigt die Verwendung `describe-thing`.

AWS CLI

Um detaillierte Informationen zu einer Sache anzuzeigen

Im folgenden `describe-thing` Beispiel werden Informationen zu einer Sache (einem Gerät) angezeigt, die in der AWS IoT-Registrierung für Ihr AWS Konto definiert ist.

```
aws iot describe-thing --thing-name "MyLightBulb"
```

Ausgabe:

```
{
  "defaultClientId": "MyLightBulb",
  "thingName": "MyLightBulb",
  "thingId": "40da2e73-c6af-406e-b415-15acae538797",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
  "thingTypeName": "LightBulb",
  "attributes": {
    "model": "123",
    "wattage": "75"
  },
  "version": 1
}
```

Weitere Informationen finden Sie unter [How to Manage Things with the Registry](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DescribeThing](#) in der AWS CLI Befehlsreferenz.

detach-policy

Das folgende Codebeispiel zeigt die Verwendung `detach-policy`.

AWS CLI

Beispiel 1: Um eine AWS IoT-Richtlinie von einer Dinggruppe zu trennen

Im folgenden `detach-policy` Beispiel wird die angegebene Richtlinie von einer Dinggruppe und damit von allen Dingen in dieser Gruppe und allen untergeordneten Gruppen der Gruppe getrennt.

```
aws iot detach-policy \
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
  --policy-name "MyFirstGroup_Core-policy"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

Beispiel 2: So trennen Sie eine AWS IoT-Richtlinie von einem Gerätezertifikat

Im folgenden detach-policy Beispiel wird die TemperatureSensorPolicy Richtlinie von einem Gerätezertifikat getrennt, das durch identifiziert wird. ARN

```
aws iot detach-policy \  
  --policy-name TemperatureSensorPolicy \  
  --target arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- APIEinzelheiten finden Sie [DetachPolicy](#) in der AWS CLI Befehlsreferenz.

detach-security-profile

Das folgende Codebeispiel zeigt die Verwendung detach-security-profile.

AWS CLI

Um die Zuordnung eines Sicherheitsprofils zu einem Ziel aufzuheben

Im folgenden detach-security-profile Beispiel wird die Zuordnung zwischen dem genannten AWS IoT Device Defender-Sicherheitsprofil Testprofile und dem Ziel „Alle registrierten Dinge“ entfernt.

```
aws iot detach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/registered-things"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- APIEinzelheiten finden Sie [DetachSecurityProfile](#) in der AWS CLI Befehlsreferenz.

detach-thing-principal

Das folgende Codebeispiel zeigt die Verwendung detach-thing-principal.

AWS CLI

Um ein Zertifikat/einen Prinzipal von einer Sache zu trennen

Im folgenden `detach-thing-principal` Beispiel wird ein Zertifikat, das einen Prinzipal darstellt, aus dem angegebenen Objekt entfernt.

```
aws iot detach-thing-principal \  
  --thing-name "MyLightBulb" \  
  --principal "arn:aws:iot:us-  
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [How to Manage Things with the Registry](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DetachThingPrincipal](#) in der AWS CLI Befehlsreferenz.

disable-topic-rule

Das folgende Codebeispiel zeigt die Verwendung `disable-topic-rule`.

AWS CLI

Um eine Themenregel zu deaktivieren

Im folgenden `disable-topic-rule` Beispiel wird die angegebene Themenregel deaktiviert.

```
aws iot disable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Regeln anzeigen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DisableTopicRule](#) in der AWS CLI Befehlsreferenz.

enable-topic-rule

Das folgende Codebeispiel zeigt die Verwendung `enable-topic-rule`.

AWS CLI

Um eine Themenregel zu aktivieren

Im folgenden `enable-topic-rule` Beispiel wird die angegebene Themenregel aktiviert (oder erneut aktiviert).

```
aws iot enable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Regeln anzeigen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [EnableTopicRule](#) in der AWS CLI Befehlsreferenz.

get-behavior-model-training-summaries

Das folgende Codebeispiel zeigt die Verwendung von `get-behavior-model-training-summaries`.

AWS CLI

Um den Status eines Trainingsmodells für das ML Detect Security Profile von Device Defender aufzulisten

Im folgenden `get-behavior-model-training-summaries` Beispiel wird der Trainingsstatus des Modells für die konfigurierten Verhaltensweisen im ausgewählten Sicherheitsprofil aufgeführt. Für jedes Verhalten werden der Name, der Modellstatus und der Prozentsatz der gesammelten Datenpunkte aufgeführt.

```
aws iot get-behavior-model-training-summaries \  
  --security-profile-name MySecuirtyProfileName
```

Ausgabe:

```
{  
  "summaries": [  
    {  
      "securityProfileName": "MySecuirtyProfileName",  
      "behaviorName": "Messages_sent_ML_behavior",  
      "modelStatus": "PENDING_BUILD",  
      "datapointsCollectionPercentage": 0.0
```

```

    },
    {
      "securityProfileName": "MySecuirtyProfileName",
      "behaviorName": "Messages_received_ML_behavior",
      "modelStatus": "PENDING_BUILD",
      "datapointsCollectionPercentage": 0.0
    },
    {
      "securityProfileName": "MySecuirtyProfileName",
      "behaviorName": "Authorization_failures_ML_behavior",
      "modelStatus": "PENDING_BUILD",
      "datapointsCollectionPercentage": 0.0
    },
    {
      "securityProfileName": "MySecuirtyProfileName",
      "behaviorName": "Message_size_ML_behavior",
      "modelStatus": "PENDING_BUILD",
      "datapointsCollectionPercentage": 0.0
    },
    {
      "securityProfileName": "MySecuirtyProfileName",
      "behaviorName": "Connection_attempts_ML_behavior",
      "modelStatus": "PENDING_BUILD",
      "datapointsCollectionPercentage": 0.0
    },
    {
      "securityProfileName": "MySPNoALerts",
      "behaviorName": "Disconnects_ML_behavior",
      "modelStatus": "PENDING_BUILD",
      "datapointsCollectionPercentage": 0.0
    }
  ]
}

```

Weitere Informationen finden Sie unter [GetBehaviorModelTrainingSummaries \(Befehle erkennen\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetBehaviorModelTrainingSummaries](#) in der AWS CLI Befehlsreferenz.

get-cardinality

Das folgende Codebeispiel zeigt die Verwendung `get-cardinality`.

AWS CLI

Um die ungefähre Anzahl der eindeutigen Werte zurückzugeben, die der Abfrage entsprechen

Sie können das folgende Setup-Skript verwenden, um 10 Objekte zu erstellen, die 10 Temperatursensoren repräsentieren. Jedes neue Ding hat 3 Attribute.

```
# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
Racks=(Rack1 Rack1 Rack2 Rack2 Rack3 Rack4 Rack5 Rack6 Rack6 Rack6)
IsNormal=(true true true true true true false false false false)
for ((i=0; i<10 ; i++))
do
  thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
  attributes="{temperature=${Temperatures[i]},rackId=${Racks[i]},stateNormal=
  ${IsNormal[i]}}")
  aws iot describe-thing --thing-name "TempSensor$i"
done
```

Beispielausgabe des Setup-Skripts:

```
{
  "version": 1,
  "thingName": "TempSensor0",
  "defaultClientId": "TempSensor0",
  "attributes": {
    "rackId": "Rack1",
    "stateNormal": "true",
    "temperature": "70"
  },
  "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
  "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

Das folgende `get-cardinality` Beispiel fragt die 10 vom Setup-Skript erstellten Sensoren ab und gibt die Anzahl der Racks zurück, deren Temperatursensoren abnormale Temperaturwerte melden. Wenn der Temperaturwert unter 60 oder über 80 liegt, befindet sich der Temperatursensor in einem abnormalen Zustand.

```
aws iot get-cardinality \
  --aggregation-field "attributes.rackId" \
```

```
--query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

Ausgabe:

```
{
  "cardinality": 2
}
```

Weitere Informationen finden Sie unter Abfragen von Aggregatdaten < <https://docs.aws.amazon.com/iot/latest/developerguide/index-aggregate.html> > im IoT Developer Guide.AWS

- Einzelheiten API finden Sie in der Befehlsreferenz. [GetCardinality](#)AWS CLI

get-effective-policies

Das folgende Codebeispiel zeigt die Verwendung `get-effective-policies`.

AWS CLI

Um die Richtlinien aufzulisten, die sich auf eine Sache auswirken

Das folgende `get-effective-policies` Beispiel listet die Richtlinien auf, die sich auf das angegebene Ding auswirken, einschließlich der Richtlinien, die allen Gruppen zugeordnet sind, zu denen es gehört.

```
aws iot get-effective-policies \
  --thing-name TemperatureSensor-001 \
  --principal arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Ausgabe:

```
{
  "effectivePolicies": [
    {
      "policyName": "TemperatureSensorPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
      "policyDocument": "{
        \"Version\": \"2012-10-17\",
```



```

    \"Statement\": [
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Publish\",
          \"iot:Receive\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Subscribe\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Connect\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub
\"
        ]
      }
    ]
  }
}

```

Weitere Informationen finden [Sie unter Get Effective Policies for a Thing](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetEffectivePolicies](#) in der AWS CLI Befehlsreferenz.

get-indexing-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-indexing-configuration`.

AWS CLI

Um die Konfiguration für die Dingindizierung zu erhalten

Im folgenden `get-indexing-configuration` Beispiel werden die aktuellen Konfigurationsdaten für die AWS IoT-Flottenindizierung abgerufen.

```
aws iot get-indexing-configuration
```

Ausgabe:

```
{
  "thingIndexingConfiguration": {
    "thingIndexingMode": "OFF",
    "thingConnectivityIndexingMode": "OFF"
  },
  "thingGroupIndexingConfiguration": {
    "thingGroupIndexingMode": "OFF"
  }
}
```

Weitere Informationen finden Sie unter [Managing Thing Indexing](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetIndexingConfiguration](#) in der AWS CLI Befehlsreferenz.

get-job-document

Das folgende Codebeispiel zeigt die Verwendung `get-job-document`.

AWS CLI

Um das Dokument für einen Job abzurufen

Im folgenden `get-job-document` Beispiel werden Details zu dem Dokument für den Job angezeigt, dessen ID lautet `example-job-01`.

```
aws iot get-job-document \
```

```
--job-id "example-job-01"
```

Ausgabe:

```
{
  "document": "\n{\n  \"operation\": \"customJob\", \n  \"otherInfo\":\n  \"someValue\"\n}\n"
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetJobDocument](#) in der AWS CLI Befehlsreferenz.

get-logging-options

Das folgende Codebeispiel zeigt die Verwendung `get-logging-options`.

AWS CLI

Um die Protokollierungsoptionen zu erhalten

Im folgenden `get-logging-options` Beispiel werden die aktuellen Protokollierungsoptionen für Ihr AWS Konto abgerufen.

```
aws iot get-logging-options
```

Ausgabe:

```
{
  "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",
  "logLevel": "ERROR"
}
```

Weitere Informationen finden Sie im Titel im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetLoggingOptions](#) in der AWS CLI Befehlsreferenz.

get-ota-update

Das folgende Codebeispiel zeigt die Verwendung `get-ota-update`.

AWS CLI

Um Informationen über ein OTA Update abzurufen

Im folgenden `get-ota-update` Beispiel werden Details zum angegebenen OTA Update angezeigt.

```
aws iot get-ota-update \  
  --ota-update-id ota12345
```

Ausgabe:

```
{  
  "otaUpdateInfo": {  
    "otaUpdateId": "ota12345",  
    "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",  
    "creationDate": 1557863215.995,  
    "lastModifiedDate": 1557863215.995,  
    "description": "A critical update needed right away.",  
    "targets": [  
      "device1",  
      "device2",  
      "device3",  
      "device4"  
    ],  
    "targetSelection": "SNAPSHOT",  
    "protocols": ["HTTP"],  
    "awsJobExecutionsRolloutConfig": {  
      "maximumPerMinute": 10  
    },  
    "otaUpdateFiles": [  
      {  
        "fileName": "firmware.bin",  
        "fileLocation": {  
          "stream": {  
            "streamId": "004",  
            "fileId": 123  
          }  
        },  
        "codeSigning": {  
          "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"  
        }  
      }  
    ]  
  }  
}
```

```

    ],
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
    "otaUpdateStatus": "CREATE_COMPLETE",
    "awsIotJobId": "job54321",
    "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/job54321",
    "errorInfo": {
    }
  }
}

```

Weitere Informationen finden Sie unter [GetOTAUpdate](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [GetOtaUpdate](#) unter AWS CLI Befehlsreferenz.

get-percentiles

Das folgende Codebeispiel zeigt die Verwendung `get-percentiles`.

AWS CLI

Um die aggregierten Werte, die der Abfrage entsprechen, in Perzentilgruppierungen zu gruppieren

Sie können das folgende Setup-Skript verwenden, um 10 Dinge zu erstellen, die 10 Temperatursensoren repräsentieren. Jedes neue Ding hat 1 Attribut.

```

# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
for ((i=0; i<10 ; i++))
do
    thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
attributes="{temperature=${Temperatures[i]}}")
    aws iot describe-thing --thing-name "TempSensor$i"
done

```

Beispielausgabe des Setup-Skripts:

```

{
  "version": 1,
  "thingName": "TempSensor0",
  "defaultClientId": "TempSensor0",
  "attributes": {
    "temperature": "70"
  },
}

```

```
"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
"thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

Im folgenden `get-percentiles` Beispiel werden die 10 vom Setup-Skript erstellten Sensoren abgefragt und für jede angegebene Perzentilgruppe ein Wert zurückgegeben. Die Perzentilgruppe „10“ enthält den aggregierten Feldwert, der in etwa 10 Prozent der Werte vorkommt, die der Abfrage entsprechen. In der folgenden Ausgabe bedeutet {"Prozent": 10,0, „Wert“: 67,7}, dass ungefähr 10,0% der Temperaturwerte unter 67,7 liegen.

```
aws iot get-percentiles \
  --aggregation-field "attributes.temperature" \
  --query-string "thingName:TempSensor*" \
  --percents 10 25 50 75 90
```

Ausgabe:

```
{
  "percentiles": [
    {
      "percent": 10.0,
      "value": 67.7
    },
    {
      "percent": 25.0,
      "value": 71.25
    },
    {
      "percent": 50.0,
      "value": 73.5
    },
    {
      "percent": 75.0,
      "value": 91.5
    },
    {
      "percent": 90.0,
      "value": 98.1
    }
  ]
}
```

Weitere Informationen finden Sie unter [Abfragen aggregierter Daten](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetPercentiles](#) in der AWS CLI Befehlsreferenz.

get-policy-version

Das folgende Codebeispiel zeigt die Verwendung `get-policy-version`.

AWS CLI

Um Informationen zu einer bestimmten Version einer Richtlinie abzurufen

Im folgenden `get-policy-version` Beispiel werden Informationen zur ersten Version der angegebenen Richtlinie abgerufen.

```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id "1"
```

Ausgabe:

```
{  
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",  
  "policyName": "UpdateDeviceCertPolicy",  
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
  \"Allow\", \"Action\": \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",  
  "policyVersionId": "1",  
  "isDefaultVersion": false,  
  "creationDate": 1559925941.924,  
  "lastModifiedDate": 1559926175.458,  
  "generationId":  
  "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"  
}
```

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetPolicyVersion](#) in der AWS CLI Befehlsreferenz.

get-policy

Das folgende Codebeispiel zeigt die Verwendung `get-policy`.

AWS CLI

Um Informationen über die Standardversion einer Richtlinie abzurufen

Im folgenden `get-policy` Beispiel werden Informationen über die Standardversion der angegebenen Richtlinie abgerufen.

```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy
```

Ausgabe:

```
{  
  "policyName": "UpdateDeviceCertPolicy",  
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",  
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
  \"Allow\", \"Action\": \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",  
  "defaultVersionId": "2",  
  "creationDate": 1559925941.924,  
  "lastModifiedDate": 1559925941.924,  
  "generationId":  
  "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"  
}
```

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetPolicy](#) in der AWS CLI Befehlsreferenz.

get-registration-code

Das folgende Codebeispiel zeigt die Verwendung `get-registration-code`.

AWS CLI

Um Ihren AWS kontospezifischen Registrierungscode zu erhalten

Im folgenden `get-registration-code` Beispiel wird Ihr AWS kontospezifischer Registrierungscode abgerufen.

```
aws iot get-registration-code
```


Ausgabe:

```
{
  "registrationCode":
  "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

Weitere Informationen finden Sie unter [Verwenden Sie Ihr eigenes Zertifikat](#) im AWS IoT-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetRegistrationCode](#) in der AWS CLI Befehlsreferenz.

get-statistics

Das folgende Codebeispiel zeigt die Verwendung `get-statistics`.

AWS CLI

Um den Geräteindex nach aggregierten Daten zu durchsuchen

Im folgenden `get-statistics` Beispiel wird die Anzahl der Dinge zurückgegeben, für die in ihrem Geräteshadow eine Eigenschaft namens `connectivity.connected` gesetzt ist `false` (d. h. die Anzahl der Geräte, die nicht verbunden sind).

```
aws iot get-statistics \
  --index-name AWS_Things \
  --query-string "connectivity.connected:false"
```

Ausgabe:

```
{
  "statistics": {
    "count": 6
  }
}
```

Weitere Informationen finden Sie unter [Getting Statistics About Your Device Fleet](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetStatistics](#) in der AWS CLI Befehlsreferenz.

get-topic-rule-destination

Das folgende Codebeispiel zeigt die Verwendung `get-topic-rule-destination`.

AWS CLI

Um ein Ziel für eine Themenregel abzurufen

Im folgenden `get-topic-rule-destination` Beispiel werden Informationen zu einem Ziel für Themenregeln abgerufen.

```
aws iot get-topic-rule-destination \
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Ausgabe:

```
{
  "topicRuleDestination": {
    "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "status": "DISABLED",
    "httpUrlProperties": {
      "confirmationUrl": "https://example.com"
    }
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Themenregelzielen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetTopicRuleDestination](#) in der AWS CLI Befehlsreferenz.

get-topic-rule

Das folgende Codebeispiel zeigt die Verwendung `get-topic-rule`.

AWS CLI

Um Informationen über eine Regel zu erhalten

Im folgenden `get-topic-rule` Beispiel werden Informationen zur angegebenen Regel abgerufen.

```
aws iot get-topic-rule \
  --rule-name MyRPiLowMoistureAlertRule
```

Ausgabe:

```
{
  "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/MyRPiLowMoistureAlertRule",
  "rule": {
    "ruleName": "MyRPiLowMoistureAlertRule",
    "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
state.reported.moisture = 'low'\n          ",
    "description": "Sends an alert whenever soil moisture level readings are too
low.",
    "createdAt": 1558624363.0,
    "actions": [
      {
        "sns": {
          "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
          "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
          "messageFormat": "RAW"
        }
      }
    ],
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23"
  }
}
```

Weitere Informationen finden Sie unter [Regeln anzeigen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetTopicRule](#) in der AWS CLI Befehlsreferenz.

get-v2-logging-options

Das folgende Codebeispiel zeigt die Verwendung von `get-v2-logging-options`.

AWS CLI

Um die aktuellen Protokollierungsoptionen aufzulisten

Das folgende `get-v2-logging-options` Beispiel listet die aktuellen Protokollierungsoptionen für AWS IoT auf.

```
aws iot get-v2-logging-options
```

Ausgabe:

```
{
  "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",
  "defaultLogLevel": "WARN",
  "disableAllLogs": false
}
```

Weitere Informationen finden Sie im Titel im AWS IoT Developer Guide.

- API-Einheiten finden Sie unter [GetV2 LoggingOptions](#) in der AWS CLI Befehlsreferenz.

list-active-violations

Das folgende Codebeispiel zeigt die Verwendung `list-active-violations`.

AWS CLI

Um die aktiven Verstöße aufzulisten

Das folgende `list-active-violations` Beispiel listet alle Verstöße für das angegebene Sicherheitsprofil auf.

```
aws iot list-active-violations \
  --security-profile-name Testprofile
```

Ausgabe:

```
{
  "activeViolations": [
    {
      "violationId": "174db59167fa474c80a652ad1583fd44",
      "thingName": "iotconsole-1560269126751-1",
      "securityProfileName": "Testprofile",
      "behavior": {
        "name": "Authorization",
        "metric": "aws:num-authorization-failures",

```

```

        "criteria": {
            "comparisonOperator": "greater-than",
            "value": {
                "count": 10
            },
            "durationSeconds": 300,
            "consecutiveDatapointsToAlarm": 1,
            "consecutiveDatapointsToClear": 1
        }
    },
    "lastViolationValue": {
        "count": 0
    },
    "lastViolationTime": 1560293700.0,
    "violationStartTime": 1560279000.0
},
{
    "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
    "thingName": "TvnQoEoU",
    "securityProfileName": "Testprofile",
    "behavior": {
        "name": "CellularBandwidth",
        "metric": "aws:message-byte-size",
        "criteria": {
            "comparisonOperator": "greater-than",
            "value": {
                "count": 128
            },
            "consecutiveDatapointsToAlarm": 1,
            "consecutiveDatapointsToClear": 1
        }
    },
    "lastViolationValue": {
        "count": 110
    },
    "lastViolationTime": 1560369000.0,
    "violationStartTime": 1560276600.0
},
{
    "violationId": "74aa393adea02e6648f3ac362beed55e",
    "thingName": "iotconsole-1560269232412-2",
    "securityProfileName": "Testprofile",
    "behavior": {
        "name": "Authorization",

```

```

        "metric": "aws:num-authorization-failures",
        "criteria": {
            "comparisonOperator": "greater-than",
            "value": {
                "count": 10
            },
            "durationSeconds": 300,
            "consecutiveDatapointsToAlarm": 1,
            "consecutiveDatapointsToClear": 1
        }
    },
    "lastViolationValue": {
        "count": 0
    },
    "lastViolationTime": 1560276600.0,
    "violationStartTime": 1560276600.0
},
{
    "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
    "thingName": "TvnQoEoU",
    "securityProfileName": "Testprofile",
    "behavior": {
        "name": "Authorization",
        "metric": "aws:num-authorization-failures",
        "criteria": {
            "comparisonOperator": "greater-than",
            "value": {
                "count": 10
            },
            "durationSeconds": 300,
            "consecutiveDatapointsToAlarm": 1,
            "consecutiveDatapointsToClear": 1
        }
    },
    "lastViolationValue": {
        "count": 0
    },
    "lastViolationTime": 1560369000.0,
    "violationStartTime": 1560276600.0
}
]
}

```

- API-Einheiten finden Sie [ListActiveViolations](#) unter AWS CLI Befehlsreferenz.

list-attached-policies

Das folgende Codebeispiel zeigt die Verwendung `list-attached-policies`.

AWS CLI

Beispiel 1: Um die Richtlinien aufzulisten, die einer Gruppe zugeordnet sind

Das folgende `list-attached-policies` Beispiel listet die Richtlinien auf, die der angegebenen Gruppe zugeordnet sind.

```
aws iot list-attached-policies \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
```

Ausgabe:

```
{  
  "policies": [  
    {  
      "policyName": "UpdateDeviceCertPolicy",  
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
UpdateDeviceCertPolicy"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

Beispiel 2: Um die Richtlinien aufzulisten, die mit einem Gerätezertifikat verknüpft sind

Das folgende `list-attached-policies` Beispiel listet die AWS IoT-Richtlinien auf, die mit dem Gerätezertifikat verknüpft sind. Das Zertifikat wird durch seinen identifiziertARN.

```
aws iot list-attached-policies \  
  --target arn:aws:iot:us-  
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Ausgabe:

```
{  
  "policies": [  
    {  
      "policyName": "TemperatureSensorPolicy",
```

```
        "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy"
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListAttachedPolicies](#) in der AWS CLI Befehlsreferenz.

list-audit-findings

Das folgende Codebeispiel zeigt die Verwendung `list-audit-findings`.

AWS CLI

Beispiel 1: Um alle Ergebnisse eines Audits aufzulisten

Das folgende `list-audit-findings` Beispiel listet alle Ergebnisse eines AWS IoT Device Defender Defender-Audits mit einer angegebenen Task-ID auf.

```
aws iot list-audit-findings \
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Ausgabe:

```
{
  "findings": []
}
```

Beispiel 2: Um Ergebnisse für einen Audit-Check-Typ aufzulisten

Das folgende `list-audit-findings` Beispiel zeigt Ergebnisse von AWS IoT Device Defender Defender-Audits, die zwischen dem 5. Juni 2019 und dem 19. Juni 2019 stattfanden und bei denen Geräte ein Gerätezertifikat gemeinsam nutzen. Wenn Sie einen Schecknamen angeben, müssen Sie eine Start- und Endzeit angeben.

```
aws iot list-audit-findings \
  --check-name DEVICE_CERTIFICATE_SHARED_CHECK \
  --start-time 1559747125 \
  --end-time 1560962028
```


Ausgabe:

```
{
  "findings": [
    {
      "taskId": "eeef61068b0eb03c456d746c5a26ee04",
      "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
      "taskStartTime": 1560161017.172,
      "findingTime": 1560161017.592,
      "severity": "CRITICAL",
      "nonCompliantResource": {
        "resourceType": "DEVICE_CERTIFICATE",
        "resourceIdentifier": {
          "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
        }
      },
      "relatedResources": [
        {
          "resourceType": "CLIENT_ID",
          "resourceIdentifier": {
            "clientId": "ZipxgAll"
          },
          "additionalInfo": {
            "CONNECTION_TIME": "1560086374068"
          }
        },
        {
          "resourceType": "CLIENT_ID",
          "resourceIdentifier": {
            "clientId": "ZipxgAll"
          },
          "additionalInfo": {
            "CONNECTION_TIME": "1560081552187",
            "DISCONNECTION_TIME": "1560086371552"
          }
        },
        {
          "resourceType": "CLIENT_ID",
          "resourceIdentifier": {
            "clientId": "ZipxgAll"
          },
          "additionalInfo": {
            "CONNECTION_TIME": "1559289863631",

```

```

        "DISCONNECTION_TIME": "1560081532716"
      }
    }
  ],
  "reasonForNonCompliance": "Certificate shared by one or more devices.",
  "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
},
{
  "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
  "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
  "taskStartTime": 1559988217.27,
  "findingTime": 1559988217.655,
  "severity": "CRITICAL",
  "nonCompliantResource": {
    "resourceType": "DEVICE_CERTIFICATE",
    "resourceIdentifier": {
      "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
    }
  },
  "relatedResources": [
    {
      "resourceType": "CLIENT_ID",
      "resourceIdentifier": {
        "clientId": "xShGENLW"
      },
      "additionalInfo": {
        "CONNECTION_TIME": "1559972350825"
      }
    },
    {
      "resourceType": "CLIENT_ID",
      "resourceIdentifier": {
        "clientId": "xShGENLW"
      },
      "additionalInfo": {
        "CONNECTION_TIME": "1559255062002",
        "DISCONNECTION_TIME": "1559972350616"
      }
    }
  ],
  "reasonForNonCompliance": "Certificate shared by one or more devices.",
  "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
},

```

```

    {
      "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
      "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
      "taskStartTime": 1559901817.31,
      "findingTime": 1559901817.767,
      "severity": "CRITICAL",
      "nonCompliantResource": {
        "resourceType": "DEVICE_CERTIFICATE",
        "resourceIdentifier": {
          "deviceCertificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
        }
      },
      "relatedResources": [
        {
          "resourceType": "CLIENT_ID",
          "resourceIdentifier": {
            "clientId": "TvnQoEoU"
          },
          "additionalInfo": {
            "CONNECTION_TIME": "1559826729768"
          }
        },
        {
          "resourceType": "CLIENT_ID",
          "resourceIdentifier": {
            "clientId": "TvnQoEoU"
          },
          "additionalInfo": {
            "CONNECTION_TIME": "1559345920964",
            "DISCONNECTION_TIME": "1559826728402"
          }
        }
      ],
      "reasonForNonCompliance": "Certificate shared by one or more devices.",
      "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListAuditFindings](#) in der AWS CLI Befehlsreferenz.

list-audit-mitigation-actions-executions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-audit-mitigation-actions-executions`.

AWS CLI

Um die Einzelheiten der Ausführung einer Audit-Minderungsmaßnahme aufzulisten

Eine Aufgabe zur Audit-Abwehr wendet eine Minderungsmaßnahme auf ein oder mehrere Ergebnisse eines AWS IoT Device Defender Defender-Audits an. Im folgenden `list-audit-mitigation-actions-executions` Beispiel werden die Details für die Abhilfemaßnahme mit dem angegebenen Ergebnis `taskId` und für das angegebene Ergebnis aufgeführt.

```
aws iot list-audit-mitigation-actions-executions \  
  --task-id myActionsTaskId \  
  --finding-id 0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464
```

Ausgabe:

```
{  
  "actionsExecutions": [  
    {  
      "taskId": "myActionsTaskId",  
      "findingId": "0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464",  
      "actionName": "ResetPolicyVersionAction",  
      "actionId": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",  
      "status": "COMPLETED",  
      "startTime": "2019-12-10T15:19:13.279000-08:00",  
      "endTime": "2019-12-10T15:19:13.337000-08:00"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [ListAuditMitigationActionsExecutions \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListAuditMitigationActionsExecutions](#) in der AWS CLI Befehlsreferenz.

list-audit-mitigation-actions-tasks

Das folgende Codebeispiel zeigt, wie man es benutzt `list-audit-mitigation-actions-tasks`.

AWS CLI

Um die Aufgaben von Audits zur Risikominderung aufzulisten

Im folgenden `list-audit-mitigation-actions-tasks` Beispiel sind die Minderungsmaßnahmen aufgeführt, die innerhalb des angegebenen Zeitraums auf Ergebnisse angewendet wurden.

```
aws iot list-audit-mitigation-actions-tasks \  
  --start-time 1594157400 \  
  --end-time 1594157430
```

Ausgabe:

```
{  
  "tasks": [  
    {  
      "taskId": "0062f2d6-3999-488f-88c7-bef005414103",  
      "startTime": "2020-07-07T14:30:15.172000-07:00",  
      "taskStatus": "COMPLETED"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [ListAuditMitigationActionsTasks \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListAuditMitigationActionsTasks](#) in der AWS CLI Befehlsreferenz.

list-audit-suppressions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-audit-suppressions`.

AWS CLI

Um alle Unterdrückungen aufzulisten, bei denen ein Audit festgestellt wurde

Das folgende `list-audit-suppressions` Beispiel listet alle aktiven Unterdrückungen von Prüfungsergebnissen auf.

```
aws iot list-audit-suppressions
```

Ausgabe:

```
{
  "suppressions": [
    {
      "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
      "resourceIdentifier": {
        "deviceCertificateId": "c7691e<shortened>"
      },
      "expirationDate": 1597881600.0,
      "suppressIndefinitely": false
    }
  ]
}
```

Weitere Informationen finden Sie unter [Audit finding suppressions](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListAuditSuppressions](#) in der AWS CLI Befehlsreferenz.

list-audit-tasks

Das folgende Codebeispiel zeigt, wie man es benutzt `list-audit-tasks`.

AWS CLI

Um alle Ergebnisse eines Audits aufzulisten

Im folgenden `list-audit-tasks` Beispiel sind die Prüfungsaufgaben aufgeführt, die zwischen dem 5. Juni 2019 und dem 12. Juni 2019 ausgeführt wurden.

```
aws iot list-audit-tasks \
  --start-time 1559747125 \
  --end-time 1560357228
```

Ausgabe:

```
{
  "tasks": [
    {
      "taskId": "a3aea009955e501a31b764abe1bebd3d",
```

```
    "taskStatus": "COMPLETED",
    "taskType": "ON_DEMAND_AUDIT_TASK"
  },
  {
    "taskId": "f76b4b5102b632cd9ae38a279c266da1",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "51d9967d9f9ff4d26529505f6d2c444a",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "eef61068b0eb03c456d746c5a26ee04",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "041c49557b7c7b04c079a49514b55589",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "82c7f2afac1562d18a4560be73998acc",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "ac9086b7222a2f5e2e17bb6fd30b3aeb",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  }
]
```

```
}
```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListAuditTasks](#) in der AWS CLI Befehlsreferenz.

list-authorizers

Das folgende Codebeispiel zeigt, wie man es benutzt `list-authorizers`.

AWS CLI

Um Ihren benutzerdefinierten Autorisierer aufzulisten

Im folgenden `list-authorizers` Beispiel werden die benutzerdefinierten Autorisierer in Ihrem Konto aufgeführt. AWS

```
aws iot list-authorizers
```

Ausgabe:

```
{
  "authorizers": [
    {
      "authorizerName": "CustomAuthorizer",
      "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer"
    },
    {
      "authorizerName": "CustomAuthorizer2",
      "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer2"
    },
    {
      "authorizerName": "CustomAuthorizer3",
      "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer3"
    }
  ]
}
```

Weitere Informationen finden Sie [ListAuthorizers](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListAuthorizers](#) unter AWS CLI Befehlsreferenz.

list-billing-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `list-billing-groups`.

AWS CLI

Um die Abrechnungsgruppen für Ihr AWS Konto und Ihre Region aufzulisten

Im folgenden `list-billing-groups` Beispiel werden alle Abrechnungsgruppen aufgeführt, die für Ihr AWS Konto und Ihre AWS Region definiert sind.

```
aws iot list-billing-groups
```

Ausgabe:

```
{
  "billingGroups": [
    {
      "groupName": "GroupOne",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListBillingGroups](#) in der AWS CLI Befehlsreferenz.

list-ca-certificates

Das folgende Codebeispiel zeigt, wie man es benutzt `list-ca-certificates`.

AWS CLI

Um die in Ihrem AWS Konto registrierten CA-Zertifikate aufzulisten

Das folgende `list-ca-certificates` Beispiel listet die CA-Zertifikate auf, die in Ihrem AWS Konto registriert sind.

```
aws iot list-ca-certificates
```

Ausgabe:

```
{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "status": "INACTIVE",
      "creationDate": 1569365372.053
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwenden Sie Ihr eigenes Zertifikat](#) im AWS IoT-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListCaCertificates](#) in der AWS CLI Befehlsreferenz.

list-certificates-by-ca

Das folgende Codebeispiel zeigt, wie man es benutzt `list-certificates-by-ca`.

AWS CLI

Um alle mit einem CA-Zertifikat signierten Gerätezertifikate aufzulisten

Im folgenden `list-certificates-by-ca` Beispiel werden alle Gerätezertifikate in Ihrem AWS Konto aufgeführt, die mit dem angegebenen CA-Zertifikat signiert sind.

```
aws iot list-certificates-by-ca \
  --ca-certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Ausgabe:

```
{
  "certificates": [
    {
```

```

        "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
        "certificateId":
"488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
        "status": "ACTIVE",
        "creationDate": 1569363250.557
    }
]
}

```

Weitere Informationen finden Sie unter [ListCertificatesByCA](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListCertificatesByCa](#) unter AWS CLI Befehlsreferenz.

list-certificates

Das folgende Codebeispiel zeigt, wie man es benutzt `list-certificates`.

AWS CLI

Beispiel 1: Um die in Ihrem AWS Konto registrierten Zertifikate aufzulisten

Das folgende `list-certificates` Beispiel listet alle in Ihrem Konto registrierten Zertifikate auf. Wenn Sie mehr als das standardmäßige Paging-Limit von 25 haben, können Sie den `nextMarker` Antwortwert aus diesem Befehl verwenden und ihn dem nächsten Befehl übergeben, um den nächsten Stapel von Ergebnissen zu erhalten. Wiederholen Sie den Vorgang, bis kein Wert `nextMarker` zurückgegeben wird.

```
aws iot list-certificates
```

Ausgabe:

```

{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
      "certificateId":
"604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
      "status": "ACTIVE",
      "creationDate": 1556810537.617
    },

```

```

    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
      "certificateId":
"262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
      "status": "ACTIVE",
      "creationDate": 1546447050.885
    },
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/
b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
      "certificateId":
"b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
      "status": "ACTIVE",
      "creationDate": 1546292258.322
    },
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
      "certificateId":
"7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
      "status": "ACTIVE",
      "creationDate": 1541457693.453
    },
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
      "certificateId":
"54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
      "status": "ACTIVE",
      "creationDate": 1541113568.611
    },
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
      "certificateId":
"4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
      "status": "ACTIVE",
      "creationDate": 1541022751.983
    }
  ]
}

```

- API-Einheiten finden Sie [ListCertificates](#) in der AWS CLI Befehlsreferenz.

list-custom-metrics

Das folgende Codebeispiel zeigt, wie man es benutzt `list-custom-metrics`.

AWS CLI

Um Ihre benutzerdefinierten Metriken aufzulisten

Das folgende `list-custom-metrics` Beispiel listet alle Ihre benutzerdefinierten Metriken auf.

```
aws iot list-custom-metrics \
  --region us-east-1
```

Ausgabe:

```
{
  "metricNames": [
    "batteryPercentage"
  ]
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Metriken](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [ListCustomMetrics](#) in der AWS CLI Befehlsreferenz.

list-dimensions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-dimensions`.

AWS CLI

Um die Dimensionen für Ihr AWS Konto aufzulisten

Das folgende `list-dimensions` Beispiel listet alle AWS IoT Device Defender-Dimensionen auf, die in Ihrem AWS Konto definiert sind.

```
aws iot list-dimensions
```

Ausgabe:

```
{
  "dimensionNames": [
    "TopicFilterForAuthMessages",
    "TopicFilterForActivityMessages"
  ]
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListDimensions](#) in der AWS CLI Befehlsreferenz.

list-domain-configurations

Das folgende Codebeispiel zeigt, wie man es benutzt `list-domain-configurations`.

AWS CLI

Um Domänenkonfigurationen aufzulisten

Im folgenden `list-domain-configurations` Beispiel werden die Domänenkonfigurationen in Ihrem AWS Konto aufgeführt, die den angegebenen Dienstyp haben.

```
aws iot list-domain-configurations \
  --service-type "DATA"
```

Ausgabe:

```
{
  "domainConfigurations":
  [
    {
      "domainConfigurationName": "additionalDataDomain",
      "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh",
      "serviceType": "DATA"
    },
    {
      "domainConfigurationName": "iot:Jobs",
      "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:Jobs",

```

```
        "serviceType": "JOBS"
      },
      {
        "domainConfigurationName": "iot:Data-ATS",
        "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:Data-ATS",
        "serviceType": "DATA"
      },
      {
        "domainConfigurationName": "iot:CredentialProvider",
        "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:CredentialProvider",
        "serviceType": "CREDENTIAL_PROVIDER"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Configurable Endpoints](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListDomainConfigurations](#) in der AWS CLI Befehlsreferenz.

list-indices

Das folgende Codebeispiel zeigt, wie man es benutzt `list-indices`.

AWS CLI

Um die konfigurierten Suchindizes aufzulisten

Das folgende `list-indices` Beispiel listet alle konfigurierten Suchindizes in Ihrem AWS Konto auf. Wenn Sie die Indizierung von Dingen nicht aktiviert haben, haben Sie möglicherweise keine Indizes.

```
aws iot list-indices
```

Ausgabe:

```
{
  "indexNames": [
    "AWS_Things"
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Managing Thing Indexing](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListIndices](#) in der AWS CLI Befehlsreferenz.

list-job-executions-for-job

Das folgende Codebeispiel zeigt, wie man es benutzt `list-job-executions-for-job`.

AWS CLI

Um die Jobs in Ihrem AWS Konto aufzulisten

Das folgende `list-job-executions-for-job` Beispiel listet alle Auftragsausführungen für einen Job in Ihrem AWS Konto auf, angegeben durch `jobId`.

```
aws iot list-job-executions-for-job \
  --job-id my-ota-job
```

Ausgabe:

```
{
  "executionSummaries": [
    {
      "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/my_thing",
      "jobExecutionSummary": {
        "status": "QUEUED",
        "queuedAt": "2022-03-07T15:58:42.195000-08:00",
        "lastUpdatedAt": "2022-03-07T15:58:42.195000-08:00",
        "executionNumber": 1,
        "retryAttempt": 0
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListJobExecutionsForJob](#) in der AWS CLI Befehlsreferenz.

list-job-executions-for-thing

Das folgende Codebeispiel zeigt, wie man es benutzt `list-job-executions-for-thing`.

AWS CLI

Um die Jobs aufzulisten, die für eine Sache ausgeführt wurden

Das folgende `list-job-executions-for-thing` Beispiel listet alle Jobs auf, die für das genannte Ding ausgeführt wurden `MyRaspberryPi`.

```
aws iot list-job-executions-for-thing \
  --thing-name "MyRaspberryPi"
```

Ausgabe:

```
{
  "executionSummaries": [
    {
      "jobId": "example-job-01",
      "jobExecutionSummary": {
        "status": "QUEUED",
        "queuedAt": 1560787023.636,
        "lastUpdatedAt": 1560787023.636,
        "executionNumber": 1
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListJobExecutionsForThing](#) in der AWS CLI Befehlsreferenz.

list-jobs

Das folgende Codebeispiel zeigt, wie man es benutzt `list-jobs`.

AWS CLI

Um die Jobs in Ihrem AWS Konto aufzulisten

Im folgenden `list-jobs` Beispiel werden alle Jobs in Ihrem AWS Konto aufgelistet, sortiert nach dem Jobstatus.

```
aws iot list-jobs
```

Ausgabe:

```
{
  "jobs": [
    {
      "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
      "jobId": "example-job-01",
      "targetSelection": "SNAPSHOT",
      "status": "IN_PROGRESS",
      "createdAt": 1560787022.733,
      "lastUpdatedAt": 1560787026.294
    }
  ]
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListJobs](#) in der AWS CLI Befehlsreferenz.

list-mitigation-actions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-mitigation-actions`.

AWS CLI

Um alle definierten Minderungsmaßnahmen aufzulisten

Das folgende `list-mitigation-actions` Beispiel listet alle definierten Minderungsmaßnahmen für Ihr AWS Konto und Ihre Region auf. Für jede Aktion werden der Name und ARN das Erstellungsdatum aufgeführt.

```
aws iot list-mitigation-actions
```

Ausgabe:

```
{
  "actionIdentifiers": [
    {
      "actionName": "DeactivateCACertAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/DeactivateCACertAction",
      "creationDate": "2019-12-10T11:12:47.574000-08:00"
    },
    {
      "actionName": "ResetPolicyVersionAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/ResetPolicyVersionAction",
      "creationDate": "2019-12-10T11:11:48.920000-08:00"
    },
    {
      "actionName": "PublishFindingToSNSAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/PublishFindingToSNSAction",
      "creationDate": "2019-12-10T11:10:49.546000-08:00"
    },
    {
      "actionName": "AddThingsToQuarantineGroupAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/AddThingsToQuarantineGroupAction",
      "creationDate": "2019-12-10T11:09:35.999000-08:00"
    },
    {
      "actionName": "UpdateDeviceCertAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/UpdateDeviceCertAction",
      "creationDate": "2019-12-10T11:08:44.263000-08:00"
    },
    {
      "actionName": "SampleMitigationAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/SampleMitigationAction",
      "creationDate": "2019-12-10T11:03:41.840000-08:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [ListMitigationActions \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListMitigationActions](#) in der AWS CLI Befehlsreferenz.

list-mitigations-actions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-mitigations-actions`.

AWS CLI

Um alle definierten Minderungsmaßnahmen aufzulisten

Das folgende `list-mitigations-actions` Beispiel listet alle definierten Minderungsmaßnahmen für Ihr AWS Konto und Ihre Region auf. Für jede Aktion werden der Name und ARN das Erstellungsdatum aufgeführt.

```
aws iot list-mitigation-actions
```

Ausgabe:

```
{
  "actionIdentifiers": [
    {
      "actionName": "DeactivateCACertAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/DeactivateCACertAction",
      "creationDate": "2019-12-10T11:12:47.574000-08:00"
    },
    {
      "actionName": "ResetPolicyVersionAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/ResetPolicyVersionAction",
      "creationDate": "2019-12-10T11:11:48.920000-08:00"
    },
    {
      "actionName": "PublishFindingToSNSAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/PublishFindingToSNSAction",
      "creationDate": "2019-12-10T11:10:49.546000-08:00"
    },
    {
      "actionName": "AddThingsToQuarantineGroupAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/AddThingsToQuarantineGroupAction",

```

```

        "creationDate": "2019-12-10T11:09:35.999000-08:00"
      },
      {
        "actionName": "UpdateDeviceCertAction",
        "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
UpdateDeviceCertAction",
        "creationDate": "2019-12-10T11:08:44.263000-08:00"
      },
      {
        "actionName": "SampleMitigationAction",
        "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
SampleMitigationAction",
        "creationDate": "2019-12-10T11:03:41.840000-08:00"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [ListMitigationActions \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListMitigationActions](#) in der AWS CLI Befehlsreferenz.

list-ota-updates

Das folgende Codebeispiel zeigt, wie man es benutzt `list-ota-updates`.

AWS CLI

Um OTA Updates für das Konto aufzulisten

Das folgende `list-ota-updates` Beispiel listet die verfügbaren OTA Updates auf.

```
aws iot list-ota-updates
```

Ausgabe:

```

{
  "otaUpdates": [
    {
      "otaUpdateId": "itsaupdate",
      "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/
itsaupdate",

```

```

        "creationDate": 1557863215.995
      }
    ]
  }

```

Weitere Informationen finden Sie unter [ListOTAUpdates](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListOtaUpdates](#) unter AWS CLI Befehlsreferenz.

list-outgoing-certificates

Das folgende Codebeispiel zeigt, wie man es benutzt `list-outgoing-certificates`.

AWS CLI

Um Zertifikate aufzulisten, die auf ein anderes AWS Konto übertragen werden

Das folgende `list-outgoing-certificates` Beispiel listet alle Gerätezertifikate auf, die gerade mithilfe des `transfer-certificate` Befehls auf ein anderes AWS Konto übertragen werden.

```
aws iot list-outgoing-certificates
```

Ausgabe:

```

{
  "outgoingCertificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId": "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "transferredTo": "030714055129",
      "transferDate": 1569427780.441,
      "creationDate": 1569363250.557
    }
  ]
}

```

Weitere Informationen finden Sie [ListOutgoingCertificates](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListOutgoingCertificates](#) unter AWS CLI Befehlsreferenz.

list-policies

Das folgende Codebeispiel zeigt, wie man es benutzt `list-policies`.

AWS CLI

Um die in Ihrem AWS Konto definierten Richtlinien aufzulisten

Das folgende `list-policies` Beispiel listet alle in Ihrem AWS Konto definierten Richtlinien auf.

```
aws iot list-policies
```

Ausgabe:

```
{
  "policies": [
    {
      "policyName": "UpdateDeviceCertPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy"
    },
    {
      "policyName": "PlantIoTPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/PlantIoTPolicy"
    },
    {
      "policyName": "MyPiGroup_Core-policy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/MyPiGroup_Core-policy"
    }
  ]
}
```

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListPolicies](#) in der AWS CLI Befehlsreferenz.

list-policy-versions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-policy-versions`.

AWS CLI

Beispiel 1: Um alle Versionen einer Richtlinie anzuzeigen

Das folgende `list-policy-versions` Beispiel listet alle Versionen der angegebenen Richtlinie und ihre Erstellungsdaten auf.

```
aws iot list-policy-versions \  
  --policy-name LightBulbPolicy
```

Ausgabe:

```
{  
  "policyVersions": [  
    {  
      "versionId": "2",  
      "isDefaultVersion": true,  
      "createDate": 1559925941.924  
    },  
    {  
      "versionId": "1",  
      "isDefaultVersion": false,  
      "createDate": 1559925941.924  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [AWS IoT-Richtlinien](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListPolicyVersions](#) in der AWS CLI Befehlsreferenz.

list-principal-things

Das folgende Codebeispiel zeigt, wie man es benutzt `list-principal-things`.

AWS CLI

Um die Dinge aufzulisten, die einem Principal zugeordnet sind

Das folgende `list-principal-things` Beispiel listet die Dinge auf, die mit dem durch einen angegebenen Prinzipal angegebenen Prinzipal verknüpft sind ARN.


```
aws iot list-principal-things \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

Ausgabe:

```
{  
  "things": [  
    "DeskLamp",  
    "TableLamp"  
  ]  
}
```

Weitere Informationen finden Sie [ListPrincipalThings](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListPrincipalThings](#) unter AWS CLI Befehlsreferenz.

list-provisioning-template-versions

Das folgende Codebeispiel zeigt, wie man es benutzt `list-provisioning-template-versions`.

AWS CLI

Um die Versionen der Provisioning-Vorlagen aufzulisten

Im folgenden `list-provisioning-template-versions` Beispiel werden die verfügbaren Versionen der angegebenen Bereitstellungsvorlage aufgeführt.

```
aws iot list-provisioning-template-versions \  
  --template-name "widget-template"
```

Ausgabe:

```
{  
  "versions": [  
    {  
      "versionId": 1,  
      "creationDate": 1574800471.339,  
      "isDefaultVersion": true  
    },  
    {  
      "versionId": 2,
```

```
        "creationDate": 1574801192.317,  
        "isDefaultVersion": false  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [ListProvisioningTemplateVersions](#) in der AWS CLI Befehlsreferenz.

list-provisioning-templates

Das folgende Codebeispiel zeigt, wie man es benutzt `list-provisioning-templates`.

AWS CLI

Um Bereitstellungsvorlagen aufzulisten

Im folgenden `list-provisioning-templates` Beispiel werden alle Bereitstellungsvorlagen in Ihrem AWS Konto aufgeführt.

```
aws iot list-provisioning-templates
```

Ausgabe:

```
{  
  "templates": [  
    {  
      "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/  
widget-template",  
      "templateName": "widget-template",  
      "description": "A provisioning template for widgets",  
      "creationDate": 1574800471.367,  
      "lastModifiedDate": 1574801192.324,  
      "enabled": false  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [ListProvisioningTemplates](#) in der AWS CLI Befehlsreferenz.

list-role-aliases

Das folgende Codebeispiel zeigt, wie man es benutzt `list-role-aliases`.

AWS CLI

Um die AWS IoT-Rollenalias in Ihrem AWS Konto aufzulisten

Das folgende `list-role-aliases` Beispiel listet die AWS IoT-Rollenalias in Ihrem AWS Konto auf.

```
aws iot list-role-aliases
```

Ausgabe:

```
{
  "roleAliases": [
    "ResidentAlias",
    "ElectricianAlias"
  ]
}
```

Weitere Informationen finden Sie [ListRoleAliases](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListRoleAliases](#) unter AWS CLI Befehlsreferenz.

list-scheduled-audits

Das folgende Codebeispiel zeigt, wie man es benutzt `list-scheduled-audits`.

AWS CLI

Um die geplanten Audits für Ihr AWS Konto aufzulisten

Das folgende `list-scheduled-audits` Beispiel listet alle für Ihr AWS Konto geplanten Audits auf.

```
aws iot list-scheduled-audits
```

Ausgabe:

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
      "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSIoTDeviceDefenderDailyAudit",
      "frequency": "DAILY"
    },
    {
      "scheduledAuditName": "AWSDeviceDefenderWeeklyAudit",
      "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSDeviceDefenderWeeklyAudit",
      "frequency": "WEEKLY",
      "dayOfWeek": "SUN"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListScheduledAudits](#) in der AWS CLI Befehlsreferenz.

list-security-profiles-for-target

Das folgende Codebeispiel zeigt, wie man es benutzt `list-security-profiles-for-target`.

AWS CLI

Um die an ein Ziel angehängten Sicherheitsprofile aufzulisten

Das folgende `list-security-profiles-for-target` Beispiel listet die AWS IoT Device Defender-Sicherheitsprofile auf, die an nicht registrierte Geräte angehängt sind.

```
aws iot list-security-profiles-for-target \
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
  unregistered-things"
```

Ausgabe:

```
{
  "securityProfileTargetMappings": [
```

```

    {
      "securityProfileIdentifier": {
        "name": "Testprofile",
        "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
Testprofile"
      },
      "target": {
        "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListSecurityProfilesForTarget](#) in der AWS CLI Befehlsreferenz.

list-security-profiles

Das folgende Codebeispiel zeigt, wie man es benutzt `list-security-profiles`.

AWS CLI

Um die Sicherheitsprofile für Ihr AWS Konto aufzulisten

Das folgende `list-security-profiles` Beispiel listet alle AWS IoT Device Defender-Sicherheitsprofile auf, die in Ihrem AWS Konto definiert sind.

```
aws iot list-security-profiles
```

Ausgabe:

```

{
  "securityProfileIdentifiers": [
    {
      "name": "Testprofile",
      "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListSecurityProfiles](#) in der AWS CLI Befehlsreferenz.

list-streams

Das folgende Codebeispiel zeigt, wie man es benutzt `list-streams`.

AWS CLI

Um die Streams im Konto aufzulisten

Das folgende `list-streams` Beispiel listet alle Streams in Ihrem AWS Konto auf.

```
aws iot list-streams
```

Ausgabe:

```
{
  "streams": [
    {
      "streamId": "stream12345",
      "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
      "streamVersion": 1,
      "description": "This stream is used for Amazon FreeRTOS OTA Update
12345."
    },
    {
      "streamId": "stream54321",
      "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream54321",
      "streamVersion": 1,
      "description": "This stream is used for Amazon FreeRTOS OTA Update
54321."
    }
  ]
}
```

Weitere Informationen finden Sie [ListStreams](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListStreams](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um die mit einer Ressource verknüpften Tags und deren Werte anzuzeigen

Im folgenden `list-tags-for-resource` Beispiel werden die Tags und Werte angezeigt, die der Dinggruppe zugeordnet sind `LightBulbs`.

```
aws iot list-tags-for-resource \  
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Ausgabe:

```
{  
  "tags": [  
    {  
      "Key": "Assembly",  
      "Value": "Fact1NW"  
    },  
    {  
      "Key": "MyTag",  
      "Value": "777"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Tagging Your AWS IoT Resources](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-targets-for-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `list-targets-for-policy`.

AWS CLI

Um die mit einer AWS IoT-Richtlinie verknüpften Prinzipale aufzulisten

Im folgenden `list-targets-for-policy` Beispiel werden die Gerätezertifikate aufgeführt, an die die angegebene Richtlinie angehängt ist.

```
aws iot list-targets-for-policy \  
  --policy-name "PolicyName"
```

```
--policy-name UpdateDeviceCertPolicy
```

Ausgabe:

```
{
  "targets": [
    "arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
    "arn:aws:iot:us-west-2:123456789012:cert/d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"
  ]
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListTargetsForPolicy](#) in der AWS CLI Befehlsreferenz.

list-targets-for-security-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `list-targets-for-security-profile`.

AWS CLI

Um die Ziele aufzulisten, auf die ein Sicherheitsprofil angewendet wird

Das folgende `list-targets-for-security-profile` Beispiel listet die Ziele auf, auf die das angegebene AWS IoT Device Defender-Sicherheitsprofil angewendet `PossibleIssue` wird.

```
aws iot list-targets-for-security-profile \
  --security-profile-name Testprofile
```

Ausgabe:

```
{
  "securityProfileTargets": [
    {
      "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
    },
    {
      "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"
    }
  ]
}
```



```
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListTargetsForSecurityProfile](#) in der AWS CLI Befehlsreferenz.

list-thing-groups-for-thing

Das folgende Codebeispiel zeigt, wie man es benutzt `list-thing-groups-for-thing`.

AWS CLI

Um die Gruppen aufzulisten, zu denen ein Ding gehört

Das folgende `list-thing-groups-for-thing` Beispiel listet die Gruppen auf, zu denen das angegebene Ding gehört.

```
aws iot list-thing-groups-for-thing \  
  --thing-name MyLightBulb
```

Ausgabe:

```
{  
  "thingGroups": [  
    {  
      "groupName": "DeadBulbs",  
      "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/DeadBulbs"  
    },  
    {  
      "groupName": "LightBulbs",  
      "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListThingGroupsForThing](#) in der AWS CLI Befehlsreferenz.

list-thing-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `list-thing-groups`.

AWS CLI

Um die in Ihrem AWS Konto definierten Dinggruppen aufzulisten

Das folgende `describe-thing-group` Beispiel listet alle in Ihrem AWS Konto definierten Dinggruppen auf.

```
aws iot list-thing-groups
```

Ausgabe:

```
{
  "thingGroups": [
    {
      "groupName": "HalogenBulbs",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs"
    },
    {
      "groupName": "LightBulbs",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListThingGroups](#) in der AWS CLI Befehlsreferenz.

list-thing-principals

Das folgende Codebeispiel zeigt, wie man es benutzt `list-thing-principals`.

AWS CLI

Um die mit einer Sache verknüpften Prinzipale aufzulisten

Das folgende `list-thing-principals` Beispiel listet die Prinzipale (X.509-Zertifikate, IAM Benutzer, Gruppen, Rollen, Amazon Cognito Cognito-Identitäten oder föderierte Identitäten) auf, die dem angegebenen Ding zugeordnet sind.

```
aws iot list-thing-principals \
  --thing-name MyRaspberryPi
```

Ausgabe:

```
{
  "principals": [
    "arn:aws:iot:us-west-2:123456789012:cert/33475ac865079a5ffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"
  ]
}
```

Weitere Informationen finden Sie [ListThingPrincipals](#) in der AWS API IoT-Referenz.

- API Einzelheiten finden Sie [ListThingPrincipals](#) unter AWS CLI Befehlsreferenz.

list-thing-types

Das folgende Codebeispiel zeigt, wie man es benutzt `list-thing-types`.

AWS CLI

Um die definierten Dingtypen aufzulisten

Das folgende `list-thing-types` Beispiel zeigt eine Liste der in Ihrem AWS Konto definierten Dingtypen.

```
aws iot list-thing-types
```

Ausgabe:

```
{
  "thingTypes": [
    {
      "thingTypeName": "LightBulb",
      "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
      "thingTypeProperties": {
        "thingTypeDescription": "light bulb type",
        "searchableAttributes": [
          "model",
          "wattage"
        ]
      },
      "thingTypeMetadata": {
```

```
        "deprecated": false,
        "creationDate": 1559772562.498
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListThingTypes](#) in der AWS CLI Befehlsreferenz.

list-things-in-billing-group

Das folgende Codebeispiel zeigt, wie man es benutzt `list-things-in-billing-group`.

AWS CLI

Um die Dinge in einer Abrechnungsgruppe aufzulisten

Im folgenden `list-things-in-billing-group` Beispiel werden die Dinge aufgeführt, die sich in der angegebenen Abrechnungsgruppe befinden.

```
aws iot list-things-in-billing-group \
  --billing-group-name GroupOne
```

Ausgabe:

```
{
  "things": [
    "MyOtherLightBulb",
    "MyLightBulb"
  ]
}
```

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListThingsInBillingGroup](#) in der AWS CLI Befehlsreferenz.

list-things-in-thing-group

Das folgende Codebeispiel zeigt, wie man es benutzt `list-things-in-thing-group`.

AWS CLI

Um die Dinge aufzulisten, die zu einer Gruppe gehören

Das folgende `list-things-in-thing-group` Beispiel listet die Dinge auf, die zu der angegebenen Dinggruppe gehören.

```
aws iot list-things-in-thing-group \  
  --thing-group-name LightBulbs
```

Ausgabe:

```
{  
  "things": [  
    "MyLightBulb"  
  ]  
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListThingsInThingGroup](#) in der AWS CLI Befehlsreferenz.

list-things

Das folgende Codebeispiel zeigt, wie man es benutzt `list-things`.

AWS CLI

Beispiel 1: Um alle Dinge in der Registrierung aufzulisten

Das folgende `list-things` Beispiel listet die Dinge (Geräte) auf, die in der AWS IoT-Registrierung für Ihr AWS Konto definiert sind.

```
aws iot list-things
```

Ausgabe:

```
{  
  "things": [  
    {  
      "thingName": "ThirdBulb",  
      "thingTypeName": "LightBulb",
```

```
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/ThirdBulb",
    "attributes": {
      "model": "123",
      "wattage": "75"
    },
    "version": 2
  },
  {
    "thingName": "MyOtherLightBulb",
    "thingTypeName": "LightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
    "attributes": {
      "model": "123",
      "wattage": "75"
    },
    "version": 3
  },
  {
    "thingName": "MyLightBulb",
    "thingTypeName": "LightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "attributes": {
      "model": "123",
      "wattage": "75"
    },
    "version": 1
  },
  {
    "thingName": "SampleIoTThing",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/SampleIoTThing",
    "attributes": {},
    "version": 1
  }
]
```

Beispiel 2: Um die definierten Dinge aufzulisten, die ein bestimmtes Attribut haben

Im folgenden `list-things` Beispiel wird eine Liste von Dingen angezeigt, für die ein Attribut benannt ist `wattage`.

```
aws iot list-things \  
  --attribute-name wattage
```

Ausgabe:

```
{
  "things": [
    {
      "thingName": "MyLightBulb",
      "thingTypeName": "LightBulb",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
      "attributes": {
        "model": "123",
        "wattage": "75"
      },
      "version": 1
    },
    {
      "thingName": "MyOtherLightBulb",
      "thingTypeName": "LightBulb",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
      "attributes": {
        "model": "123",
        "wattage": "75"
      },
      "version": 3
    }
  ]
}
```

Weitere Informationen finden Sie unter [How to Manage Things with the Registry](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListThings](#) in der AWS CLI Befehlsreferenz.

list-topic-rule-destinations

Das folgende Codebeispiel zeigt, wie man es benutzt `list-topic-rule-destinations`.

AWS CLI

Um die Ziele Ihrer Themenregeln aufzulisten

Im folgenden `list-topic-rule-destinations` Beispiel werden alle Ziele für Themenregeln aufgeführt, die Sie in der aktuellen AWS Region definiert haben.

```
aws iot list-topic-rule-destinations
```

Ausgabe:

```
{
  "destinationSummaries": [
    {
      "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "status": "ENABLED",
      "httpUrlSummary": {
        "confirmationUrl": "https://example.com"
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Themenregelzielen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListTopicRuleDestinations](#) in der AWS CLI Befehlsreferenz.

list-topic-rules

Das folgende Codebeispiel zeigt, wie man es benutzt `list-topic-rules`.

AWS CLI

Um deine Regeln aufzulisten

Das folgende `list-topic-rules` Beispiel listet alle Regeln auf, die Sie definiert haben.

```
aws iot list-topic-rules
```

Ausgabe:

```
{
  "rules": [
    {
      "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyRPiLowMoistureAlertRule",
      "ruleName": "MyRPiLowMoistureAlertRule",
```



```
    "topicPattern": "$aws/things/MyRPi/shadow/update/accepted",
    "createdAt": 1558624363.0,
    "ruleDisabled": false
  },
  {
    "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyPlantPiMoistureAlertRule",
    "ruleName": "MyPlantPiMoistureAlertRule",
    "topicPattern": "$aws/things/MyPlantPi/shadow/update/accepted",
    "createdAt": 1541458459.0,
    "ruleDisabled": false
  }
]
}
```

Weitere Informationen finden Sie unter [Regeln anzeigen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListTopicRules](#) in der AWS CLI Befehlsreferenz.

list-v2-logging-levels

Das folgende Codebeispiel zeigt, wie man es benutzt `list-v2-logging-levels`.

AWS CLI

Um die Protokollierungsebenen aufzulisten

Das folgende `list-v2-logging-levels` Beispiel listet die konfigurierten Protokollierungsebenen auf. Wenn keine Protokollierungsebenen festgelegt wurden, `NotConfigurationException` tritt ein auf, wenn Sie diesen Befehl ausführen.

```
aws iot list-v2-logging-levels
```

Ausgabe:

```
{
  "logTargetConfigurations": [
    {
      "logTarget": {
        "targetType": "DEFAULT"
      },
      "logLevel": "ERROR"
    }
  ]
}
```

```
]
}
```

- API-Einheiten finden Sie unter [ListV2 LoggingLevels](#) in der AWS CLI Befehlsreferenz.

list-violation-events

Das folgende Codebeispiel zeigt die Verwendung `list-violation-events`.

AWS CLI

Um die Verletzungen des Sicherheitsprofils während eines bestimmten Zeitraums aufzulisten

Das folgende `list-violation-events` Beispiel listet Verstöße auf, die zwischen dem 5. Juni 2019 und dem 12. Juni 2019 für alle AWS IoT Device Defender-Sicherheitsprofile für das aktuelle AWS Konto und die AWS Region aufgetreten sind.

```
aws iot list-violation-events \
  --start-time 1559747125 \
  --end-time 1560351925
```

Ausgabe:

```
{
  "violationEvents": [
    {
      "violationId": "174db59167fa474c80a652ad1583fd44",
      "thingName": "iotconsole-1560269126751-1",
      "securityProfileName": "Testprofile",
      "behavior": {
        "name": "Authorization",
        "metric": "aws:num-authorization-failures",
        "criteria": {
          "comparisonOperator": "greater-than",
          "value": {
            "count": 10
          },
          "durationSeconds": 300,
          "consecutiveDatapointsToAlarm": 1,
          "consecutiveDatapointsToClear": 1
        }
      }
    },
  ],
}
```

```
    "metricValue": {
      "count": 0
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560279000.0
  },
  {
    "violationId": "c8a9466a093d3b7b35cd44ca58bdbbeab",
    "thingName": "TvnQoEoU",
    "securityProfileName": "Testprofile",
    "behavior": {
      "name": "CellularBandwidth",
      "metric": "aws:message-byte-size",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 128
        },
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    },
    "metricValue": {
      "count": 110
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560276600.0
  },
  {
    "violationId": "74aa393adea02e6648f3ac362beed55e",
    "thingName": "iotconsole-1560269232412-2",
    "securityProfileName": "Testprofile",
    "behavior": {
      "name": "Authorization",
      "metric": "aws:num-authorization-failures",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 10
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    }
  }
}
```

```

    },
    "metricValue": {
      "count": 0
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560276600.0
  },
  {
    "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
    "thingName": "TvnQoEoU",
    "securityProfileName": "Testprofile",
    "behavior": {
      "name": "Authorization",
      "metric": "aws:num-authorization-failures",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 10
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    },
    "metricValue": {
      "count": 0
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560276600.0
  }
]
}

```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ListViolationEvents](#) in der AWS CLI Befehlsreferenz.

register-ca-certificate

Das folgende Codebeispiel zeigt, wie man es benutzt `register-ca-certificate`.

AWS CLI

Um ein Zertifikat einer Zertifizierungsstelle (CA) zu registrieren

Im folgenden `register-ca-certificate` Beispiel wird ein CA-Zertifikat registriert. Der Befehl stellt das CA-Zertifikat und ein Schlüssilverifizierungszertifikat bereit, das nachweist, dass Sie der Eigentümer des privaten Schlüssels sind, der mit dem CA-Zertifikat verknüpft ist.

```
aws iot register-ca-certificate \  
  --ca-certificate file://rootCA.pem \  
  --verification-cert file://verificationCert.pem
```

Ausgabe:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/  
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",  
  "certificateId":  
  "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467"  
}
```

Weitere Informationen finden Sie unter [R registerCACertificate](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [RegisterCaCertificate](#) unter AWS CLI Befehlsreferenz.

register-certificate

Das folgende Codebeispiel zeigt, wie man es benutzt `register-certificate`.

AWS CLI

Um ein selbstsigniertes Gerätezertifikat zu registrieren

Im folgenden `register-certificate` Beispiel wird das mit dem `rootCA.pem` CA-Zertifikat signierte `deviceCert.pem` Gerätezertifikat registriert. Das CA-Zertifikat muss registriert werden, bevor Sie es zur Registrierung eines selbstsignierten Gerätezertifikats verwenden können. Das selbstsignierte Zertifikat muss mit demselben CA-Zertifikat signiert sein, das Sie an diesen Befehl übergeben.

```
aws iot register-certificate \  
  --certificate-pem file://deviceCert.pem \  
  --ca-certificate-pem file://rootCA.pem
```

Ausgabe:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
  "certificateId":
  "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}
```

Weitere Informationen finden Sie [RegisterCertificate](#) in der AWS API IoT-Referenz.

- API Einzelheiten finden Sie [RegisterCertificate](#) unter AWS CLI Befehlsreferenz.

register-thing

Das folgende Codebeispiel zeigt, wie man es benutzt `register-thing`.

AWS CLI

Um eine Sache zu registrieren

Im folgenden `register-thing` Beispiel wird ein Ding mithilfe einer Bereitstellungsvorlage registriert.

```
aws iot register-thing \
  --template-body '{"Parameters":{"ThingName":
{"Type":"String"},"AWS::IoT::Certificate::Id":{"Type":"String"}}, "Resources":
{"certificate":{"Properties":{"CertificateId":
{"Ref":"AWS::IoT::Certificate::Id"},"Status":"Active"},"Type":"AWS::IoT::Certificate"},"poli
{"Properties":{"PolicyName":"MyIotPolicy"},"Type":"AWS::IoT::Policy"},"thing":
{"OverrideSettings":
{"AttributePayload":"MERGE","ThingGroups":"DO_NOTHING","ThingTypeName":"REPLACE"},"Propertie
{"AttributePayload":{},"ThingGroups":[],"ThingName":
{"Ref":"ThingName"},"ThingTypeName":"VirtualThings"},"Type":"AWS::IoT::Thing"}}}' \
  --parameters '{"ThingName":"Register-thing-
trial-1","AWS::IoT::Certificate::Id":"799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e3
```

Ausgabe:

```
{
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAKGgAwIBAgIUYLk81I35cIppobpw
Hi0J2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLEKGA1UECwxCQW1hem9uIFd1YiBTZXJ2aWNlcyBPPUFtYXpvbi
```

```

5jb20g\nSW5jLiBMPVN1YXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVtMB4XDTIwMDcyMzE2NDUw
\n0VoXDTQ5MTIzMT
IzNTk1OVowHjEcMBoGA1UEAwwTQVdTIE1vVCBDZXJ0aWZpY2F0\nZTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBA071uADhdBajqTmg1MV5\nmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMAckeX8Keb1I0TKzORI
RDXnyE
\n6lV0wjgAsd0ku22rFxex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nnKqiuBJVNOGKTcQ
Hd1RN0rddwwu6kFJLeKDMEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nGxu7u0W3hn6n1RN8qVoka0uW12p53xM7oHVz
Gf+cxKBx1b0hGkp6yCfTskUBm3Sp\n9zLw35kiHXVm4EVpWgNlnk6XcIGIkw8a/iy4pzmVUGAANY1/uU/
zgCjymw
ZT5S30\nbV0CAwEAAANgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR00BBYEF0VtvZ
9Aj2RYFnkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nnAQH/
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IB
AQXCXQcp0tubS5ft0sDMTcP/jNX
\nDHYArxmjpSc2aCdm7WX591TKWyAdxGAvqaDVWqTo0oXI7tZ8w7aIN1Gi5
pXnifx\n3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fe7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4sr0CKynS5wMJUQZ1HQ0nabVwnwK4Y0Mflp
\np9+4susFUR9aT3BT1AcIwqSpzh1Khh4Iz7ND
kRn4amsUT210jg/z0010w+BTHcVQ\nJly8XDu0CWSu04q6SnaBzHmlySIajxuRTP/AdfRouP10Xe
+q1bPOBcvVvF
8o\n-----END CERTIFICATE-----\n",
  "resourceArns": {
    "certificate": "arn:aws:iot:us-
west-2:571032923833:cert/799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e30404b9233c",
    "thing": "arn:aws:iot:us-west-2:571032923833:thing/Register-thing-trial-1"
  }
}

```

Weitere Informationen finden Sie unter [Bereitstellung durch einen vertrauenswürdigen Benutzer](#) im AWS IoT Core Developers Guide.

- API-Einheiten finden Sie [RegisterThing](#) in der AWS CLI Befehlsreferenz.

reject-certificate-transfer

Das folgende Codebeispiel zeigt, wie man es benutzt `reject-certificate-transfer`.

AWS CLI

Um eine Zertifikatsübertragung abzulehnen

Im folgenden `reject-certificate-transfer` Beispiel wird die Übertragung des angegebenen Gerätezertifikats von einem anderen AWS Konto abgelehnt.

```
aws iot reject-certificate-transfer \  
  --certificate-  
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78add5e605d630e05c7fc8
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Übertragen eines Zertifikats auf ein anderes Konto](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [RejectCertificateTransfer](#) in der AWS CLI Befehlsreferenz.

remove-thing-from-billing-group

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-thing-from-billing-group`.

AWS CLI

Um eine Sache aus einer Abrechnungsgruppe zu entfernen

Im folgenden `remove-thing-from-billing-group` Beispiel wird das angegebene Ding aus einer Abrechnungsgruppe entfernt.

```
aws iot remove-thing-from-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyOtherLightBulb
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [RemoveThingFromBillingGroup](#) in der AWS CLI Befehlsreferenz.

remove-thing-from-thing-group

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-thing-from-thing-group`.

AWS CLI

Um ein Ding aus einer Dinggruppe zu entfernen

Im folgenden `remove-thing-from-thing-group` Beispiel wird das angegebene Ding aus einer Dinggruppe entfernt.

```
aws iot remove-thing-from-thing-group \  
  --thing-name bulb7 \  
  --thing-group-name DeadBulbs
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Thing Groups < <https://docs.aws.amazon.com/iot/latest/developerguide/thing-groups.html> > im AWS IoT Developer Guide.

- API-Einheiten finden Sie [RemoveThingFromThingGroup](#) in der AWS CLI Befehlsreferenz.

replace-topic-rule

Das folgende Codebeispiel zeigt, wie man es benutzt `replace-topic-rule`.

AWS CLI

Um die Regeldefinition eines Themas zu aktualisieren

Im folgenden `replace-topic-rule` Beispiel wird die angegebene Regel aktualisiert, sodass eine SNS Warnung gesendet wird, wenn die Bodenfeuchte zu niedrig ist.

```
aws iot replace-topic-rule \  
  --rule-name MyRPiLowMoistureAlertRule \  
  --topic-rule-payload '{"sql": "SELECT * FROM '$aws/things/MyRPi/shadow/  
update/accepted' WHERE state.reported.moisture = 'low'", "description": "Sends  
an alert when soil moisture level readings are too low.", "actions": [{"sns  
": {"targetArn": "arn:aws:sns:us-west-2:123456789012:MyRPiLowMoistureTopic",  
"roleArn": "arn:aws:iam::123456789012:role/service-role/MyRPiLowMoistureTopicRole  
"}, {"messageFormat": "RAW"}]}, {"ruleDisabled": false, "awsIotSqlVersion":  
"2016-03-23"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen einer AWS IoT-Regel](#) im AWS IoT-Entwicklerhandbuch.

- API-Einheiten finden Sie [ReplaceTopicRule](#) in der AWS CLI Befehlsreferenz.

search-index

Das folgende Codebeispiel zeigt, wie man es benutzt `search-index`.

AWS CLI

Um den Dingindex abzufragen

Im folgenden `search-index` Beispiel wird der `AWS_Things` Index nach Dingen abgefragt, die den Typ `LightBulb` haben.

```
aws iot search-index \  
  --index-name "AWS_Things" \  
  --query-string "thingTypeName:LightBulb"
```

Ausgabe:

```
{  
  "things": [  
    {  
      "thingName": "MyLightBulb",  
      "thingId": "40da2e73-c6af-406e-b415-15acae538797",  
      "thingTypeName": "LightBulb",  
      "thingGroupNames": [  
        "LightBulbs",  
        "DeadBulbs"  
      ],  
      "attributes": {  
        "model": "123",  
        "wattage": "75"  
      },  
      "connectivity": {  
        "connected": false  
      }  
    },  
    {  
      "thingName": "ThirdBulb",  
      "thingId": "615c8455-33d5-40e8-95fd-3ee8b24490af",  
      "thingTypeName": "LightBulb",  
      "attributes": {  
        "model": "123",  
        "wattage": "75"  
      },  
    },  
  ],  
}
```

```

        "connectivity": {
            "connected": false
        }
    },
    {
        "thingName": "MyOtherLightBulb",
        "thingId": "6dae0d3f-40c1-476a-80c4-1ed24ba6aa11",
        "thingTypeName": "LightBulb",
        "attributes": {
            "model": "123",
            "wattage": "75"
        },
        "connectivity": {
            "connected": false
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Managing Thing Indexing](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [SearchIndex](#) in der AWS CLI Befehlsreferenz.

set-default-authorizer

Das folgende Codebeispiel zeigt, wie man es benutzt `set-default-authorizer`.

AWS CLI

Um einen Standard-Autorisierer festzulegen

Im folgenden `set-default-authorizer` Beispiel wird der benutzerdefinierte Autorisierer `CustomAuthorizer` als Standardautorisierer festgelegt.

```

aws iot set-default-authorizer \
  --authorizer-name CustomAuthorizer

```

Ausgabe:

```

{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer"
}

```

```
}
```

Weitere Informationen finden Sie [CreateDefaultAuthorizer](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [SetDefaultAuthorizer](#) unter AWS CLI Befehlsreferenz.

set-default-policy-version

Das folgende Codebeispiel zeigt, wie man es benutzt `set-default-policy-version`.

AWS CLI

Um die Standardversion für eine Richtlinie festzulegen

Im folgenden `set-default-policy-version` Beispiel wird die Standardversion 2 für die Richtlinie mit dem Namen auf festgelegt `UpdateDeviceCertPolicy`.

```
aws iot set-default-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SetDefaultPolicyVersion](#) unter AWS CLI Befehlsreferenz.

set-v2-logging-level

Das folgende Codebeispiel zeigt, wie man es benutzt `set-v2-logging-level`.

AWS CLI

Um die Protokollierungsebene für eine Dinggruppe festzulegen

Im folgenden `set-v2-logging-level` Beispiel wird die Protokollierungsebene so eingestellt, dass Warnungen für die angegebene Dinggruppe protokolliert werden.

```
aws iot set-v2-logging-level \  
  --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \  
  --log-level WARN
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [SetV2 LoggingLevel](#) in der AWS CLI Befehlsreferenz.

set-v2-logging-options

Das folgende Codebeispiel zeigt die Verwendung `set-v2-logging-options`.

AWS CLI

Um die Protokolloptionen festzulegen

Im folgenden `set-v2-logging-options` Beispiel wird die standardmäßige Ausführlichkeitsstufe für die Protokollierung auf `ERROR` festgelegt und angegeben, welche für die Protokollierung verwendet werden soll.

```
aws iot set-v2-logging-options \  
  --default-log-level ERROR \  
  --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [SetV2 LoggingOptions](#) in AWS CLI der Befehlsreferenz.

start-audit-mitigation-actions-task

Das folgende Codebeispiel zeigt die Verwendung `start-audit-mitigation-actions-task`.

AWS CLI

Um eine Abhilfemaßnahme auf die Ergebnisse eines Audits anzuwenden

Im folgenden `start-audit-mitigation-actions-task` Beispiel wird die `ResetPolicyVersionAction` Aktion (die die Richtlinie löscht) auf das angegebene Einzelergebnis angewendet.

```
aws iot start-audit-mitigation-actions-task \  
  --task-id "myActionsTaskId" \  
  --target "findingIds=[\"0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464\"]" \  
  --audit-check-to-actions-mapping  
  "IOT_POLICY_OVERLY_PERMISSIVE_CHECK=[\"ResetPolicyVersionAction\"]" \  
  --client-request-token "adhadhahda"
```

Ausgabe:

```
{
```

```
"taskId": "myActionsTaskId"
}
```

Weitere Informationen finden Sie unter [StartAuditMitigationActionsTask \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [StartAuditMitigationActionsTask](#) in der AWS CLI Befehlsreferenz.

start-on-demand-audit-task

Das folgende Codebeispiel zeigt, wie man es benutzt `start-on-demand-audit-task`.

AWS CLI

Um sofort ein Audit zu starten

Das folgende `start-on-demand-audit-task` Beispiel startet ein AWS IoT Device Defender Defender-Audit und führt drei Zertifikatsprüfungen durch.

```
aws iot start-on-demand-audit-task \
  --target-check-
  names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Ausgabe:

```
{
  "taskId": "a3aea009955e501a31b764abe1bebd3d"
}
```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [StartOnDemandAuditTask](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um einen Tag-Schlüssel und einen Wert für eine Ressource anzugeben

Im folgenden `tag-resource` Beispiel wird das Tag mit einem Schlüssel `Assembly` und dem Wert `Fact1NW` auf die Dinggruppe angewendet `LightBulbs`.

```
aws iot tag-resource \  
  --tags Key=Assembly,Value="Fact1NW" \  
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your AWS IoT Resources](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

test-authorization

Das folgende Codebeispiel zeigt, wie man es benutzt `test-authorization`.

AWS CLI

Um Ihre AWS IoT-Richtlinien zu testen

Im folgenden `test-authorization` Beispiel werden die AWS IoT-Richtlinien getestet, die dem angegebenen Prinzipal zugeordnet sind.

```
aws iot test-authorization \  
  --auth-infos actionType=CONNECT,resources=arn:aws:iot:us-  
east-1:123456789012:client/client1 \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/  
aab1068f7f43ac3e3cae4b3a8aa3f308d2a750e6350507962e32c1eb465d9775
```

Ausgabe:

```
{  
  "authResults": [  
    {  
      "authInfo": {  
        "actionType": "CONNECT",  
        "resources": [  
          "arn:aws:iot:us-east-1:123456789012:client/client1"  
        ]  
      }  
    }  
  ]  
}
```

```

    },
    "allowed": {
      "policies": [
        {
          "policyName": "TestPolicyAllowed",
          "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyAllowed"
        }
      ]
    },
    "denied": {
      "implicitDeny": {
        "policies": [
          {
            "policyName": "TestPolicyDenied",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyDenied"
          }
        ]
      },
      "explicitDeny": {
        "policies": [
          {
            "policyName": "TestPolicyExplicitDenied",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TestPolicyExplicitDenied"
          }
        ]
      }
    },
    "authDecision": "IMPLICIT_DENY",
    "missingContextValues": []
  }
]
}

```

Weitere Informationen finden Sie [TestAuthorization](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [TestAuthorization](#) unter AWS CLI Befehlsreferenz.

test-invoke-authorizer

Das folgende Codebeispiel zeigt, wie man es benutzt `test-invoke-authorizer`.

AWS CLI

Um Ihren benutzerdefinierten Authorizer zu testen

Im folgenden `test-invoke-authorizer` Beispiel wird Ihr benutzerdefinierter Authorizer getestet.

```
aws iot test-invoke-authorizer \
  --authorizer-name IoTAuthorizer \
  --token allow \
  --token-signature "mE0GvaHqy9nER/
FdgtJX5LXYEJ3b3vE7t1gEszc0TKGgLKWXtnPkb2AbKn0AZ81GyoN5dVtWDWVmr25m7+
+zjbYIMk2TBvyGXh0mvKFBPkdgyA43KL6SiZy0cTqLPMcQDsP7VX2rXr7CTowCxSNKphGXdqE0/
I5dQ+J06KUaHwCmupt0/MejKtaNwiia064j6wpr0AUwG5S1IYFuRd0X
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUShEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcAjxzGL02ExvyCUNctCV9dYLRRGJj"
```

Ausgabe:

```
{
  "isAuthenticated": true,
  "principalId": "principalId",
  "policyDocuments": [
    [{"Version": "2012-10-17", "Statement":
[{"Action": "iot:Publish", "Effect": "Allow", "Resource": "arn:aws:iot:us-
west-2:123456789012:topic/customauthtesting"}]}]
  ],
  "refreshAfterInSeconds": 600,
  "disconnectAfterInSeconds": 3600
}
```

Weitere Informationen finden Sie [TestInvokeAuthorizer](#) in der AWS API IoT-Referenz.

- API Einzelheiten finden Sie [TestInvokeAuthorizer](#) unter AWS CLI Befehlsreferenz.

transfer-certificate

Das folgende Codebeispiel zeigt, wie man es benutzt `transfer-certificate`.

AWS CLI

Um ein Gerätezertifikat auf ein anderes AWS Konto zu übertragen

Im folgenden `transfer-certificate` Beispiel wird ein Gerätezertifikat auf ein anderes AWS Konto übertragen. Das Zertifikat und das AWS Konto werden anhand der ID identifiziert.

```
aws iot transfer-certificate \  
  --certificate-  
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \  
  --target-aws-account 030714055129
```

Ausgabe:

```
{  
  "transferredCertificateArn": "arn:aws:iot:us-  
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"  
}
```

Weitere Informationen finden Sie unter [Übertragen eines Zertifikats auf ein anderes Konto](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [TransferCertificate](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um einen Tag-Schlüssel aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden das Tag `MyTag` und sein Wert aus der Dinggruppe entfernt `LightBulbs`.

```
command
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your AWS IoT Resources](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-account-audit-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `update-account-audit-configuration`.

AWS CLI

Beispiel 1: So aktivieren Sie SNS Amazon-Benachrichtigungen für Audit-Benachrichtigungen

Das folgende `update-account-audit-configuration` Beispiel aktiviert SNS Amazon-Benachrichtigungen für AWS IoT Device Defender Defender-Audit-Benachrichtigungen, wobei ein Ziel und die Rolle angegeben werden, die zum Schreiben in dieses Ziel verwendet wird.

```
aws iot update-account-audit-configuration \  
  --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-  
west-2:123456789012:ddauids\",roleArn=\"arn:aws:iam::123456789012:role/service-  
role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einen Audit-Check zu aktivieren

Das folgende `update-account-audit-configuration` Beispiel aktiviert die AWS IoT Device Defender Defender-Auditprüfung mit dem Namen `AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`. Sie können eine Audit-Prüfung nicht deaktivieren, wenn sie Teil der `targetCheckNames` für ein oder mehrere geplante Audits für das AWS Konto ist.

```
aws iot update-account-audit-configuration \  
  --audit-check-configurations  
  "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateAccountAuditConfiguration](#) in der AWS CLI Befehlsreferenz.

update-audit-suppression

Das folgende Codebeispiel zeigt, wie man es benutzt `update-audit-suppression`.

AWS CLI

Um ein Audit zu aktualisieren, bei dem eine Unterdrückung festgestellt wurde

Im folgenden `update-audit-suppression` Beispiel wird das Ablaufdatum einer Prüfungserkennung auf den 21.09.2020 aktualisiert.

```
aws iot update-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --resource-identifier deviceCertificateId=c7691e<shortened> \  
  --no-suppress-indefinitely \  
  --expiration-date 2020-09-21
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Audit finding suppressions](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateAuditSuppression](#) in der AWS CLI Befehlsreferenz.

update-authorizer

Das folgende Codebeispiel zeigt die Verwendung `update-authorizer`.

AWS CLI

Um einen benutzerdefinierten Autorisierer zu aktualisieren

Das folgende `update-authorizer` Beispiel zeigt den Status von `zweiCustomAuthorizer2`.

INACTIVE

```
aws iot update-authorizer \  
  --authorizer-name CustomAuthorizer2 \  
  --status INACTIVE
```

Ausgabe:

```
{  
  "authorizerName": "CustomAuthorizer2",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer2"  
}
```

Weitere Informationen finden Sie [UpdateAuthorizer](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [UpdateAuthorizer](#) unter AWS CLI Befehlsreferenz.

update-billing-group

Das folgende Codebeispiel zeigt die Verwendung `update-billing-group`.

AWS CLI

Um Informationen zu einer Abrechnungsgruppe zu aktualisieren

Im folgenden `update-billing-group` Beispiel wird die Beschreibung für die angegebene Abrechnungsgruppe aktualisiert.

```
aws iot update-billing-group \
  --billing-group-name GroupOne \
  --billing-group-properties "billingGroupDescription=\"Primary bulb billing group\""
```

Ausgabe:

```
{
  "version": 2
}
```

Weitere Informationen finden Sie unter [Billing Groups](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateBillingGroup](#) in der AWS CLI Befehlsreferenz.

update-ca-certificate

Das folgende Codebeispiel zeigt die Verwendung `update-ca-certificate`.

AWS CLI

Um ein Zertifikat einer Zertifizierungsstelle (CA) zu aktualisieren

Im folgenden `update-ca-certificate` Beispiel wird dem angegebenen CA-Zertifikat der ACTIVE Status zugewiesen.

```
aws iot update-ca-certificate \
  --certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467 \
```

```
--new-status ACTIVE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [UpdateCACertificate](#) in der AWS APIIoT-Referenz.

- APIEinzelheiten finden Sie [UpdateCaCertificate](#) unter AWS CLI Befehlsreferenz.

update-certificate

Das folgende Codebeispiel zeigt die Verwendung `update-certificate`.

AWS CLI

Um ein Gerätezertifikat zu aktualisieren

Im folgenden `update-certificate` Beispiel wird das angegebene Gerätezertifikat auf INACTIVE Status gesetzt.

```
aws iot update-certificate \  
  --certificate-  
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \  
  --new-status INACTIVE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UpdateCertificate](#) in der AWS APIIoT-Referenz.

- APIEinzelheiten finden Sie [UpdateCertificate](#) unter AWS CLI Befehlsreferenz.

update-custom-metric

Das folgende Codebeispiel zeigt die Verwendung `update-custom-metric`.

AWS CLI

Um eine benutzerdefinierte Metrik zu aktualisieren

Im folgenden `update-custom-metric` Beispiel wird eine benutzerdefinierte Metrik aktualisiert, sodass sie über eine neue verfügbare `display-name`.

```
aws iot update-custom-metric \  

```

```
--metric-name batteryPercentage \  
--display-name 'remaining battery percentage on device' \  
--region us-east-1
```

Ausgabe:

```
{  
  "metricName": "batteryPercentage",  
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/  
batteryPercentage",  
  "metricType": "number",  
  "displayName": "remaining battery percentage on device",  
  "creationDate": "2020-11-17T23:01:35.110000-08:00",  
  "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Metriken](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [UpdateCustomMetric](#) in der AWS CLI Befehlsreferenz.

update-dimension

Das folgende Codebeispiel zeigt die Verwendung `update-dimension`.

AWS CLI

Um eine Dimension zu aktualisieren

Im folgenden `update-dimension` Beispiel wird eine Dimension aktualisiert.

```
aws iot update-dimension \  
  --name TopicFilterForAuthMessages \  
  --string-values device/${iot:ClientId}/auth
```

Ausgabe:

```
{  
  "name": "TopicFilterForAuthMessages",  
  "lastModifiedDate": 1585866222.317,  
  "stringValues": [  
    "device/${iot:ClientId}/auth"  
  ]  
}
```

```
  ],
  "creationDate": 1585854500.474,
  "type": "TOPIC_FILTER",
  "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/
TopicFilterForAuthMessages"
}
```

Weitere Informationen finden Sie unter [Umfangsmetriken in Sicherheitsprofilen mithilfe von Dimensionen](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [UpdateDimension](#) in der AWS CLI Befehlsreferenz.

update-domain-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-domain-configuration`.

AWS CLI

Um eine Domänenkonfiguration zu aktualisieren

Das folgende `update-domain-configuration` Beispiel deaktiviert die angegebene Domänenkonfiguration.

```
aws iot update-domain-configuration \
  --domain-configuration-name "additionalDataDomain" \
  --domain-configuration-status "DISABLED"
```

Ausgabe:

```
{
  "domainConfigurationName": "additionalDataDomain",
  "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

Weitere Informationen finden Sie unter [Configurable Endpoints](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateDomainConfiguration](#) in der AWS CLI Befehlsreferenz.

update-dynamic-thing-group

Das folgende Codebeispiel zeigt die Verwendung `update-dynamic-thing-group`.

AWS CLI

Um eine dynamische Dinggruppe zu aktualisieren

Im folgenden `update-dynamic-thing-group` Beispiel wird die angegebene dynamische Dinggruppe aktualisiert. Es enthält eine Beschreibung und aktualisiert die Abfragezeichenfolge, um die Kriterien für die Gruppenzugehörigkeit zu ändern.

```
aws iot update-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm" \  
  --thing-group-properties "thingGroupDescription=\"This thing group contains \  
rooms warmer than 65F.\" \" \  
  --query-string "attributes.temperature>65"
```

Ausgabe:

```
{  
  "version": 2  
}
```

Weitere Informationen finden Sie unter [Dynamische Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateDynamicThingGroup](#) in der AWS CLI Befehlsreferenz.

update-event-configurations

Das folgende Codebeispiel zeigt die Verwendung `update-event-configurations`.

AWS CLI

Um zu zeigen, welche Ereignistypen veröffentlicht werden

Im folgenden `update-event-configurations` Beispiel wird die Konfiguration aktualisiert, sodass Meldungen aktiviert werden, wenn das CA-Zertifikat hinzugefügt, aktualisiert oder gelöscht wird.

```
aws iot update-event-configurations \  
  --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ereignismeldungen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateEventConfigurations](#) in der AWS CLI Befehlsreferenz.

update-indexing-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-indexing-configuration`.

AWS CLI

Um die Indizierung von Dingen zu aktivieren

Im folgenden `update-indexing-configuration` Beispiel wird die Dingindizierung aktiviert, sodass die Suche nach Registrierungsdaten, Shadow-Daten und dem Status der Ding-Konnektivität mithilfe des `AWS_Things-Index` unterstützt wird.

```
aws iot update-indexing-configuration
  --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Thing Indexing](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateIndexingConfiguration](#) in der AWS CLI Befehlsreferenz.

update-job

Das folgende Codebeispiel zeigt die Verwendung `update-job`.

AWS CLI

Um den detaillierten Status für einen Job abzurufen

Im folgenden `update-job` Beispiel wird der detaillierte Status für den Job abgerufen, dessen ID lautet `example-job-01`.

```
aws iot describe-job \
  --job-id "example-job-01"
```

Ausgabe:

```
{
```

```
"job": {
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
  "jobId": "example-job-01",
  "targetSelection": "SNAPSHOT",
  "status": "IN_PROGRESS",
  "targets": [
    "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"
  ],
  "description": "example job test",
  "presignedUrlConfig": {},
  "jobExecutionsRolloutConfig": {},
  "createdAt": 1560787022.733,
  "lastUpdatedAt": 1560787026.294,
  "jobProcessDetails": {
    "numberOfCanceledThings": 0,
    "numberOfSucceededThings": 0,
    "numberOfFailedThings": 0,
    "numberOfRejectedThings": 0,
    "numberOfQueuedThings": 1,
    "numberOfInProgressThings": 0,
    "numberOfRemovedThings": 0,
    "numberOfTimedOutThings": 0
  },
  "timeoutConfig": {}
}
}
```

Weitere Informationen finden Sie unter [Jobs erstellen und verwalten \(CLI\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateJob](#) in der AWS CLI Befehlsreferenz.

update-mitigation-action

Das folgende Codebeispiel zeigt die Verwendung `update-mitigation-action`.

AWS CLI

Um eine Abhilfemaßnahme zu aktualisieren

Im folgenden `update-mitigation-action` Beispiel wird die angegebene Schadensbegrenzungsaktion mit dem Namen

`aktualisiertAddThingsToQuarantineGroupAction`, der Name der Dinggruppe geändert

und auf festgelegt `overrideDynamicGroups`. `false` Sie können Ihre Änderungen mit dem `describe-mitigation-action` Befehl überprüfen.

```
aws iot update-mitigation-action \
  --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",
  \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":
  [\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"
```

Ausgabe:

```
{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
  AddThingsToQuarantineGroupAction",
  "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"
}
```

Weitere Informationen finden Sie unter [UpdateMitigationAction \(Mitigation Action Commands\)](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateMitigationAction](#) in der AWS CLI Befehlsreferenz.

update-provisioning-template

Das folgende Codebeispiel zeigt die Verwendung `update-provisioning-template`.

AWS CLI

Um eine Bereitstellungsvorlage zu aktualisieren

Im folgenden `update-provisioning-template` Beispiel werden die Beschreibung und der Rollen-ARN für die angegebene Bereitstellungsvorlage geändert und die Vorlage aktiviert.

```
aws iot update-provisioning-template \
  --template-name widget-template \
  --enabled \
  --description "An updated provisioning template for widgets" \
  --provisioning-role-arn arn:aws:iam::504350838278:role/Provision_role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS IoT Secure Tunneling](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [UpdateProvisioningTemplate](#) in der AWS CLI Befehlsreferenz.

update-role-alias

Das folgende Codebeispiel zeigt die Verwendung `update-role-alias`.

AWS CLI

Um einen Rollenalias zu aktualisieren

Im folgenden `update-role-alias` Beispiel wird der `LightBulbRole` Rollenalias aktualisiert.

```
aws iot update-role-alias \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Ausgabe:

```
{  
  "roleAlias": "LightBulbRole",  
  "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"  
}
```

Weitere Informationen finden Sie [UpdateRoleAlias](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [UpdateRoleAlias](#) unter AWS CLI Befehlsreferenz.

update-scheduled-audit

Das folgende Codebeispiel zeigt die Verwendung `update-scheduled-audit`.

AWS CLI

Um eine geplante Auditdefinition zu aktualisieren

Im folgenden `update-scheduled-audit` Beispiel werden die Namen der Zielprüfungen für ein geplantes AWS IoT Device Defender Defender-Audit geändert.

```
aws iot update-scheduled-audit \  
  --scheduled-audit-name WednesdayCertCheck \  
  --target-check-  
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Ausgabe:

```
{
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

Weitere Informationen finden Sie unter [Audit Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateScheduledAudit](#) in der AWS CLI Befehlsreferenz.

update-security-profile

Das folgende Codebeispiel zeigt die Verwendung `update-security-profile`.

AWS CLI

Um ein Sicherheitsprofil zu ändern

Im folgenden `update-security-profile` Beispiel werden sowohl die Beschreibung als auch das Verhalten für ein AWS IoT Device Defender-Sicherheitsprofil aktualisiert.

```
aws iot update-security-profile \
  --security-profile-name PossibleIssue \
  --security-profile-description "Check to see if authorization fails 12 times in
5 minutes or if cellular bandwidth exceeds 128" \
  --behaviors "[{\\"name\\":\\"CellularBandwidth\\",\\"metric\\":\\"aws:message-byte-size
\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":128},
\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}},{\\"name
\\":\\"Authorization\\",\\"metric\\":\\"aws:num-authorization-failures\\",\\"criteria\\":
{\\"comparisonOperator\\":\\"less-than\\",\\"value\\":{\\"count\\":12},\\"durationSeconds
\\":300,\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}}]"
```

Ausgabe:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
  "securityProfileDescription": "check to see if authorization fails 12 times in 5
minutes or if cellular bandwidth exceeds 128",
```

```
"behaviors": [
  {
    "name": "CellularBandwidth",
    "metric": "aws:message-byte-size",
    "criteria": {
      "comparisonOperator": "greater-than",
      "value": {
        "count": 128
      },
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1
    }
  },
  {
    "name": "Authorization",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "comparisonOperator": "less-than",
      "value": {
        "count": 12
      },
      "durationSeconds": 300,
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1
    }
  }
],
"version": 2,
"creationDate": 1560278102.528,
"lastModifiedDate": 1560352711.207
}
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateSecurityProfile](#) in der AWS CLI Befehlsreferenz.

update-stream

Das folgende Codebeispiel zeigt die Verwendung `update-stream`.

AWS CLI

Um einen Stream zu aktualisieren

Das folgende `update-stream` Beispiel aktualisiert einen vorhandenen Stream. Die Stream-Version wird um eins erhöht.

```
aws iot update-stream \  
  --cli-input-json file://update-stream.json
```

Inhalt von `update-stream.json`:

```
{  
  "streamId": "stream12345",  
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
  "files": [  
    {  
      "fileId": 123,  
      "s3Location": {  
        "bucket": "codesign-ota-bucket",  
        "key": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"  
      }  
    }  
  ]  
  "roleArn": "arn:aws:iam:us-west-2:123456789012:role/service-role/  
my_ota_stream_role"  
}
```

Ausgabe:

```
{  
  "streamId": "stream12345",  
  "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",  
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
  "streamVersion": 2  
}
```

Weitere Informationen finden Sie [UpdateStream](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [UpdateStream](#) unter AWS CLI Befehlsreferenz.

update-thing-group

Das folgende Codebeispiel zeigt die Verwendung `update-thing-group`.

AWS CLI

Um die Definition für eine Dinggruppe zu aktualisieren

Im folgenden `update-thing-group` Beispiel wird die Definition für die angegebene Dinggruppe aktualisiert, wobei die Beschreibung und zwei Attribute geändert werden.

```
aws iot update-thing-group \  
  --thing-group-name HalogenBulbs \  
  --thing-group-properties "thingGroupDescription=\"Halogen bulb group\  
attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

Ausgabe:

```
{  
  "version": 2  
}
```

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateThingGroup](#) in der AWS CLI Befehlsreferenz.

update-thing-groups-for-thing

Das folgende Codebeispiel zeigt die Verwendung `update-thing-groups-for-thing`.

AWS CLI

Um die Gruppen zu ändern, zu denen ein Ding gehört

Im folgenden `update-thing-groups-for-thing` Beispiel wird das Objekt mit dem Namen `MyLightBulb` aus der Gruppe mit dem Namen `DeadBulbs` entfernt und der `replaceableItems` gleichzeitig benannten Gruppe hinzugefügt.

```
aws iot update-thing-groups-for-thing \  
  --thing-name MyLightBulb \  
  --thing-groups-to-add "replaceableItems" \  
  --thing-groups-to-remove "DeadBulbs"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dinggruppen](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateThingGroupsForThing](#) in der AWS CLI Befehlsreferenz.

update-thing

Das folgende Codebeispiel zeigt die Verwendung `update-thing`.

AWS CLI

Um ein Ding einem Dingtyp zuzuordnen

Das folgende `update-thing` Beispiel ordnet ein Ding in der AWS IoT-Registrierung einem Dingtyp zu. Wenn Sie die Zuordnung vornehmen, geben Sie Werte für die Attribute an, die durch den Dingtyp definiert sind.

```
aws iot update-thing \  
  --thing-name "MyOtherLightBulb" \  
  --thing-type-name "LightBulb" \  
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

Dieser Befehl erzeugt keine Ausgabe. Verwenden Sie den `describe-thing` Befehl, um das Ergebnis zu sehen.

Weitere Informationen finden Sie unter [Thing Types](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateThing](#) in der AWS CLI Befehlsreferenz.

update-topic-rule-destination

Das folgende Codebeispiel zeigt die Verwendung `update-topic-rule-destination`.

AWS CLI

Beispiel 1: Um ein Ziel für Themenregeln zu aktivieren

Das folgende `update-topic-rule-destination` Beispiel aktiviert den Datenverkehr zu einem Ziel für Themenregeln.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --topic-rule-name "MyRule"
```

```
--status ENABLED
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren eines Ziels für Themenregeln](#) im AWS IoT Developer Guide.

Beispiel 2: So deaktivieren Sie das Ziel einer Themenregel

Im folgenden `update-topic-rule-destination` Beispiel wird der Datenverkehr zu einem Ziel für Themenregeln deaktiviert.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status DISABLED
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deaktivieren eines Ziels für Themenregeln](#) im AWS IoT Developer Guide.

Beispiel 3: Um eine neue Bestätigungsnachricht zu senden

Im folgenden `update-topic-rule-destination` Beispiel wird eine neue Bestätigungsnachricht für ein Ziel für eine Themenregel gesendet.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status IN_PROGRESS
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Senden einer neuen Bestätigungsnachricht](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateTopicRuleDestination](#) in der AWS CLI Befehlsreferenz.

validate-security-profile-behaviors

Das folgende Codebeispiel zeigt die Verwendung `validate-security-profile-behaviors`.

AWS CLI

Beispiel 1: Um die Verhaltensparameter für ein Sicherheitsprofil zu überprüfen

Das folgende `validate-security-profile-behaviors` Beispiel validiert wohlgeformte und korrekte Verhaltensmuster für ein AWS IoT Device Defender-Sicherheitsprofil.

```
aws iot validate-security-profile-behaviors \
  --behaviors "[{"name":"CellularBandwidth","metric":"aws:message-byte-size",
  "criteria":{"comparisonOperator":"greater-than","value":{"count":128},
  "consecutiveDatapointsToAlarm":1,"consecutiveDatapointsToClear":1}},{"name":
  "Authorization","metric":"aws:num-authorization-failures","criteria":
  {"comparisonOperator":"greater-than","value":{"count":12},"durationSeconds":
  300,"consecutiveDatapointsToAlarm":1,"consecutiveDatapointsToClear":1}]]"
```

Ausgabe:

```
{
  "valid": true,
  "validationErrors": []
}
```

Beispiel 2: Um falsche Verhaltensparameter für ein Sicherheitsprofil zu validieren

Das folgende `validate-security-profile-behaviors` Beispiel validiert eine Reihe von Verhaltensweisen, die einen Fehler für ein AWS IoT Device Defender-Sicherheitsprofil enthalten.

```
aws iot validate-security-profile-behaviors \
  --behaviors "[{"name":"CellularBandwidth","metric":"aws:message-byte-size",
  "criteria":{"comparisonOperator":"greater-than","value":{"count":128},
  "consecutiveDatapointsToAlarm":1,"consecutiveDatapointsToClear":1}},{"name":
  "Authorization","metric":"aws:num-authorization-failures","criteria":
  {"comparisonOperator":"greater-than","value":{"count":12},"durationSeconds":
  300,"consecutiveDatapointsToAlarm":100000,"consecutiveDatapointsToClear":
  1}]]"
```

Ausgabe:

```
{
  "valid": false,
  "validationErrors": [
    {
```

```
        "errorMessage": "Behavior Authorization is malformed.  
consecutiveDatapointsToAlarm 100000 should be in range[1,10]"  
      }  
    ]  
  }
```

Weitere Informationen finden Sie unter [Detect Commands](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [ValidateSecurityProfileBehaviors](#) in der AWS CLI Befehlsreferenz.

AWS IoT 1-Click Gerätebeispiele mit AWS CLI

In den folgenden Codebeispielen wird gezeigt, wie Sie mithilfe von AWS Command Line Interface With AWS IoT 1-Click Devices Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

claim-devices-by-claim-code

Das folgende Codebeispiel zeigt die Verwendung `claim-devices-by-claim-code`.

AWS CLI

Um ein oder mehrere AWS IoT 1-Click 1-Click-Geräte mit einem Einlösungscode zu beanspruchen

Im folgenden `claim-devices-by-claim-code` Beispiel wird das angegebene AWS IoT 1-Click 1-Click-Gerät mithilfe eines Einlösungscode (anstelle einer Geräte-ID) beansprucht.

```
aws iot1click-devices claim-devices-by-claim-code \  
  --claim-code C-123EXAMPLE
```

Ausgabe:

```
{
  "Total": 9
  "ClaimCode": "C-123EXAMPLE"
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ClaimDevicesByClaimCode](#) in der AWS CLI Befehlsreferenz.

describe-device

Das folgende Codebeispiel zeigt die Verwendung `describe-device`.

AWS CLI

Um ein Gerät zu beschreiben

Das folgende `describe-device` Beispiel beschreibt das angegebene Gerät.

```
aws iot1click-devices describe-device \
  --device-id G030PM0123456789
```

Ausgabe:

```
{
  "DeviceDescription": {
    "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",
    "Attributes": {
      "projectRegion": "us-west-2",
      "projectName": "AnytownDumpsters",
      "placementName": "customer217",
      "deviceTemplateName": "empty-dumpster-request"
    },
    "DeviceId": "G030PM0123456789",
    "Enabled": false,
    "RemainingLife": 99.9,
    "Type": "button",
    "Tags": {}
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [DescribeDevice](#) in der AWS CLI Befehlsreferenz.

finalize-device-claim

Das folgende Codebeispiel zeigt die Verwendung `finalize-device-claim`.

AWS CLI

So schließen Sie eine Reklamationsanfrage für ein AWS IoT 1-Click 1-Click-Gerät mithilfe einer Geräte-ID ab

Im folgenden `finalize-device-claim` Beispiel wird eine Anspruchsanforderung für das angegebene AWS IoT 1-Click 1-Click-Gerät mithilfe einer Geräte-ID (anstelle eines Einlöscodes) abgeschlossen.

```
aws iot1click-devices finalize-device-claim \  
  --device-id G030PM0123456789
```

Ausgabe:

```
{  
  "State": "CLAIMED"  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [FinalizeDeviceClaim](#) in der AWS CLI Befehlsreferenz.

get-device-methods

Das folgende Codebeispiel zeigt die Verwendung `get-device-methods`.

AWS CLI

Um die verfügbaren Methoden für ein Gerät aufzulisten

Das folgende `get-device-methods` Beispiel listet die verfügbaren Methoden für ein Gerät auf.

```
aws iot1click-devices get-device-methods \  
  --device-id G030PM0123456789
```

Ausgabe:

```
{  
  "DeviceMethods": [  
    {  
      "MethodName": "getDeviceHealthParameters"  
    },  
    {  
      "MethodName": "setDeviceHealthMonitorCallback"  
    },  
    {  
      "MethodName": "getDeviceHealthMonitorCallback"  
    },  
    {  
      "MethodName": "setOnClickCallback"  
    },  
    {  
      "MethodName": "getOnClickCallback"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [GetDeviceMethods](#) in der AWS CLI Befehlsreferenz.

initiate-device-claim

Das folgende Codebeispiel zeigt die Verwendung `initiate-device-claim`.

AWS CLI

Um mithilfe einer Geräte-ID einen Anspruch für ein AWS IoT 1-Click-Gerät geltend zu machen

Im folgenden `initiate-device-claim` Beispiel wird mithilfe einer Geräte-ID (anstelle eines Einlöscodes) eine Anspruchsanforderung für das angegebene AWS IoT 1-Click-Gerät initiiert.


```
aws iot1click-devices initiate-device-claim \  
  --device-id G030PM0123456789
```

Ausgabe:

```
{  
  "State": "CLAIM_INITIATED"  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [InitiateDeviceClaim](#) in der AWS CLI Befehlsreferenz.

invoke-device-method

Das folgende Codebeispiel zeigt die Verwendung `invoke-device-method`.

AWS CLI

Um eine Gerätemethode auf einem Gerät aufzurufen

Im folgenden `invoke-device-method` Beispiel wird die angegebene Methode auf einem Gerät aufgerufen.

```
aws iot1click-devices invoke-device-method \  
  --cli-input-json file://invoke-device-method.json
```

Inhalt von `invoke-device-method.json`:

```
{  
  "DeviceId": "G030PM0123456789",  
  "DeviceMethod": {  
    "DeviceType": "device",  
    "MethodName": "getDeviceHealthParameters"  
  }  
}
```

Ausgabe:

```
{
```

```
"DeviceMethodResponse": "{\\"remainingLife\\": 99.8}"
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [InvokeDeviceMethod](#) in der AWS CLI Befehlsreferenz.

list-device-events

Das folgende Codebeispiel zeigt die Verwendung `list-device-events`.

AWS CLI

Um die Ereignisse eines Geräts für einen bestimmten Zeitraum aufzulisten

Das folgende `list-device-events` Beispiel listet die Ereignisse des angegebenen Geräts für den angegebenen Zeitraum auf.

```
aws iot1click-devices list-device-events \
  --device-id G030PM0123456789 \
  --from-time-stamp 2019-07-17T15:45:12.880Z --to-time-
stamp 2019-07-19T15:45:12.880Z
```

Ausgabe:

```
{
  "Events": [
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\\"clickType\\": \\"SINGLE\\",
\\"reportedTime\\": \\"2019-07-18T23:47:55.015Z\\", \\"certificateId\\":
\\"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\\",
\\"remainingLife\\": 99.85000000000001, \\"testMode\\": false}"
    },
    {
      "Device": {
        "Attributes": {},
```

```

        "DeviceId": "G030PM0123456789",
        "Type": "button"
    },
    "StdEvent": "{\\"clickType\\": \\"DOUBLE\\",
\\"reportedTime\\": \\"2019-07-19T00:14:41.353Z\\", \\"certificateId\\":
\\"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\\",
\\"remainingLife\\": 99.8, \\"testMode\\": false}"
    }
]
}

```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ListDeviceEvents](#) in der AWS CLI Befehlsreferenz.

list-devices

Das folgende Codebeispiel zeigt die Verwendung `list-devices`.

AWS CLI

Um die Geräte eines bestimmten Typs aufzulisten

Das folgende `list-devices` Beispiel listet die Geräte eines bestimmten Typs auf.

```

aws iot1click-devices list-devices \
  --device-type button

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Ausgabe:

```

{
  "Devices": [
    {
      "remainingLife": 99.9,
      "attributes": {
        "arn": "arn:aws:iot1click:us-west-2:123456789012:devices/
G030PM0123456789",
        "type": "button",
        "deviceId": "G030PM0123456789",
        "enabled": false
      }
    }
  ]
}

```

```
}  
  }  
] }  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ListDevices](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für ein Gerät aufzulisten

Im folgenden `list-tags-for-resource` Beispiel werden die Tags für das angegebene Gerät aufgelistet.

```
aws iot1click-devices list-tags-for-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/  
G030PM0123456789"
```

Ausgabe:

```
{  
  "Tags": {  
    "Driver Phone": "123-555-0199",  
    "Driver": "Jorge Souza"  
  }  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer AWS Gerätesource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel werden der angegebenen Ressource zwei Tags hinzugefügt.

```
aws iot1click-devices tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

Inhalt von `devices-tag-resource.json`:

```
{  
  "ResourceArn": "arn:aws:iot1click:us-west-2:123456789012:devices/  
G030PM0123456789",  
  "Tags": {  
    "Driver": "Jorge Souza",  
    "Driver Phone": "123-555-0199"  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

unclaim-device

Das folgende Codebeispiel zeigt die Verwendung `unclaim-device`.

AWS CLI

Um die Inanspruchnahme (Abmeldung) eines Geräts von Ihrem Konto aufzuheben AWS

Im folgenden `unclaim-device` Beispiel wird das angegebene Gerät von Ihrem Konto zurückgenommen (deregistriert). AWS

```
aws iot1click-devices unclaim-device \  
  --device-id G030PM0123456789
```

Ausgabe:

```
{
  "State": "UNCLAIMED"
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [UnclaimDevice](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer AWS-Geräteressource zu entfernen

Im folgenden `untag-resource`-Beispiel werden die Tags mit den Namen `Driver`, `Phone` und `Driver` aus der angegebenen Gerätereource entfernt.

```
aws iot1click-devices untag-resource \
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters" \
  --tag-keys "Driver Phone" "Driver"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-device-state

Das folgende Codebeispiel zeigt die Verwendung `update-device-state`.

AWS CLI

Um den Status `aktiviert` für ein Gerät zu aktualisieren

Im Folgenden wird der Status des `update-device-state` angegebenen Geräts auf `gesetzt`.
`enabled`

```
aws iot1click-devices update-device-state \  
  --device-id G030PM0123456789 \  
  --enabled
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [UpdateDeviceState](#) in der AWS CLI Befehlsreferenz.

AWS IoT 1-Click Beispiele für Projekte mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with AWS IoT 1-Click Projects Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-device-with-placement

Das folgende Codebeispiel zeigt die Verwendung `associate-device-with-placement`.

AWS CLI

So verknüpfen Sie ein AWS IoT 1-Click 1-Click-Gerät mit einer vorhandenen Platzierung

Im folgenden `associate-device-with-placement` Beispiel wird das angegebene AWS IoT 1-Click 1-Click-Gerät einer vorhandenen Platzierung zugeordnet.

```
aws iot1click-projects associate-device-with-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request \  
  --device-id G030PM0123456789
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [AssociateDeviceWithPlacement](#) in der AWS CLI Befehlsreferenz.

create-placement

Das folgende Codebeispiel zeigt, wie man es benutzt `create-placement`.

AWS CLI

So erstellen Sie eine AWS IoT-1-Click-Platzierung für ein Projekt

Im folgenden `create-placement` Beispiel wird eine AWS IoT-1-Click-Platzierung für das angegebene Projekt erstellt.

```
aws iot1click-projects create-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --attributes '{"location": "123 Any Street Anytown, USA 10001", "phone":  
  "123-456-7890"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [CreatePlacement](#) in der AWS CLI Befehlsreferenz.

create-project

Das folgende Codebeispiel zeigt, wie man es benutzt `create-project`.

AWS CLI

Um ein AWS IoT-1-Click-Projekt für null oder mehr Platzierungen zu erstellen

Im folgenden `create-project` Beispiel wird ein AWS IoT-1-Click-Projekt für eine Platzierung erstellt.

```
aws iot1click-projects create-project -- file: //create-project.json cli-input-json
```

Inhalt von `create-project.json`:

```
{
  "projectName": "AnytownDumpsters",
  "description": "All dumpsters in the Anytown region.",
  "placementTemplate": {
    "defaultAttributes": {
      "City" : "Anytown"
    },
    "deviceTemplates": {
      "empty-dumpster-request" : {
        "deviceType": "button"
      }
    }
  }
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [CreateProject](#) in der AWS CLI Befehlsreferenz.

delete-placement

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-placement`.

AWS CLI

Um eine Platzierung aus einem Projekt zu löschen

Im folgenden `delete-placement` Beispiel wird die angegebene Platzierung aus einem Projekt gelöscht.

```
aws iot1click-projects delete-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [DeletePlacement](#) in der AWS CLI Befehlsreferenz.

delete-project

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-project`.

AWS CLI

Um ein Projekt aus Ihrem AWS Konto zu löschen

Das folgende `delete-project` Beispiel löscht das angegebene Projekt aus Ihrem AWS Konto.

```
aws iot1click-projects delete-project \  
  --project-name AnytownDumpsters
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [DeleteProject](#) in der AWS CLI Befehlsreferenz.

describe-placement

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-placement`.

AWS CLI

Um ein Praktikum für ein Projekt zu beschreiben

Das folgende `describe-placement` Beispiel beschreibt ein Praktikum für das angegebene Projekt.

```
aws iot1click-projects describe-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Ausgabe:

```
{  
  "placement": {  
    "projectName": "AnytownDumpsters",  
    "placementName": "customer217",  
    "attributes": {  
      "phone": "123-555-0110",  
      "location": "123 Any Street Anytown, USA 10001"  
    },  
    "createdDate": 1563488454,  
    "updatedAt": 1563488454  
  }  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [DescribePlacement](#) in der AWS CLI Befehlsreferenz.

describe-project

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-project`.

AWS CLI

Um ein AWS IoT-1-Click-Projekt zu beschreiben

Das folgende `describe-project` Beispiel beschreibt das angegebene AWS IoT 1-Click 1-Click-Projekt.

```
aws iot1click-projects describe-project \  
  --project-name AnytownDumpsters
```

Ausgabe:

```
{
```

```
"project": {
  "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/AnytownDumpsters",
  "projectName": "AnytownDumpsters",
  "description": "All dumpsters in the Anytown region.",
  "createdDate": 1563483100,
  "updatedAt": 1563483100,
  "placementTemplate": {
    "defaultAttributes": {
      "City": "Anytown"
    },
    "deviceTemplates": {
      "empty-dumpster-request": {
        "deviceType": "button",
        "callbackOverrides": {}
      }
    }
  },
  "tags": {}
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [DescribeProject](#) in der AWS CLI Befehlsreferenz.

disassociate-device-from-placement

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-device-from-placement`.

AWS CLI

Um die Zuordnung eines Geräts zu einer Platzierung zu trennen

Im folgenden `disassociate-device-from-placement` Beispiel wird die Zuordnung des angegebenen Geräts zu einer Platzierung aufgehoben.

```
aws iot1click-projects disassociate-device-from-placement \
  --project-name AnytownDumpsters \
  --placement-name customer217 \
  --device-template-name empty-dumpster-request
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [DisassociateDeviceFromPlacement](#) in der AWS CLI Befehlsreferenz.

get-devices-in-placement

Das folgende Codebeispiel zeigt, wie man es benutzt `get-devices-in-placement`.

AWS CLI

Um alle Geräte in einer Platzierung aufzulisten, die in einem Projekt enthalten sind

Im folgenden `get-devices-in-placement` Beispiel werden alle Geräte an der angegebenen Platzierung aufgeführt, die im angegebenen Projekt enthalten sind.

```
aws iot1click-projects get-devices-in-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Ausgabe:

```
{  
  "devices": {  
    "empty-dumpster-request": "G030PM0123456789"  
  }  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [GetDevicesInPlacement](#) in der AWS CLI Befehlsreferenz.

list-placements

Das folgende Codebeispiel zeigt, wie man es benutzt `list-placements`.

AWS CLI

Um alle AWS IoT 1-Click 1-Click-Platzierungen für ein Projekt aufzulisten

Das folgende `list-placements` Beispiel listet alle AWS IoT 1-Click 1-Click-Platzierungen für das angegebene Projekt auf.

```
aws iot1click-projects list-placements \  
  --project-name AnytownDumpsters
```

Ausgabe:

```
{  
  "placements": [  
    {  
      "projectName": "AnytownDumpsters",  
      "placementName": "customer217",  
      "createdDate": 1563488454,  
      "updatedAt": 1563488454  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ListPlacements](#) in der AWS CLI Befehlsreferenz.

list-projects

Das folgende Codebeispiel zeigt, wie man es benutzt `list-projects`.

AWS CLI

Um alle AWS IoT 1-Click 1-Click-Projekte aufzulisten

Das folgende `list-projects` Beispiel listet alle AWS IoT 1-Click 1-Click-Projekte in Ihrem Konto auf.

```
aws iot1click-projects list-projects
```

Ausgabe:

```
{  
  "projects": [  
    {  
      "projectName": "AnytownDumpsters",  
      "placementName": "customer217",  
      "createdDate": 1563488454,  
      "updatedAt": 1563488454  
    }  
  ]  
}
```

```
{
  "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/
AnytownDumpsters",
  "projectName": "AnytownDumpsters",
  "createdDate": 1563483100,
  "updatedDate": 1563483100,
  "tags": {}
}
]
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ListProjects](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Projektressource aufzulisten

Im folgenden `list-tags-for-resource` Beispiel werden die Tags für die angegebene Projektressource aufgeführt.

```
aws iot1click-projects list-tags-for-resource \
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters"
```

Ausgabe:

```
{
  "tags": {
    "Manager": "Li Juan",
    "Account": "45215"
  }
}
```

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um einer Projektressource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel werden der angegebenen Projektressource zwei Tags hinzugefügt.

```
aws iot1click-projects tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

Inhalt von `devices-tag-resource.json`:

```
{  
  "resourceArn": "arn:aws:iot1click:us-west-2:123456789012:projects/  
  AnytownDumpsters",  
  "tags": {  
    "Account": "45215",  
    "Manager": "Li Juan"  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um Tags aus einer Projektressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem Schlüsselnamen `Manager` aus dem angegebenen Projekt entfernt.

```
aws iot1click-projects untag-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters" \  
  --tag-keys "Manager"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-placement

Das folgende Codebeispiel zeigt, wie man es benutzt `update-placement`.

AWS CLI

Um die Schlüssel-Wert-Paare für „Attribute“ einer Platzierung zu aktualisieren

Im folgenden `update-placement` Beispiel werden die Schlüssel-Wert-Paare für „Attribute“ einer Platzierung aktualisiert.

```
aws iot1click-projects update-placement \  
  --cli-input-json file://update-placement.json
```

Inhalt von `update-placement.json`:

```
{  
  "projectName": "AnytownDumpsters",  
  "placementName": "customer217",  
  "attributes": {  
    "phone": "123-456-7890",  
    "location": "123 Any Street Anytown, USA 10001"  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [UpdatePlacement](#) in der AWS CLI Befehlsreferenz.

update-project

Das folgende Codebeispiel zeigt, wie man es benutzt `update-project`.

AWS CLI

Um die Einstellungen für ein Projekt zu aktualisieren

Im folgenden `update-project` Beispiel wird die Beschreibung für ein Projekt aktualisiert.

```
aws iot1click-projects update-project \  
  --project-name AnytownDumpsters \  
  --description "All dumpsters (yard waste, recycling, garbage) in the Anytown region."
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using AWS IoT 1-Click with the AWS CLI](#) im AWS IoT 1-Click Developer Guide.

- API-Einheiten finden Sie [UpdateProject](#) in der AWS CLI Befehlsreferenz.

AWS IoT Analytics Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Analytics.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-put-message

Das folgende Codebeispiel zeigt die Verwendung `batch-put-message`.

AWS CLI

Um eine Nachricht an einen Kanal zu senden

Das folgende `batch-put-message` Beispiel sendet eine Nachricht an den angegebenen Kanal.

```
aws iotanalytics batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://batch-put-message.json
```

Inhalt von `batch-put-message.json`:

```
{  
  "channelName": "mychannel",  
  "messages": [  
    {  
      "messageId": "0001",  
      "payload": "eyJhdGVtcGVyYXR1cmUiOiAyMCB9"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "batchPutMessageErrorEntries": []  
}
```

Weitere Informationen finden Sie [BatchPutMessage](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [BatchPutMessage](#) unter AWS CLI Befehlsreferenz.

cancel-pipeline-reprocessing

Das folgende Codebeispiel zeigt die Verwendung `cancel-pipeline-reprocessing`.

AWS CLI

Um die Wiederverarbeitung von Daten über eine Pipeline abzubrechen

Im folgenden `cancel-pipeline-reprocessing` Beispiel wird die Wiederverarbeitung von Daten über die angegebene Pipeline abgebrochen.

```
aws iotanalytics cancel-pipeline-reprocessing \  
  --pipeline-name mypipeline \  
  --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [CancelPipelineReprocessing](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [CancelPipelineReprocessing](#) unter AWS CLI Befehlsreferenz.

create-channel

Das folgende Codebeispiel zeigt die Verwendung `create-channel`.

AWS CLI

Um einen Kanal zu erstellen

Im folgenden `create-channel` Beispiel wird ein Kanal mit der angegebenen Konfiguration erstellt. Ein Kanal sammelt Daten zu einem MQTT Thema und archiviert die unverarbeiteten Rohnachrichten, bevor die Daten in einer Pipeline veröffentlicht werden.

```
aws iotanalytics create-channel \  
  --cli-input-json file://create-channel.json
```

Inhalt von `create-channel.json`:

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "unlimited": true  
  },  
}
```

```
"tags": [
  {
    "key": "Environment",
    "value": "Production"
  }
]
```

Ausgabe:

```
{
  "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",
  "channelName": "mychannel",
  "retentionPeriod": {
    "unlimited": true
  }
}
```

Weitere Informationen finden Sie [CreateChannel](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [CreateChannel](#) unter AWS CLI Befehlsreferenz.

create-dataset-content

Das folgende Codebeispiel zeigt die Verwendung `create-dataset-content`.

AWS CLI

Um den Inhalt eines Datensatzes zu erstellen

Im folgenden `create-dataset-content` Beispiel wird der Inhalt des angegebenen Datensatzes erstellt, indem eine `queryAction` (eine SQL-Abfrage) oder eine `containerAction` (Ausführung einer containerisierten Anwendung) angewendet wird.

```
aws iotanalytics create-dataset-content \
  --dataset-name mydataset
```

Ausgabe:

```
{
  "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"
```

```
}
```

Weitere Informationen finden Sie [CreateDatasetContent](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [CreateDatasetContent](#) unter AWS CLI Befehlsreferenz.

create-dataset

Das folgende Codebeispiel zeigt die Verwendung `create-dataset`.

AWS CLI

Um einen Datensatz zu erstellen

Im folgenden `create-dataset` Beispiel wird ein Datensatz erstellt. Eine Datenmenge speichert Daten, die aus einem Datenspeicher abgerufen wurden, indem eine `queryAction` (eine SQL Abfrage) oder eine `containerAction` (Ausführung einer containerisierten Anwendung) angewendet wird. Diese Operation erstellt das Grundgerüst eines Datensatzes. Sie können den Datensatz manuell auffüllen, indem Sie ihn aufrufen, `CreateDatasetContent` oder automatisch entsprechend einem von `trigger` Ihnen angegebenen Wert.

```
aws iotanalytics create-dataset \  
  --cli-input-json file://create-dataset.json
```

Inhalt von `create-dataset.json`:

```
{  
  "datasetName": "mydataset",  
  "actions": [  
    {  
      "actionName": "myDatasetAction",  
      "queryAction": {  
        "sqlQuery": "SELECT * FROM mydatastore"  
      }  
    }  
  ],  
  "retentionPeriod": {  
    "unlimited": true  
  },  
  "tags": [  
    {  
      "key": "mykey",  
      "value": "myvalue"  
    }  
  ]  
}
```

```
{
  "key": "Environment",
  "value": "Production"
}
]
```

Ausgabe:

```
{
  "datasetName": "mydataset",
  "retentionPeriod": {
    "unlimited": true
  },
  "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

Weitere Informationen finden Sie [CreateDataset](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [CreateDataset](#) unter AWS CLI Befehlsreferenz.

create-datastore

Das folgende Codebeispiel zeigt die Verwendung `create-datastore`.

AWS CLI

Um einen Datenspeicher zu erstellen

Im folgenden `create-datastore` Beispiel wird ein Datenspeicher erstellt, der ein Repository für Nachrichten ist.

```
aws iotanalytics create-datastore \
  --cli-input-json file://create-datastore.json
```

Inhalt von `create-datastore.json`:

```
{
  "datastoreName": "mydatastore",
  "retentionPeriod": {
    "numberOfDays": 90
  }
}
```

```
    },
    "tags": [
      {
        "key": "Environment",
        "value": "Production"
      }
    ]
  }
}
```

Ausgabe:

```
{
  "datastoreName": "mydatastore",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
mydatastore",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

Weitere Informationen finden Sie [CreateDatastore](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [CreateDatastore](#) unter AWS CLI Befehlsreferenz.

create-pipeline

Das folgende Codebeispiel zeigt die Verwendung `create-pipeline`.

AWS CLI

Erstellen Sie eine IoT-Analytics-Pipeline

Im folgenden `create-pipeline` Beispiel wird eine Pipeline erstellt. Eine Pipeline nimmt Nachrichten aus einem Kanal auf und ermöglicht Ihnen, die Nachrichten vor dem Speichern in einem Datenspeicher zu verarbeiten. Sie müssen sowohl eine Kanal- als auch eine Datenspeicheraktivität und optional bis zu 23 zusätzliche Aktivitäten im `pipelineActivities` Array angeben.

```
aws iotanalytics create-pipeline \
```



```
--cli-input-json file://create-pipeline.json
```

Inhalt von `create-pipeline.json`:

```
{
  "pipelineName": "mypipeline",
  "pipelineActivities": [
    {
      "channel": {
        "name": "myChannelActivity",
        "channelName": "mychannel",
        "next": "myMathActivity"
      }
    },
    {
      "datastore": {
        "name": "myDatastoreActivity",
        "datastoreName": "mydatastore"
      }
    },
    {
      "math": {
        "name": "myMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC",
        "next": "myDatastoreActivity"
      }
    }
  ],
  "tags": [
    {
      "key": "Environment",
      "value": "Beta"
    }
  ]
}
```

Ausgabe:

```
{
  "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
  "pipelineName": "mypipeline"
}
```

```
}
```

Weitere Informationen finden Sie [CreatePipeline](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [CreatePipeline](#) unter AWS CLI Befehlsreferenz.

delete-channel

Das folgende Codebeispiel zeigt die Verwendung `delete-channel`.

AWS CLI

Löschen Sie einen IoT Analytics Analytics-Kanal

Im folgenden `delete-channel` Beispiel wird der angegebene Kanal gelöscht.

```
aws iotanalytics delete-channel \  
  --channel-name mychannel
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteChannel](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DeleteChannel](#) unter AWS CLI Befehlsreferenz.

delete-dataset-content

Das folgende Codebeispiel zeigt die Verwendung `delete-dataset-content`.

AWS CLI

Um den Inhalt eines Datensatzes zu löschen

Im folgenden `delete-dataset-content` Beispiel wird der Inhalt des angegebenen Datensatzes gelöscht.

```
aws iotanalytics delete-dataset-content \  
  --dataset-name mydataset
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteDatasetContent](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DeleteDatasetContent](#) unter AWS CLI Befehlsreferenz.

delete-dataset

Das folgende Codebeispiel zeigt die Verwendung `delete-dataset`.

AWS CLI

Um einen Datensatz zu löschen

Im folgenden `delete-dataset` Beispiel wird der angegebene Datensatz gelöscht. Sie müssen den Inhalt des Datensatzes nicht löschen, bevor Sie diesen Vorgang ausführen.

```
aws iotanalytics delete-dataset \  
  --dataset-name mydataset
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteDataset](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DeleteDataset](#) unter AWS CLI Befehlsreferenz.

delete-datastore

Das folgende Codebeispiel zeigt die Verwendung `delete-datastore`.

AWS CLI

Um einen Datenspeicher zu löschen

Im folgenden `delete-datastore` Beispiel wird der angegebene Datenspeicher gelöscht.

```
aws iotanalytics delete-datastore \  
  --datastore-name mydatastore
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteDatastore](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DeleteDatastore](#) unter AWS CLI Befehlsreferenz.

delete-pipeline

Das folgende Codebeispiel zeigt die Verwendung `delete-pipeline`.

AWS CLI

Um eine Pipeline zu löschen

Im folgenden `delete-pipeline` Beispiel wird die angegebene Pipeline gelöscht.

```
aws iotanalytics delete-pipeline \  
  --pipeline-name mypipeline
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeletePipeline](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DeletePipeline](#) unter AWS CLI Befehlsreferenz.

describe-channel

Das folgende Codebeispiel zeigt die Verwendung `describe-channel`.

AWS CLI

Um Informationen über einen Kanal abzurufen

Im folgenden `describe-channel` Beispiel werden Details, einschließlich Statistiken, für den angegebenen Kanal angezeigt.

```
aws iotanalytics describe-channel \  
  --channel-name mychannel \  
  --include-statistics
```

Ausgabe:

```
{  
  "statistics": {
```

```

    "size": {
      "estimatedSizeInBytes": 402.0,
      "estimatedOn": 1561504380.0
    }
  },
  "channel": {
    "status": "ACTIVE",
    "name": "mychannel",
    "lastUpdateTime": 1557860351.001,
    "creationTime": 1557860351.001,
    "retentionPeriod": {
      "unlimited": true
    },
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
  }
}

```

Weitere Informationen finden Sie [DescribeChannel](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DescribeChannel](#) unter AWS CLI Befehlsreferenz.

describe-dataset

Das folgende Codebeispiel zeigt die Verwendung `describe-dataset`.

AWS CLI

Um Informationen über einen Datensatz abzurufen

Im folgenden `describe-dataset` Beispiel werden Details für den angegebenen Datensatz angezeigt.

```

aws iotanalytics describe-dataset \
  --dataset-name mydataset

```

Ausgabe:

```

{
  "dataset": {
    "status": "ACTIVE",
    "contentDeliveryRules": [],

```

```
"name": "mydataset",
"lastUpdateTime": 1557859240.658,
"triggers": [],
"creationTime": 1557859240.658,
"actions": [
  {
    "actionName": "query_32",
    "queryAction": {
      "sqlQuery": "SELECT * FROM mydatastore",
      "filters": []
    }
  }
],
"retentionPeriod": {
  "numberOfDays": 90,
  "unlimited": false
},
"arn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

Weitere Informationen finden Sie [DescribeDataset](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DescribeDataset](#) unter AWS CLI Befehlsreferenz.

describe-datastore

Das folgende Codebeispiel zeigt die Verwendung `describe-datastore`.

AWS CLI

Um Informationen über einen Datenspeicher abzurufen

Im folgenden `describe-datastore` Beispiel werden Details, einschließlich Statistiken, für den angegebenen Datenspeicher angezeigt.

```
aws iotanalytics describe-datastore \
  --datastore-name mydatastore \
  --include-statistics
```

Ausgabe:

```
{
  "datastore": {
    "status": "ACTIVE",
    "name": "mydatastore",
    "lastUpdateTime": 1557858971.02,
    "creationTime": 1557858971.02,
    "retentionPeriod": {
      "unlimited": true
    },
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/mydatastore"
  },
  "statistics": {
    "size": {
      "estimatedSizeInBytes": 397.0,
      "estimatedOn": 1561592040.0
    }
  }
}
```

Weitere Informationen finden Sie [DescribeDatastore](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DescribeDatastore](#) unter AWS CLI Befehlsreferenz.

describe-logging-options

Das folgende Codebeispiel zeigt die Verwendung `describe-logging-options`.

AWS CLI

Um die aktuellen Protokollierungsoptionen abzurufen

Das folgende `describe-logging-options` Beispiel zeigt die aktuellen AWS IoT Analytics Analytics-Protokollierungsoptionen.

```
aws iotanalytics describe-logging-options
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
```

```
    "enabled": true,  
    "level": "ERROR"  
  }  
}
```

Weitere Informationen finden Sie [DescribeLoggingOptions](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DescribeLoggingOptions](#) unter AWS CLI Befehlsreferenz.

describe-pipeline

Das folgende Codebeispiel zeigt die Verwendung `describe-pipeline`.

AWS CLI

Um Informationen über eine Pipeline abzurufen

Im folgenden `describe-pipeline` Beispiel werden Details für die angegebene Pipeline angezeigt.

```
aws iotanalytics describe-pipeline \  
  --pipeline-name mypipeline
```

Ausgabe:

```
{  
  "pipeline": {  
    "activities": [  
      {  
        "channel": {  
          "channelName": "mychannel",  
          "name": "mychannel_28",  
          "next": "mydatastore_29"  
        }  
      },  
      {  
        "datastore": {  
          "datastoreName": "mydatastore",  
          "name": "mydatastore_29"  
        }  
      }  
    ]  
  }  
}
```



```
    ],
    "name": "mypipeline",
    "lastUpdateTime": 1561676362.515,
    "creationTime": 1557859124.432,
    "reprocessingSummaries": [
      {
        "status": "SUCCEEDED",
        "creationTime": 1561676362.189,
        "id": "6ad2764f-fb13-4de3-b101-4e74af03b043"
      }
    ],
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/mypipeline"
  }
}
```

Weitere Informationen finden Sie [DescribePipeline](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [DescribePipeline](#) unter AWS CLI Befehlsreferenz.

get-dataset-content

Das folgende Codebeispiel zeigt die Verwendung `get-dataset-content`.

AWS CLI

Um den Inhalt eines Datensatzes abzurufen

Im folgenden `get-dataset-content` Beispiel wird der Inhalt eines Datensatzes mit URIs Vorzeichen abgerufen.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Ausgabe:

```
{
  "status": {
    "state": "SUCCEEDED"
  },
  "timestamp": 1557863215.995,
  "entries": [
    {
```

```

        "dataURI": "https://aws-radiant-
dataset-12345678-1234-1234-1234-123456789012.s3.us-west-2.amazonaws.com/
results/12345678-e8b3-46ba-b2dd-efe8d86cf385.csv?X-Amz-Security-Token=...-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190628T173437Z&X-Amz-SignedHeaders=host&X-
Amz-Expires=7200&X-Amz-Credential=...F20190628%2Fus-west-2%2Fs3%2Faws4_request&X-
Amz-Signature=..."
    }
]
}

```

Weitere Informationen finden Sie [GetDatasetContent](#) im Handbuch.

- API-Einheiten finden Sie [GetDatasetContent](#) in der AWS CLI Befehlsreferenz.

list-channels

Das folgende Codebeispiel zeigt die Verwendung `list-channels`.

AWS CLI

Um eine Liste von Kanälen abzurufen

Im folgenden `list-channels` Beispiel werden Übersichtsinformationen für die verfügbaren Kanäle angezeigt.

```
aws iotanalytics list-channels
```

Ausgabe:

```

{
  "channelSummaries": [
    {
      "status": "ACTIVE",
      "channelName": "mychannel",
      "creationTime": 1557860351.001,
      "lastUpdateTime": 1557860351.001
    }
  ]
}

```

Weitere Informationen finden Sie [ListChannels](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [ListChannels](#) unter AWS CLI Befehlsreferenz.

list-dataset-contents

Das folgende Codebeispiel zeigt die Verwendung `list-dataset-contents`.

AWS CLI

Um Informationen über den Inhalt von Datensätzen aufzulisten

Das folgende `list-dataset-contents` Beispiel listet Informationen zu Datensatzinhalten auf, die erstellt wurden.

```
aws iotanalytics list-dataset-contents \  
  --dataset-name mydataset
```

Ausgabe:

```
{  
  "datasetContentSummaries": [  
    {  
      "status": {  
        "state": "SUCCEEDED"  
      },  
      "scheduleTime": 1557863215.995,  
      "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",  
      "creationTime": 1557863215.995  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ListDatasetContents](#) in der AWS IoT Analytics API Analytics-Referenz.

- API Einzelheiten finden Sie [ListDatasetContents](#) unter AWS CLI Befehlsreferenz.

list-datasets

Das folgende Codebeispiel zeigt die Verwendung `list-datasets`.

AWS CLI

Um Informationen über Datensätze abzurufen

Das folgende `list-datasets` Beispiel listet zusammenfassende Informationen zu verfügbaren Datensätzen auf.

```
aws iotanalytics list-datasets
```

Ausgabe:

```
{
  "datasetSummaries": [
    {
      "status": "ACTIVE",
      "datasetName": "mydataset",
      "lastUpdateTime": 1557859240.658,
      "triggers": [],
      "creationTime": 1557859240.658,
      "actions": [
        {
          "actionName": "query_32",
          "actionType": "QUERY"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie [ListDatasets](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [ListDatasets](#) unter AWS CLI Befehlsreferenz.

list-datastores

Das folgende Codebeispiel zeigt die Verwendung `list-datastores`.

AWS CLI

Um eine Liste von Datenspeichern abzurufen

Im folgenden `list-datastores` Beispiel werden zusammenfassende Informationen zu den verfügbaren Datenspeichern angezeigt.

```
aws iotanalytics list-datastores
```

Ausgabe:

```
{
  "datastoreSummaries": [
    {
      "status": "ACTIVE",
      "datastoreName": "mydatastore",
      "creationTime": 1557858971.02,
      "lastUpdateTime": 1557858971.02
    }
  ]
}
```

Weitere Informationen finden Sie [ListDatastores](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [ListDatastores](#) unter AWS CLI Befehlsreferenz.

list-pipelines

Das folgende Codebeispiel zeigt die Verwendung `list-pipelines`.

AWS CLI

Um eine Liste von Pipelines abzurufen

Im folgenden `list-pipelines` Beispiel wird eine Liste verfügbarer Pipelines angezeigt.

```
aws iotanalytics list-pipelines
```

Ausgabe:

```
{
  "pipelineSummaries": [
    {
      "pipelineName": "mypipeline",
      "creationTime": 1557859124.432,
      "lastUpdateTime": 1557859124.432,
      "reprocessingSummaries": []
    }
  ]
}
```

Weitere Informationen finden Sie [ListPipelines](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [ListPipelines](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags auf, die Sie an die angegebene Ressource angehängt haben.

```
aws iotanalytics list-tags-for-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

Ausgabe:

```
{  
  "tags": [  
    {  
      "value": "bar",  
      "key": "foo"  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ListTagsForResource](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

put-logging-options

Das folgende Codebeispiel zeigt die Verwendung `put-logging-options`.

AWS CLI

Um Protokollierungsoptionen festzulegen oder zu aktualisieren

Im folgenden `put-logging-options` Beispiel werden die AWS IoT Analytics Analytics-Protokollierungsoptionen festgelegt oder aktualisiert. Wenn Sie den Wert eines `loggingOptions` Felds aktualisieren, kann es bis zu einer Minute dauern, bis die Änderung wirksam wird. Wenn Sie außerdem die Richtlinie ändern, die der Rolle zugeordnet ist, die Sie im Feld `roleArn` angegeben haben (z. B. um eine ungültige Richtlinie zu korrigieren), kann es bis zu fünf Minuten dauern, bis diese Änderung wirksam wird.

```
aws iotanalytics put-logging-options \  
  --cli-input-json file://put-logging-options.json
```

Inhalt von `put-logging-options.json`:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",  
    "level": "ERROR",  
    "enabled": true  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [PutLoggingOptions](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [PutLoggingOptions](#) unter AWS CLI Befehlsreferenz.

run-pipeline-activity

Das folgende Codebeispiel zeigt die Verwendung `run-pipeline-activity`.

AWS CLI

Um eine Pipeline-Aktivität zu simulieren

Im folgenden `run-pipeline-activity` Beispiel werden die Ergebnisse der Ausführung einer Pipeline-Aktivität auf einer Nachrichtennutzlast simuliert.

```
aws iotanalytics run-pipeline-activity \  
  --pipeline-activity file://maths.json \  
  --cli-input-json file://run-pipeline-activity-input.json
```

```
--payloads file://payloads.json
```

Inhalt von `maths.json`:

```
{
  "math": {
    "name": "MyMathActivity",
    "math": "((temp - 32) * 5.0) / 9.0",
    "attribute": "tempC"
  }
}
```

Inhalt von `payloads.json`:

```
[
  "{\"humidity\": 52, \"temp\": 68 }",
  "{\"humidity\": 52, \"temp\": 32 }"
]
```

Ausgabe:

```
{
  "logResult": "",
  "payloads": [
    "eyJodW1pZG10eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
    "eyJodW1pZG10eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
  ]
}
```

Weitere Informationen finden Sie [RunPipelineActivity](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [RunPipelineActivity](#) unter AWS CLI Befehlsreferenz.

sample-channel-data

Das folgende Codebeispiel zeigt die Verwendung `sample-channel-data`.

AWS CLI

Um Beispielnachrichten von einem Kanal abzurufen

Im folgenden `sample-channel-data` Beispiel wird eine Stichprobe von Nachrichten aus dem angegebenen Kanal abgerufen, die während des angegebenen Zeitraums aufgenommen wurden. Sie können bis zu 10 Nachrichten abrufen.

```
aws iotanalytics sample-channel-data \  
  --channel-name mychannel
```

Ausgabe:

```
{  
  "payloads": [  
    "eyJhdGVtcGVyYXR1cmUiOiAyMCM9",  
    "eyJhZm9vIjogImJhcnVzIj0="
```

Weitere Informationen finden Sie [SampleChannelData](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [SampleChannelData](#) unter AWS CLI Befehlsreferenz.

start-pipeline-reprocessing

Das folgende Codebeispiel zeigt die Verwendung `start-pipeline-reprocessing`.

AWS CLI

Um die Wiederverarbeitung der Pipeline zu starten

Im folgenden `start-pipeline-reprocessing` Beispiel wird die Neuverarbeitung von Nachrichtenrohdaten über die angegebene Pipeline gestartet.

```
aws iotanalytics start-pipeline-reprocessing \  
  --pipeline-name mypipeline
```

Ausgabe:

```
{  
  "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

Weitere Informationen finden Sie [StartPipelineReprocessing](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [StartPipelineReprocessing](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags für eine Ressource hinzuzufügen oder zu ändern

Im folgenden `tag-resource` Beispiel werden die Tags, die an die angegebene Ressource angehängt sind, erweitert oder geändert.

```
aws iotanalytics tag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tags "[{"key": "Environment", "value": "Production"}]"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [TagResource](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden die Tags mit den angegebenen Schlüsselnamen aus der angegebenen Ressource entfernt.

```
aws iotanalytics untag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tag-keys ["Environment"]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter `UntagResource` < https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html > in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-channel

Das folgende Codebeispiel zeigt die Verwendung `update-channel`.

AWS CLI

Um einen Kanal zu ändern

Im folgenden `update-channel` Beispiel werden die Einstellungen für den angegebenen Kanal geändert.

```
aws iotanalytics update-channel \  
  --cli-input-json file://update-channel.json
```

Inhalt von `update-channel.json`:

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UpdateChannel](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [UpdateChannel](#) unter AWS CLI Befehlsreferenz.

update-dataset

Das folgende Codebeispiel zeigt die Verwendung `update-dataset`.

AWS CLI

Um einen Datensatz zu aktualisieren

Im folgenden update-dataset Beispiel werden die Einstellungen des angegebenen Datensatzes geändert.

```
aws iotanalytics update-dataset \  
  --cli-input-json file://update-dataset.json
```

Inhalt von update-dataset.json:

```
{  
  "datasetName": "mydataset",  
  "actions": [  
    {  
      "actionName": "myDatasetUpdateAction",  
      "queryAction": {  
        "sqlQuery": "SELECT * FROM mydatastore"  
      }  
    }  
  ],  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter UpdateDataset < https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html > in der AWS IoT Analytics API Analytics-Referenz.

- APIEinzelheiten finden Sie [UpdateDataset](#) unter AWS CLI Befehlsreferenz.

update-datastore

Das folgende Codebeispiel zeigt die Verwendung update-datastore.

AWS CLI

Um einen Datenspeicher zu aktualisieren

Im folgenden `update-datastore` Beispiel werden die Einstellungen des angegebenen Datenspeichers geändert.

```
aws iotanalytics update-datastore \  
  --cli-input-json file://update-datastore.json
```

Inhalt von `update-datastore.json`:

```
{  
  "datastoreName": "mydatastore",  
  "retentionPeriod": {  
    "numberOfDays": 93  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UpdateDatastore](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [UpdateDatastore](#) unter AWS CLI Befehlsreferenz.

update-pipeline

Das folgende Codebeispiel zeigt die Verwendung `update-pipeline`.

AWS CLI

Um eine Pipeline zu aktualisieren

Im folgenden `update-pipeline` Beispiel werden die Einstellungen der angegebenen Pipeline geändert. Sie müssen sowohl eine Kanal- als auch eine Datenspeicheraktivität und optional bis zu 23 zusätzliche Aktivitäten im `pipelineActivities` Array angeben.

```
aws iotanalytics update-pipeline \  
  --cli-input-json file://update-pipeline.json
```

Inhalt von `update-pipeline.json`:

```
{  
  "pipelineName": "mypipeline",
```

```
"pipelineActivities": [  
  {  
    "channel": {  
      "name": "myChannelActivity",  
      "channelName": "mychannel",  
      "next": "myMathActivity"  
    }  
  },  
  {  
    "datastore": {  
      "name": "myDatastoreActivity",  
      "datastoreName": "mydatastore"  
    }  
  },  
  {  
    "math": {  
      "name": "myMathActivity",  
      "math": "(((temp - 32) * 5.0) / 9.0) + 273.15",  
      "attribute": "tempK",  
      "next": "myDatastoreActivity"  
    }  
  }  
]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UpdatePipeline](#) in der AWS IoT Analytics API Analytics-Referenz.

- API-Einheiten finden Sie [UpdatePipeline](#) unter AWS CLI Befehlsreferenz.

Device Advisor-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Device Advisor Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-suite-definition

Das folgende Codebeispiel zeigt die Verwendung `create-suite-definition`.

AWS CLI

Beispiel 1: So erstellen Sie eine IoT Device Advisor-Testsuite

Im folgenden `create-suite-definition` Beispiel wird eine Device Advisor-Testsuite im AWS IoT mit der angegebenen Suite-Definitionskonfiguration erstellt.

```
aws iotdeviceadvisor create-suite-definition \
  --suite-definition-configuration '{ \
    "suiteDefinitionName": "TestSuiteName", \
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIoTThing"}], \
    "intendedForQualification": false, \
    "rootGroup": "{\"configuration\": {}, \"tests\": [{\"name\": \"MQTT Connect\", \
  \"configuration\": {\"EXECUTION_TIMEOUT\": 120}, \"tests\": [{\"name\": \"MQTT_Connect\", \
  \"configuration\": {}, \"test\": {\"id\": \"MQTT_Connect\", \"testCase\": null, \"version \
  \": \"0.0.0\"}}]}]}", \
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

Ausgabe:

```
{
  "suiteDefinitionId": "0jtsgio7yenu",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/0jtsgio7yenu",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

Weitere Informationen finden Sie unter [Erstellen einer Testsuite-Definition](#) im AWS IoT Core Developer Guide.

Beispiel 2: So erstellen Sie eine Testsuite für die neueste Qualifikation von IoT Device Advisor

Im folgenden `create-suite-definition` Beispiel wird eine Device Advisor-Qualifizierungstestsuite mit der neuesten Version im AWS IoT mit der angegebenen Suite-Definitionskonfiguration erstellt.

```
aws iotdeviceadvisor create-suite-definition \
  --suite-definition-configuration '{ \
    "suiteDefinitionName": "TestSuiteName", \
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIoTThing"}], \
    "intendedForQualification": true, \
    "rootGroup": "", \
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

Ausgabe:

```
{
  "suiteDefinitionId": "txgsuolk2myj",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/txgsuolk2myj",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

Weitere Informationen finden Sie unter [Erstellen einer Testsuite-Definition](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [CreateSuiteDefinition](#) in der AWS CLI Befehlsreferenz.

delete-suite-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-suite-definition`.

AWS CLI

So löschen Sie die IoT Device Advisor-Testsuite

Im folgenden `delete-suite-definition` Beispiel wird die Device Advisor-Testsuite mit der angegebenen Suite-Definition-ID gelöscht.

```
aws iotdeviceadvisor delete-suite-definition \
```



```
--suite-definition-id 0jtsgio7yenu
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteSuiteDefinition](#) in der AWS API IoT-Referenz.

- API Einzelheiten finden Sie [DeleteSuiteDefinition](#) unter AWS CLI Befehlsreferenz.

get-endpoint

Das folgende Codebeispiel zeigt die Verwendung `get-endpoint`.

AWS CLI

Beispiel 1: Um die Informationen über einen IoT Device Advisor-Endpoint auf Kontoebene abzurufen

Im folgenden `get-endpoint` Beispiel werden die Informationen zu einem Testendpunkt auf Device Advisor-Kontoebene abgerufen.

```
aws iotdeviceadvisor get-endpoint
```

Ausgabe:

```
{
  "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Beispiel 2: Um die Informationen über einen IoT Device Advisor-Endpoint auf Geräteebene abzurufen

Im folgenden `get-endpoint` Beispiel werden die Informationen zu einem Device Advisor-Testendpunkt auf Geräteebene mit dem angegebenen Thing-ARN oder Certificate-ARN abgerufen.

```
aws iotdeviceadvisor get-endpoint \
  --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIotThing
```

Ausgabe:

```
{
```

```
"endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Weitere Informationen finden [Sie unter Holen Sie sich einen Testendpunkt](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [GetEndpoint](#) in der AWS CLI Befehlsreferenz.

get-suite-definition

Das folgende Codebeispiel zeigt die Verwendung `get-suite-definition`.

AWS CLI

Um Informationen über eine IoT Device Advisor-Testsuite zu erhalten

Im folgenden `get-suite-definition` Beispiel werden die Informationen zu einer ADevice Advisor-Testsuite mit der angegebenen Suite-Definition-ID abgerufen.

```
aws iotdeviceadvisor get-suite-definition \
  --suite-definition-id qqcsmtyyjabl
```

Ausgabe:

```
{
  "suiteDefinitionId": "qqcsmtyyjabl",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/qqcsmtyyjabl",
  "suiteDefinitionVersion": "v1",
  "latestVersion": "v1",
  "suiteDefinitionConfiguration": {
    "suiteDefinitionName": "MQTT connection",
    "devices": [],
    "intendedForQualification": false,
    "isLongDurationTest": false,
    "rootGroup": "{\"configuration\": {}, \"tests\": [{\"id\": \"uta5d9j1kvw\", \"name\": \"Test group 1\", \"configuration\": {}, \"tests\": [{\"id\": \"awr8pq5vc9yp\", \"name\": \"MQTT Connect\", \"configuration\": {}, \"test\": {\"id\": \"MQTT_Connect\", \"testCase\": null, \"version\": \"0.0.0\"}}]}]}",
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole",
    "protocol": "MqttV3_1_1"
  },
  "createdAt": "2022-11-11T22:28:52.389000-05:00",
```

```
"lastModifiedAt": "2022-11-11T22:28:52.389000-05:00",  
"tags": {}  
}
```

Weitere Informationen finden [Sie unter Get a Test Suite-Definition](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [GetSuiteDefinition](#) in der AWS CLI Befehlsreferenz.

get-suite-run-report

Das folgende Codebeispiel zeigt die Verwendung `get-suite-run-report`.

AWS CLI

Um Informationen zu einer qualifizierten Testsuite für IoT Device Advisor zu erhalten, führen Sie einen Bericht aus

Im folgenden `get-suite-run-report` Beispiel wird der Link zum Herunterladen des Berichts für eine erfolgreiche Ausführung einer für Device Advisor qualifizierten Testsuite mit der angegebenen Suite-Definition-ID und Suite-Run-ID abgerufen.

```
aws iotdeviceadvisor get-suite-run-report \  
  --suite-definition-id ztvb5aek4w4x \  
  --suite-run-id p6awv83nre6v
```

Ausgabe:

```
{  
  "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-  
prod.s3.amazonaws.com/report.downloadlink"  
}
```

Weitere Informationen finden [Sie im AWS IoT Core Developer Guide unter Einen Qualifizierungsbericht für eine erfolgreiche Ausführung einer Qualifizierungstestsuite](#) abrufen.

- API-Einheiten finden Sie [GetSuiteRunReport](#) in der AWS CLI Befehlsreferenz.

get-suite-run

Das folgende Codebeispiel zeigt die Verwendung `get-suite-run`.

AWS CLI

Um Informationen zum Ausführungsstatus einer IoT Device Advisor-Testsuite abzurufen

Im folgenden `get-suite-run` Beispiel werden die Informationen zum Ausführungsstatus einer Device Advisor-Testsuite mit der angegebenen Suite-Definition-ID und Suite-Run-ID abgerufen.

```
aws iotdeviceadvisor get-suite-run \  
  --suite-definition-id qqcsmtyyjabl \  
  --suite-run-id nzlfyhaa18oa
```

Ausgabe:

```
{  
  "suiteDefinitionId": "qqcsmtyyjabl",  
  "suiteDefinitionVersion": "v1",  
  "suiteRunId": "nzlfyhaa18oa",  
  "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/  
qqcsmtyyjabl/nzlfyhaa18oa",  
  "suiteRunConfiguration": {  
    "primaryDevice": {  
      "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing",  
      "certificateArn": "arn:aws:iot:us-east-1:123456789012:cert/certFile"  
    },  
    "parallelRun": false  
  },  
  "testResult": {  
    "groups": [  
      {  
        "groupId": "uta5d9j1kvw",  
        "groupName": "Test group 1",  
        "tests": [  
          {  
            "testCaseRunId": "2ve2twrqyr0s",  
            "testCaseDefinitionId": "awr8pq5vc9yp",  
            "testCaseDefinitionName": "MQTT Connect",  
            "status": "PASS",  
            "startTime": "2022-11-12T00:01:53.693000-05:00",  
            "endTime": "2022-11-12T00:02:15.443000-05:00",  
            "logUrl": "https://console.aws.amazon.com/  
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws/iot/deviceadvisor/  
qqcsmtyyjabl;stream=nzlfyhaa18oa_2ve2twrqyr0s",  
            "warnings": "null",  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

        "failure": "null"
      }
    ]
  },
  "startTime": "2022-11-12T00:01:52.673000-05:00",
  "endTime": "2022-11-12T00:02:16.496000-05:00",
  "status": "PASS",
  "tags": {}
}

```

Weitere Informationen finden [Sie unter Get a Test Suite Run](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [GetSuiteRun](#) in der AWS CLI Befehlsreferenz.

list-suite-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-suite-definitions`.

AWS CLI

Beispiel 1: Um die von Ihnen erstellten IoT Device Advisor-Testsuiten aufzulisten

Das folgende `list-suite-definitions` Beispiel listet bis zu 25 Device Advisor-Testsuiten auf, die Sie in AWS IoT erstellt haben. Wenn Sie über mehr als 25 Testsuiten verfügen, wird `nextToken ""` in der Ausgabe angezeigt. Sie können dieses "nextToken" verwenden, um die restlichen Testsuiten anzuzeigen, die Sie erstellt haben.

```
aws iotdeviceadvisor list-suite-definitions
```

Ausgabe:

```

{
  "suiteDefinitionInformationList": [
    {
      "suiteDefinitionId": "3hsn88h4p2g5",
      "suiteDefinitionName": "TestSuite1",
      "defaultDevices": [
        {
          "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"
        }
      ]
    }
  ]
}

```

```

    ],
    "intendedForQualification": false,
    "isLongDurationTest": false,
    "protocol": "MqttV3_1_1",
    "createdAt": "2022-11-17T14:15:56.830000-05:00"
  },
  {
    .....
  }
],
"nextToken": "nextTokenValue"
}

```

Beispiel 2: Um die IoT Device Advisor-Testsuiten aufzulisten, die Sie mit den angegebenen Einstellungen erstellt haben

Das folgende `list-suite-definitions` Beispiel listet Device Advisor-Testsuiten auf, die Sie in AWS IoT mit der angegebenen maximalen Ergebniszahl erstellt haben. Wenn Sie mehr Testsuiten als die maximale Anzahl haben, wird `"nextToken"` in der Ausgabe angezeigt. Wenn Sie `"nextToken"` haben, können Sie `"nextToken"` verwenden, um die von Ihnen erstellten Testsuiten anzuzeigen, die zuvor nicht angezeigt wurden.

```

aws iotdeviceadvisor list-suite-definitions \
  --max-result 1 \
  --next-token "nextTokenValue"

```

Ausgabe:

```

{
  "suiteDefinitionInformationList": [
    {
      "suiteDefinitionId": "ztlvb5aew4w4x",
      "suiteDefinitionName": "TestSuite2",
      "defaultDevices": [],
      "intendedForQualification": true,
      "isLongDurationTest": false,
      "protocol": "MqttV3_1_1",
      "createdAt": "2022-11-17T14:15:56.830000-05:00"
    }
  ],
  "nextToken": "nextTokenValue"
}

```

Weitere Informationen finden Sie [ListSuiteDefinitions](#) in der AWS API IoT-Referenz.

- API Einheiten finden Sie [ListSuiteDefinitions](#) unter AWS CLI Befehlsreferenz.

list-suite-runs

Das folgende Codebeispiel zeigt die Verwendung `list-suite-runs`.

AWS CLI

Beispiel 1: Um alle Informationen über den angegebenen Status der IoT Device Advisor-Testsuite aufzulisten

Im folgenden `list-suite-runs` Beispiel werden alle Informationen zum Ausführungsstatus einer Device Advisor-Testsuite mit der angegebenen Suite-Definition-ID aufgeführt. Wenn Sie mehr als 25 Testsuite-Läufe durchgeführt haben, wird `nextToken` "" in der Ausgabe angezeigt. Sie können dieses "nextToken" verwenden, um die restlichen Testsuite-Läufe anzuzeigen.

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id ztvb5aew4w4x
```

Ausgabe:

```
{  
  "suiteRunsList": [  
    {  
      "suiteDefinitionId": "ztvb5aew4w4x",  
      "suiteDefinitionVersion": "v1",  
      "suiteDefinitionName": "TestSuite",  
      "suiteRunId": "p6awv89nre6v",  
      "createdAt": "2022-12-01T16:33:14.212000-05:00",  
      "startedAt": "2022-12-01T16:33:15.710000-05:00",  
      "endAt": "2022-12-01T16:42:03.323000-05:00",  
      "status": "PASS",  
      "passed": 6,  
      "failed": 0  
    }  
  ]  
}
```

Beispiel 2: Um Informationen über den angegebenen IoT Device Advisor-Testsuite-Ausführungsstatus mit den angegebenen Einstellungen aufzulisten

Im folgenden `list-suite-runs` Beispiel werden Informationen zum Ausführungsstatus einer Device Advisor-Testsuite mit der angegebenen Suite-Definition-ID und der angegebenen maximalen Ergebniszahl aufgeführt. Wenn Sie mehr Testsuite-Läufe als die maximale Anzahl haben, wird `nextToken` "" in der Ausgabe angezeigt. Wenn Sie "nextToken" haben, können Sie "nextToken" verwenden, um die Testsuite-Läufe anzuzeigen, die zuvor nicht angezeigt wurden.

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id qqcsmtyyjam1 \  
  --max-result 1 \  
  --next-token "nextTokenValue"
```

Ausgabe:

```
{  
  "suiteRunsList": [  
    {  
      "suiteDefinitionId": "qqcsmtyyjam1",  
      "suiteDefinitionVersion": "v1",  
      "suiteDefinitionName": "MQTT connection",  
      "suiteRunId": "gz9vm2s6d2jy",  
      "createdAt": "2022-12-01T20:10:27.079000-05:00",  
      "startedAt": "2022-12-01T20:10:28.003000-05:00",  
      "endAt": "2022-12-01T20:10:45.084000-05:00",  
      "status": "STOPPED",  
      "passed": 0,  
      "failed": 0  
    }  
  ],  
  "nextToken": "nextTokenValue"  
}
```

Weitere Informationen finden Sie [ListSuiteRuns](#) in der AWS API IoT-Referenz.

- API-Einheiten finden Sie [ListSuiteRuns](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die an eine IoT Device Advisor-Ressource angehängten Tags aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags auf, die an eine Device Advisor-Ressource angehängt sind. Die Device Advisor-Ressource kann ein Suitedefinition-Arn oder ein Suiterun-Arn sein.

```
aws iotdeviceadvisor list-tags-for-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny
```

Ausgabe:

```
{  
  "tags": {  
    "TestTagKey": "TestTagValue"  
  }  
}
```

Weitere Informationen finden Sie [ListTagsForResource](#) in der AWS API IoT-Referenz und unter [Ressourcentypen, die von AWS IoT Core Device Advisor definiert wurden](#), in der Service Authorization Reference.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-suite-run

Das folgende Codebeispiel zeigt die Verwendung `start-suite-run`.

AWS CLI

Um eine IoT Device Advisor-Testsuite zu starten, führen Sie

Das folgende `start-suite-run` Beispiel listet die verfügbaren Widgets in Ihrem AWS Konto auf.

```
aws iotdeviceadvisor start-suite-run \  
  --suite-definition-id qqcsmtyyjabl \  
  --suite-definition-version v1 \  
  --suite-run-configuration '{"primaryDevice":{"thingArn": "arn:aws:iot:us-  
east-1:123456789012:thing/MyIoTThing", "certificateArn": "arn:aws:iot:us-  
east-1:123456789012:cert/certFile"}}'
```

Ausgabe:

```
{
  "suiteRunId": "pwmucgw71t9s",
  "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/
qqcsmtyyjabl/pwmucgw71k9s",
  "createdAt": "2022-12-02T15:43:05.581000-05:00"
}
```

Weitere Informationen finden Sie unter [Starten einer Testsuite-Ausführung](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [StartSuiteRun](#) in der AWS CLI Befehlsreferenz.

stop-suite-run

Das folgende Codebeispiel zeigt die Verwendung `stop-suite-run`.

AWS CLI

So beenden Sie eine IoT Device Advisor-Testsuite, die gerade ausgeführt wird

Im folgenden `stop-suite-run` Beispiel wird eine Device Advisor-Testsuite gestoppt, die derzeit mit der angegebenen Suite-Definition-ID und Suite-Run-ID ausgeführt wird.

```
aws iotdeviceadvisor stop-suite-run \
  --suite-definition-id qqcsmtyyjabl \
  --suite-run-id nzlfyhaa18oa
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Beenden einer Testsuite-Ausführung](#) im AWS IoT Core Developer Guide.

- API-Einheiten finden Sie [StopSuiteRun](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um die vorhandenen Tags einer IoT Device Advisor-Ressource hinzuzufügen und zu ändern

Im folgenden `tag-resource` Beispiel werden die vorhandenen Tags einer Device Advisor-Ressource um den angegebenen Ressourcen-ARN und die angegebenen Tags erweitert und geändert. Die Device Advisor-Ressource kann ein Suitedefinition-ARN oder ein Suiterun-ARN sein.

```
aws iotdeviceadvisor tag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tags '{"TagKey": "TagValue"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [TagResource](#) in der AWS API IoT-Referenz und unter [Ressourcentypen, die von AWS IoT Core Device Advisor definiert wurden](#), in der Service Authorization Reference.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

So entfernen Sie die vorhandenen Tags aus einer IoT Device Advisor-Ressource

Im folgenden `untag-resource` Beispiel werden die vorhandenen Tags mit dem angegebenen Ressourcen-ARN und Tag-Schlüssel aus einer Device Advisor-Ressource entfernt. Die Device Advisor-Ressource kann ein Suitedefinition-ARN oder ein Suiterun-ARN sein.

```
aws iotdeviceadvisor untag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tag-keys "TagKey"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UntagResource](#) in der AWS API IoT-Referenz und unter [Ressourcentypen, die von AWS IoT Core Device Advisor definiert wurden](#), in der Service Authorization Reference.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-suite-definition

Das folgende Codebeispiel zeigt die Verwendung `update-suite-definition`.

AWS CLI

Beispiel 1: So aktualisieren Sie eine IoT Device Advisor-Testsuite

Das folgende `update-suite-definition` Beispiel aktualisiert eine Device Advisor-Testsuite im AWS IoT mit der angegebenen Suite-Definition-ID und der Suite-Definitionskonfiguration.

```
aws iotdeviceadvisor update-suite-definition \
  --suite-definition-id 3hsn88h4p2g5 \
  --suite-definition-configuration '{ \
    "suiteDefinitionName": "TestSuiteName", \
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing"}], \
    "intendedForQualification": false, \
    "rootGroup": "{\"configuration\": {}, \"tests\": [{\"name\": \"MQTT Connect\", \
  \"configuration\": {\"EXECUTION_TIMEOUT\": 120}, \"tests\": [{\"name\": \"MQTT_Connect\", \
  \"configuration\": {}, \"test\": {\"id\": \"MQTT_Connect\", \"testCase\": null, \"version \
  \": \"0.0.0\"}]}]}]}", \
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

Ausgabe:

```
{
  "suiteDefinitionId": "3hsn88h4p2g5",
  "suiteDefinitionName": "TestSuiteName",
  "suiteDefinitionVersion": "v3",
  "createdAt": "2022-11-17T14:15:56.830000-05:00",
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

Beispiel 2: So aktualisieren Sie eine IoT Device Advisor-Qualifizierungstestsuite

Im folgenden `update-suite-definition` Beispiel wird eine Device Advisor-Qualifizierungstestsuite im AWS IoT mit der angegebenen Suite-Definition-ID und der Suite-Definitionskonfiguration aktualisiert.

```
aws iotdeviceadvisor update-suite-definition \
  --suite-definition-id txgsuolk2myj \
  --suite-definition-configuration '{
```

```
"suiteDefinitionName": "TestSuiteName", \  
"devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/  
MyIotThing"}], \  
"intendedForQualification": true, \  
"rootGroup": "", \  
"devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

Ausgabe:

```
{  
  "suiteDefinitionId": "txgsuolk2myj",  
  "suiteDefinitionName": "TestSuiteName",  
  "suiteDefinitionVersion": "v3",  
  "createdAt": "2022-11-17T14:15:56.830000-05:00",  
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"  
}
```

Weitere Informationen finden Sie [UpdateSuiteDefinition](#) in der AWS API IoT-Referenz.

- API Einzelheiten finden Sie [UpdateSuiteDefinition](#) unter AWS CLI Befehlsreferenz.

AWS IoT data Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT data.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-thing-shadow

Das folgende Codebeispiel zeigt die Verwendung `delete-thing-shadow`.

AWS CLI

Um das Schattendokument eines Geräts zu löschen

Im folgenden `delete-thing-shadow` Beispiel wird das gesamte Shadow-Dokument für das angegebene MyRPi Gerät gelöscht.

```
aws iot-data delete-thing-shadow \  
  --thing-name MyRPi \  
  "output.txt"
```

Der Befehl erzeugt keine Ausgabe auf dem Bildschirm, `output.txt` enthält jedoch Informationen, die die Version und den Zeitstempel des gelöschten Schattendokuments bestätigen.

```
{"version":2,"timestamp":1560270384}
```

Weitere Informationen finden Sie unter [Using Shadows](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteThingShadow](#) in der AWS CLI Befehlsreferenz.

get-thing-shadow

Das folgende Codebeispiel zeigt die Verwendung `get-thing-shadow`.

AWS CLI

Um ein Shadow-Dokument zu bekommen

Im folgenden `get-thing-shadow` Beispiel wird das Thing-Shadow-Dokument für das angegebene IoT-Ding abgerufen.

```
aws iot-data get-thing-shadow \  
  --thing-name MyRPi \  
  output.txt
```

Der Befehl erzeugt keine Ausgabe auf dem Display, aber im Folgenden wird der Inhalt von `output.txt` angezeigt:

```
{
```

```

"state":{
  "reported":{
    "moisture":"low"
  }
},
"metadata":{
  "reported":{
    "moisture":{
      "timestamp":1560269319
    }
  }
},
"version":1,"timestamp":1560269405
}

```

Weitere Informationen finden Sie unter [Device Shadow Service Data Flow](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetThingShadow](#) in der AWS CLI Befehlsreferenz.

update-thing-shadow

Das folgende Codebeispiel zeigt die Verwendung `update-thing-shadow`.

AWS CLI

Um einen Ding-Shadow zu aktualisieren

Im folgenden `update-thing-shadow` Beispiel wird der aktuelle Status des Geräteschattens für das angegebene Ding geändert und in der Datei `output.txt` gespeichert.

```

aws iot-data update-thing-shadow \
  --thing-name MyRPI \
  --payload '{"state":{"reported":{"moisture":"okay"}}}' \
  output.txt

```

Der Befehl erzeugt keine Ausgabe auf dem Display, aber im Folgenden wird der Inhalt von `output.txt` angezeigt:

```

{
  "state": {

```

```
    "reported": {
      "moisture": "okay"
    },
    "metadata": {
      "reported": {
        "moisture": {
          "timestamp": 1560270036
        }
      }
    },
    "version": 2,
    "timestamp": 1560270036
  }
```

Weitere Informationen finden Sie unter [Device Shadow Service Data Flow](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateThingShadow](#) in der AWS CLI Befehlsreferenz.

AWS IoT Events Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Events.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-put-message

Das folgende Codebeispiel zeigt die Verwendung `batch-put-message`.

AWS CLI

Um Nachrichten (Eingaben) an AWS IoT Events zu senden

Das folgende `batch-put-message` Beispiel sendet eine Reihe von Nachrichten an das AWS IoT Events Events-System. Jede Nachrichtennutzlast wird in die von Ihnen angegebene Eingabe (`inputName`) umgewandelt und in alle Detektoren aufgenommen, die diese Eingabe überwachen. Wenn mehrere Nachrichten gesendet werden, kann die Reihenfolge, in der die Nachrichten verarbeitet werden, nicht garantiert werden. Um die Bestellung zu garantieren, müssen Sie Nachrichten nacheinander senden und auf eine erfolgreiche Antwort warten.

```
aws iotevents-data batch-put-message \  
  --cli-input-json file://highPressureMessage.json
```

Inhalt von `highPressureMessage.json`:

```
{  
  "messages": [  
    {  
      "messageId": "00001",  
      "inputName": "PressureInput",  
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":  
80, \"temperature\": 39} }"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "BatchPutMessageErrorEntries": []  
}
```

Weitere Informationen finden Sie [BatchPutMessage](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [BatchPutMessage](#) unter AWS CLI Befehlsreferenz.

batch-update-detector

Das folgende Codebeispiel zeigt die Verwendung `batch-update-detector`.

AWS CLI

Um einen Detektor (Instanz) zu aktualisieren

Das folgende `batch-update-detector` Beispiel aktualisiert den Status, die Variablenwerte und die Timer-Einstellungen eines oder mehrerer Detektoren (Instanzen) eines bestimmten Meldermodells.

```
aws iotevents-data batch-update-detector \  
  --cli-input-json file://budFulton-A32.json
```

Inhalt von `budFulton-A32.json`:

```
{  
  "detectors": [  
    {  
      "messageId": "00001",  
      "detectorModelName": "motorDetectorModel",  
      "keyValue": "Fulton-A32",  
      "state": {  
        "stateName": "Normal",  
        "variables": [  
          {  
            "name": "pressureThresholdBreach",  
            "value": "0"  
          }  
        ],  
        "timers": [  
        ]  
      }  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "batchUpdateDetectorErrorEntries": []  
}
```

Weitere Informationen finden Sie [BatchUpdateDetector](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [BatchUpdateDetector](#) unter AWS CLI Befehlsreferenz.

create-detector-model

Das folgende Codebeispiel zeigt die Verwendung `create-detector-model`.

AWS CLI

Um ein Detektormodell zu erstellen

Im folgenden `create-detector-model` Beispiel wird ein Detektormodell erstellt, dessen Konfiguration durch eine Parameterdatei spezifiziert wird.

```
aws iotevents create-detector-model \  
  --cli-input-json file://motorDetectorModel.json
```

Inhalt von `motorDetectorModel.json`:

```
{  
  "detectorModelName": "motorDetectorModel",  
  "detectorModelDefinition": {  
    "states": [  
      {  
        "stateName": "Normal",  
        "onEnter": {  
          "events": [  
            {  
              "eventName": "init",  
              "condition": "true",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreach",  
                    "value": "0"  
                  }  
                }  
              ]  
            }  
          ]  
        },  
        "onInput": {  
          "transitionEvents": [  
            {  
              "eventName": "Overpressurized",
```

```

        "condition": "$input.PressureInput.sensorData.pressure
    > 70",
        "actions": [
            {
                "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached + 3"
                }
            }
        ],
        "nextState": "Dangerous"
    }
]
},
{
    "stateName": "Dangerous",
    "onEnter": {
        "events": [
            {
                "eventName": "Pressure Threshold Breached",
                "condition": "$variable.pressureThresholdBreached >
1",
                "actions": [
                    {
                        "sns": {
                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                        }
                    }
                ]
            }
        ]
    },
    "onInput": {
        "events": [
            {
                "eventName": "Overpressurized",
                "condition": "$input.PressureInput.sensorData.pressure
    > 70",
                "actions": [
                    {
                        "setVariable": {

```

```

        "variableName": "pressureThresholdBreached",
        "value": "3"
    }
  ]
},
{
  "eventName": "Pressure Okay",
  "condition": "$input.PressureInput.sensorData.pressure
<= 70",
  "actions": [
    {
      "setVariable": {
        "variableName": "pressureThresholdBreached",
        "value":
"$variable.pressureThresholdBreached - 1"
      }
    ]
  }
],
"transitionEvents": [
  {
    "eventName": "BackToNormal",
    "condition": "$input.PressureInput.sensorData.pressure
<= 70 && $variable.pressureThresholdBreached <= 1",
    "nextState": "Normal"
  }
],
"onExit": {
  "events": [
    {
      "eventName": "Normal Pressure Restored",
      "condition": "true",
      "actions": [
        {
          "sns": {
            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
          }
        ]
      }
    ]
  }
}

```

```

    ]
  }
},
"initialStateName": "Normal"
},
"key": "motorid",
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

Ausgabe:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560796816.077,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560796816.077,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
}

```

Weitere Informationen finden Sie [CreateDetectorModel](#) in der API Referenz zu AWS IoT Events.

- API Einzelheiten finden Sie [CreateDetectorModel](#) unter AWS CLI Befehlsreferenz.

create-input

Das folgende Codebeispiel zeigt die Verwendung `create-input`.

AWS CLI

Um eine Eingabe zu erstellen

Das folgende `create-input` Beispiel erstellt eine Eingabe.

```

aws iotevents create-input \
  --cli-input-json file://pressureInput.json

```

Inhalt von `pressureInput.json`:

```
{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}
```

Ausgabe:

```
{
  "inputConfiguration": {
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1560795312.542,
    "creationTime": 1560795312.542,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
  }
}
```

Weitere Informationen finden Sie [CreateInput](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [CreateInput](#) unter AWS CLI Befehlsreferenz.

delete-detector-model

Das folgende Codebeispiel zeigt die Verwendung `delete-detector-model`.

AWS CLI

Um ein Detektormodell zu löschen

Im folgenden `delete-detector-model` Beispiel wird das angegebene Detektormodell gelöscht. Alle aktiven Instanzen des Detektormodells werden ebenfalls gelöscht.

```
aws iotevents delete-detector-model \
```

```
--detector-model-name motorDetectorModel
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteDetectorModel](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [DeleteDetectorModel](#) unter AWS CLI Befehlsreferenz.

delete-input

Das folgende Codebeispiel zeigt die Verwendung `delete-input`.

AWS CLI

Um eine Eingabe zu löschen

Im folgenden `delete-input` Beispiel wird die angegebene Eingabe gelöscht.

```
aws iotevents delete-input \  
--input-name PressureInput
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteInput](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [DeleteInput](#) unter AWS CLI Befehlsreferenz.

describe-detector-model

Das folgende Codebeispiel zeigt die Verwendung `describe-detector-model`.

AWS CLI

Um Informationen über ein Detektormodell zu erhalten

Im folgenden `describe-detector-model` Beispiel werden Details für das angegebene Detektormodell angezeigt. Da der `version` Parameter nicht angegeben ist, werden Informationen über die neueste Version zurückgegeben.

```
aws iotevents describe-detector-model \  
--detector-model-name motorDetectorModel
```


Ausgabe:

```

{
  "detectorModel": {
    "detectorModelConfiguration": {
      "status": "ACTIVE",
      "lastUpdateTime": 1560796816.077,
      "roleArn": "arn:aws:iam:123456789012:role/IoTEventsRole",
      "creationTime": 1560796816.077,
      "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
      "key": "motorid",
      "detectorModelName": "motorDetectorModel",
      "detectorModelVersion": "1"
    },
    "detectorModelDefinition": {
      "states": [
        {
          "onInput": {
            "transitionEvents": [
              {
                "eventName": "Overpressurized",
                "actions": [
                  {
                    "setVariable": {
                      "variableName":
"pressureThresholdBreached",
                      "value":
"$variable.pressureThresholdBreached + 3"
                    }
                  ]
                },
                "condition":
"$input.PressureInput.sensorData.pressure > 70",
                "nextState": "Dangerous"
              }
            ],
            "events": []
          },
          "stateName": "Normal",
          "onEnter": {
            "events": [
              {
                "eventName": "init",

```

```

        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreach",
                    "value": "0"
                }
            }
        ],
        "condition": "true"
    }
]
},
"onExit": {
    "events": []
}
},
{
    "onInput": {
        "transitionEvents": [
            {
                "eventName": "BackToNormal",
                "actions": [],
                "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreach <= 1",
                "nextState": "Normal"
            }
        ],
        "events": [
            {
                "eventName": "Overpressurized",
                "actions": [
                    {
                        "setVariable": {
                            "variableName":
"pressureThresholdBreach",
                            "value": "3"
                        }
                    }
                ],
                "condition":
"$input.PressureInput.sensorData.pressure > 70"
            }
        ],

```

```

        {
            "eventName": "Pressure Okay",
            "actions": [
                {
                    "setVariable": {
                        "variableName":
"pressureThresholdBreach",
                        "value":
"$variable.pressureThresholdBreach - 1"
                    }
                }
            ],
            "condition":
"$input.PressureInput.sensorData.pressure <= 70"
        }
    ],
    "stateName": "Dangerous",
    "onEnter": {
        "events": [
            {
                "eventName": "Pressure Threshold Breached",
                "actions": [
                    {
                        "sns": {
                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                        }
                    }
                ]
            },
            {
                "condition": "$variable.pressureThresholdBreach >
1"
            }
        ]
    },
    "onExit": {
        "events": [
            {
                "eventName": "Normal Pressure Restored",
                "actions": [
                    {
                        "sns": {
                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                        }
                    }
                ]
            }
        ]
    }
}

```

```

    }
    ],
    "condition": "true"
  }
]
}
"initialStateName": "Normal"
}
}
}

```

Weitere Informationen finden Sie [DescribeDetectorModel](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [DescribeDetectorModel](#) unter AWS CLI Befehlsreferenz.

describe-detector

Das folgende Codebeispiel zeigt die Verwendung `describe-detector`.

AWS CLI

Um Informationen über einen Detektor (Instanz) zu erhalten.

Im folgenden `describe-detector` Beispiel werden Details für den angegebenen Detektor (Instanz) angezeigt.

```

aws iotevents-data describe-detector \
  --detector-model-name motorDetectorModel \
  --key-value "Fulton-A32"

```

Ausgabe:

```

{
  "detector": {
    "lastUpdateTime": 1560797852.776,
    "creationTime": 1560797852.775,
    "state": {
      "variables": [
        {

```

```

        "name": "pressureThresholdBreached",
        "value": "3"
      }
    ],
    "stateName": "Dangerous",
    "timers": []
  },
  "keyValue": "Fulton-A32",
  "detectorModelName": "motorDetectorModel",
  "detectorModelVersion": "1"
}
}

```

Weitere Informationen finden Sie [DescribeDetector](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [DescribeDetector](#) unter AWS CLI Befehlsreferenz.

describe-input

Das folgende Codebeispiel zeigt die Verwendung `describe-input`.

AWS CLI

Um Informationen über eine Eingabe zu erhalten

Im folgenden `describe-input` Beispiel werden Details für die angegebene Eingabe angezeigt.

```

aws iotevents describe-input \
  --input-name PressureInput

```

Ausgabe:

```

{
  "input": {
    "inputConfiguration": {
      "status": "ACTIVE",
      "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/
PressureInput",
      "lastUpdateTime": 1560795312.542,
      "creationTime": 1560795312.542,
      "inputName": "PressureInput",
      "inputDescription": "Pressure readings from a motor"
    }
  }
}

```

```
    },
    "inputDefinition": {
      "attributes": [
        {
          "jsonPath": "sensorData.pressure"
        },
        {
          "jsonPath": "motorid"
        }
      ]
    }
  }
}
```

Weitere Informationen finden Sie [BeschreibenInput](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [BeschreibenInput](#) unter AWS CLI Befehlsreferenz.

describe-logging-options

Das folgende Codebeispiel zeigt die Verwendung `describe-logging-options`.

AWS CLI

Um Informationen zu den Protokollierungseinstellungen zu erhalten

Im folgenden `describe-logging-options` Beispiel werden die aktuellen Einstellungen der Protokollierungsoptionen für AWS IoT Events abgerufen.

```
aws iotevents describe-logging-options
```

Ausgabe:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "enabled": false,
    "level": "ERROR"
  }
}
```

Weitere Informationen finden Sie [DescribeLoggingOptions](#) in der APIReferenz zu AWS IoT Events.

- API-Einheiten finden Sie [DescribeLoggingOptions](#) unter AWS CLI Befehlsreferenz.

list-detector-model-versions

Das folgende Codebeispiel zeigt die Verwendung `list-detector-model-versions`.

AWS CLI

Um Informationen über Versionen eines Detektormodells zu erhalten

Das folgende `list-detector-model-versions` Beispiel listet alle Versionen eines Detektormodells auf. Es werden nur die Metadaten für die jeweilige Detektormodellversion zurückgegeben.

```
aws iotevents list-detector-model-versions \  
  --detector-model-name motorDetectorModel
```

Ausgabe:

```
{  
  "detectorModelVersionSummaries": [  
    {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-  
west-2:123456789012:detectorModel/motorDetectorModel",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ListDetectorModelVersions](#) in der API-Referenz zu AWS IoT Events.

- API-Einheiten finden Sie [ListDetectorModelVersions](#) unter AWS CLI Befehlsreferenz.

list-detector-models

Das folgende Codebeispiel zeigt die Verwendung `list-detector-models`.

AWS CLI

Um eine Liste Ihrer Detektormodelle zu erhalten

Das folgende `list-detector-models` Beispiel listet die von Ihnen erstellten Meldermodelle auf. Es werden nur die Metadaten für das jeweilige Detektormodell zurückgegeben.

```
aws iotevents list-detector-models
```

Ausgabe:

```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

Weitere Informationen finden Sie [ListDetectorModels](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [ListDetectorModels](#) unter AWS CLI Befehlsreferenz.

list-detectors

Das folgende Codebeispiel zeigt die Verwendung `list-detectors`.

AWS CLI

Um eine Liste von Detektoren für ein Detektormodell zu erhalten

Das folgende `list-detectors` Beispiel listet die Detektoren (die Instanzen eines Meldermodells) in Ihrem Konto auf.

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

Ausgabe:

```
{
```



```
"detectorSummaries": [
  {
    "lastUpdateTime": 1558129925.2,
    "creationTime": 1552073155.527,
    "state": {
      "stateName": "Normal"
    },
    "keyValue": "Fulton-A32",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
]
```

Weitere Informationen finden Sie [ListDetectors](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [ListDetectors](#) unter AWS CLI Befehlsreferenz.

list-inputs

Das folgende Codebeispiel zeigt die Verwendung `list-inputs`.

AWS CLI

Um Eingaben aufzulisten

Das folgende `list-inputs` Beispiel listet die Eingaben auf, die Sie in Ihrem Konto erstellt haben.

```
aws iotevents list-inputs
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  {
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1551742986.768,
    "creationTime": 1551742986.768,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
  }
}
```

```
}
```

Weitere Informationen finden Sie [ListInputs](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [ListInputs](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags aufzulisten, die einer Ressource zugewiesen sind.

Das folgende `list-tags-for-resource` Beispiel listet die Namen und Werte der Tag-Schlüssel auf, die Sie der Ressource zugewiesen haben.

```
aws iotevents list-tags-for-resource \
  --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Ausgabe:

```
{
  "tags": [
    {
      "value": "motor",
      "key": "deviceType"
    }
  ]
}
```

Weitere Informationen finden Sie [ListTagsForResource](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

put-logging-options

Das folgende Codebeispiel zeigt die Verwendung `put-logging-options`.

AWS CLI

Um Protokollierungsoptionen festzulegen

Im folgenden `put-logging-options` Beispiel werden die Protokollierungsoptionen für AWS IoT Events festgelegt oder aktualisiert. Wenn Sie den Wert eines `loggingOptions`` field, it can take up to one minute for the change to take effect. Also, if you change the policy attached to the role you specified in the `roleArn` Felds aktualisieren (z. B. um eine ungültige Richtlinie zu korrigieren), kann es bis zu fünf Minuten dauern, bis diese Änderung wirksam wird.

```
aws iotevents put-logging-options \  
--cli-input-json file://logging-options.json
```

Inhalt von `logging-options.json`:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",  
    "enabled": true,  
    "detectorDebugOptions": [  
      {  
        "detectorModelName": "motorDetectorModel",  
        "keyValue": "Fulton-A32"  
      }  
    ]  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [PutLoggingOptions](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [PutLoggingOptions](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer Ressource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel wird das der angegebenen Ressource zugeordnete Tag hinzugefügt oder geändert (falls der Schlüssel `deviceType` bereits vorhanden ist).

```
aws iotevents tag-resource \  
  --cli-input-json file://pressureInput.tag.json
```

Inhalt von `pressureInput.tag.json`:

```
{  
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
  "tags": [  
    {  
      "key": "deviceType",  
      "value": "motor"  
    }  
  ]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [TagResource](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem angegebenen Schlüsselnamen aus der angegebenen Ressource entfernt.

```
aws iotevents untag-resource \  
  --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \  
  --tagkeys deviceType
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UntagResource](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-detector-model

Das folgende Codebeispiel zeigt die Verwendung `update-detector-model`.

AWS CLI

Um ein Detektormodell zu aktualisieren

Im folgenden `update-detector-model` Beispiel wird das angegebene Detektormodell aktualisiert. Von der vorherigen Version erzeugte Detektoren (Instanzen) werden gelöscht und dann neu erstellt, sobald neue Eingaben eintreffen.

```
aws iotevents update-detector-model \  
  --cli-input-json file://motorDetectorModel.update.json
```

Inhalt von `motorDetectorModel.update.json`:

```
{  
  "detectorModelName": "motorDetectorModel",  
  "detectorModelDefinition": {  
    "states": [  
      {  
        "stateName": "Normal",  
        "onEnter": {  
          "events": [  
            {  
              "eventName": "init",  
              "condition": "true",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreached",  
                    "value": "0"  
                  }  
                }  
              ]  
            }  
          ]  
        },  
        "onInput": {  
          "transitionEvents": [  
            {  
              "eventName": "Overpressurized",
```

```

    "condition": "$input.PressureInput.sensorData.pressure >
70",
    "actions": [
      {
        "setVariable": {
          "variableName": "pressureThresholdBreach",
          "value":
"$variable.pressureThresholdBreach + 3"
        }
      },
      {
        "nextState": "Dangerous"
      }
    ]
  },
  {
    "stateName": "Dangerous",
    "onEnter": {
      "events": [
        {
          "eventName": "Pressure Threshold Breached",
          "condition": "$variable.pressureThresholdBreach > 1",
          "actions": [
            {
              "sns": {
                "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
              }
            }
          ]
        }
      ]
    },
    "onInput": {
      "events": [
        {
          "eventName": "Overpressurized",
          "condition": "$input.PressureInput.sensorData.pressure >
70",
          "actions": [
            {
              "setVariable": {
                "variableName": "pressureThresholdBreach",

```

```

        "value": "3"
      }
    }
  ],
},
{
  "eventName": "Pressure Okay",
  "condition": "$input.PressureInput.sensorData.pressure
<= 70",
  "actions": [
    {
      "setVariable": {
        "variableName": "pressureThresholdBreached",
        "value":
"$variable.pressureThresholdBreached - 1"
      }
    }
  ]
},
],
"transitionEvents": [
  {
    "eventName": "BackToNormal",
    "condition": "$input.PressureInput.sensorData.pressure
<= 70 && $variable.pressureThresholdBreached <= 1",
    "nextState": "Normal"
  }
]
},
"onExit": {
  "events": [
    {
      "eventName": "Normal Pressure Restored",
      "condition": "true",
      "actions": [
        {
          "sns": {
            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
          }
        }
      ]
    }
  ]
}
]

```

```

    }
  },
  ],
  "initialStateName": "Normal"
},
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

Ausgabe:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560799387.719,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560799387.719,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "2"
  }
}

```

Weitere Informationen finden Sie [UpdateDetectorModel](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [UpdateDetectorModel](#) unter AWS CLI Befehlsreferenz.

update-input

Das folgende Codebeispiel zeigt die Verwendung `update-input`.

AWS CLI

Um eine Eingabe zu aktualisieren

Im folgenden `update-input` Beispiel wird die angegebene Eingabe mit einer neuen Beschreibung und Definition aktualisiert.

```

aws iotevents update-input \
  --cli-input-json file://pressureInput.json

```

Inhalt von `pressureInput.json`:


```
{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}
```

Ausgabe:

```
{
  "inputConfiguration": {
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1560795976.458,
    "creationTime": 1560795312.542,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
  }
}
```

Weitere Informationen finden Sie [UpdateInput](#) in der APIReferenz zu AWS IoT Events.

- APIEinzelheiten finden Sie [UpdateInput](#) unter AWS CLI Befehlsreferenz.

AWS IoT Events-Data Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Events-Data.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-put-message

Das folgende Codebeispiel zeigt die Verwendung `batch-put-message`.

AWS CLI

Um Nachrichten (Eingaben) an AWS IoT Events zu senden

Das folgende `batch-put-message` Beispiel sendet eine Reihe von Nachrichten an das AWS IoT Events Events-System. Jede Nachrichtennutzlast wird in die von Ihnen angegebene Eingabe (`inputName`) umgewandelt und in alle Detektoren aufgenommen, die diese Eingabe überwachen. Wenn mehrere Nachrichten gesendet werden, kann die Reihenfolge, in der die Nachrichten verarbeitet werden, nicht garantiert werden. Um die Bestellung zu garantieren, müssen Sie Nachrichten nacheinander senden und auf eine erfolgreiche Antwort warten.

```
aws iotevents-data batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://highPressureMessage.json
```

Inhalt von `highPressureMessage.json`:

```
{  
  "messages": [  
    {  
      "messageId": "00001",  
      "inputName": "PressureInput",  
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":  
80, \"temperature\": 39} }"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "BatchPutMessageErrorEntries": []  
}
```

Weitere Informationen finden Sie [BatchPutMessage](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [BatchPutMessage](#) in der AWS CLI Befehlsreferenz.

batch-update-detector

Das folgende Codebeispiel zeigt die Verwendung `batch-update-detector`.

AWS CLI

Um einen Detektor (Instanz) zu aktualisieren

Das folgende `batch-update-detector` Beispiel aktualisiert den Status, die Variablenwerte und die Timer-Einstellungen eines oder mehrerer Detektoren (Instanzen) eines bestimmten Meldermodells.

```
aws iotevents-data batch-update-detector \  
  --cli-input-json file://budFulton-A32.json
```

Inhalt von `budFulton-A32.json`:

```
{  
  "detectors": [  
    {  
      "messageId": "00001",  
      "detectorModelName": "motorDetectorModel",  
      "keyValue": "Fulton-A32",  
      "state": {  
        "stateName": "Normal",  
        "variables": [  
          {  
            "name": "pressureThresholdBreach",  
            "value": "0"  
          }  
        ],  
        "timers": [  
        ]  
      }  
    }  
  ]  
}
```

Ausgabe:

```
{
  "batchUpdateDetectorErrorEntries": []
}
```

Weitere Informationen finden Sie [BatchUpdateDetector](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [BatchUpdateDetector](#) in der AWS CLI Befehlsreferenz.

create-detector-model

Das folgende Codebeispiel zeigt die Verwendung `create-detector-model`.

AWS CLI

Um ein Detektormodell zu erstellen

Im folgenden `create-detector-model` Beispiel wird ein Detektormodell erstellt.

```
aws iotevents create-detector-model \
  --cli-input-json file://motorDetectorModel.json
```

Inhalt von `motorDetectorModel.json`:

```
{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "0"
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

```

    ]
  },
  "onInput": {
    "transitionEvents": [
      {
        "eventName": "Overpressurized",
        "condition": "$input.PressureInput.sensorData.pressure
> 70",
        "actions": [
          {
            "setVariable": {
              "variableName": "pressureThresholdBreach",
              "value":
"$variable.pressureThresholdBreach + 3"
            }
          }
        ],
        "nextState": "Dangerous"
      }
    ]
  }
},
{
  "stateName": "Dangerous",
  "onEnter": {
    "events": [
      {
        "eventName": "Pressure Threshold Breached",
        "condition": "$variable.pressureThresholdBreach >
1",
        "actions": [
          {
            "sns": {
              "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
            }
          }
        ]
      }
    ]
  },
  "onInput": {
    "events": [
      {

```

```

        "eventName": "Overpressurized",
        "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
        "actions": [
            {
                "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "3"
                }
            }
        ]
    },
    {
        "eventName": "Pressure Okay",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70",
        "actions": [
            {
                "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached - 1"
                }
            }
        ]
    }
],
"transitionEvents": [
    {
        "eventName": "BackToNormal",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 &amp; $variable.pressureThresholdBreached &lt;= 1",
        "nextState": "Normal"
    }
]
},
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "condition": "true",
            "actions": [
                {
                    "sns": {

```

```

        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
    }
}
]
}
],
"initialStateName": "Normal"
},
"key": "motorid",
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

Ausgabe:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560796816.077,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560796816.077,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
}

```

Weitere Informationen finden Sie [CreateDetectorModel](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [CreateDetectorModel](#) in der AWS CLI Befehlsreferenz.

create-input

Das folgende Codebeispiel zeigt die Verwendung `create-input`.

AWS CLI

Um eine Eingabe zu erstellen

Das folgende `create-input` Beispiel erstellt eine Eingabe.

```
aws iotevents create-input \  
  --cli-input-json file://pressureInput.json
```

Inhalt von `pressureInput.json`:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "inputConfiguration": {  
    "status": "ACTIVE",  
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
    "lastUpdateTime": 1560795312.542,  
    "creationTime": 1560795312.542,  
    "inputName": "PressureInput",  
    "inputDescription": "Pressure readings from a motor"  
  }  
}
```

Weitere Informationen finden Sie [CreateInput](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [CreateInput](#) in der AWS CLI Befehlsreferenz.

delete-detector-model

Das folgende Codebeispiel zeigt die Verwendung `delete-detector-model`.

AWS CLI

Um ein Detektormodell zu löschen

Im folgenden `delete-detector-model` Beispiel wird ein Detektormodell gelöscht. Alle aktiven Instanzen des Detektormodells werden ebenfalls gelöscht.

```
aws iotevents delete-detector-model \  
  --detector-model-name motorDetectorModel*
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteDetectorModel](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [DeleteDetectorModel](#) in der AWS CLI Befehlsreferenz.

delete-input

Das folgende Codebeispiel zeigt die Verwendung `delete-input`.

AWS CLI

Um eine Eingabe zu löschen

Das folgende `delete-input` Beispiel löscht eine Eingabe.

```
aws iotevents delete-input \  
  --input-name PressureInput
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteInput](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [DeleteInput](#) in der AWS CLI Befehlsreferenz.

describe-detector-model

Das folgende Codebeispiel zeigt die Verwendung `describe-detector-model`.

AWS CLI

Um Informationen über ein Detektormodell zu erhalten

Das folgende `describe-detector-model` Beispiel beschreibt ein Detektormodell. Wenn der `version` Parameter nicht angegeben ist, gibt der Befehl Informationen über die neueste Version zurück.

```
aws iotevents describe-detector-model \  
--detector-model-name motorDetectorModel
```

Ausgabe:

```
{  
  "detectorModel": {  
    "detectorModelConfiguration": {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam:123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-  
west-2:123456789012:detectorModel/motorDetectorModel",  
      "key": "motorid",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    },  
    "detectorModelDefinition": {  
      "states": [  
        {  
          "onInput": {  
            "transitionEvents": [  
              {  
                "eventName": "Overpressurized",  
                "actions": [  
                  {  
                    "setVariable": {  
                      "variableName":  
"pressureThresholdBreached",  
                      "value":  
"$variable.pressureThresholdBreached + 3"  
                    }  
                  }  
                ],  
                "condition":  
"$input.PressureInput.sensorData.pressure > 70",  
                "nextState": "Dangerous"  
              }  
            ],  
            "events": []  
          },  
          "stateName": "Normal",  
        }  
      ]  
    }  
  }  
}
```

```

        "onEnter": {
            "events": [
                {
                    "eventName": "init",
                    "actions": [
                        {
                            "setVariable": {
                                "variableName":
"pressureThresholdBreach",
                                "value": "0"
                            }
                        }
                    ],
                    "condition": "true"
                }
            ]
        },
        "onExit": {
            "events": []
        }
    },
    {
        "onInput": {
            "transitionEvents": [
                {
                    "eventName": "BackToNormal",
                    "actions": [],
                    "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreach <= 1",
                    "nextState": "Normal"
                }
            ],
            "events": [
                {
                    "eventName": "Overpressurized",
                    "actions": [
                        {
                            "setVariable": {
                                "variableName":
"pressureThresholdBreach",
                                "value": "3"
                            }
                        }
                    ]
                }
            ]
        }
    }
}

```

```

        ],
        "condition":
"$input.PressureInput.sensorData.pressure > 70"
    },
    {
        "eventName": "Pressure Okay",
        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached - 1"
                }
            }
        ],
        "condition":
"$input.PressureInput.sensorData.pressure <= 70"
    }
]
},
"stateName": "Dangerous",
"onEnter": {
    "events": [
        {
            "eventName": "Pressure Threshold Breached",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                    }
                }
            ]
        },
        {
            "condition": "$variable.pressureThresholdBreached >
1"
        }
    ]
},
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "actions": [

```



```

    "creationTime": 1560797852.775,
    "state": {
      "variables": [
        {
          "name": "pressureThresholdBreached",
          "value": "3"
        }
      ],
      "stateName": "Dangerous",
      "timers": []
    },
    "keyValue": "Fulton-A32",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
}

```

Weitere Informationen finden Sie [DescribeDetector](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [DescribeDetector](#) in der AWS CLI Befehlsreferenz.

describe-input

Das folgende Codebeispiel zeigt die Verwendung `describe-input`.

AWS CLI

Um Informationen über eine Eingabe zu erhalten

Im folgenden `describe-input` Beispiel werden die Details einer Eingabe abgerufen.

```

aws iotevents describe-input \
  --input-name PressureInput

```

Ausgabe:

```

{
  "input": {
    "inputConfiguration": {
      "status": "ACTIVE",
      "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/
PressureInput",

```

```
    "lastUpdateTime": 1560795312.542,
    "creationTime": 1560795312.542,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
  },
  "inputDefinition": {
    "attributes": [
      {
        "jsonPath": "sensorData.pressure"
      },
      {
        "jsonPath": "motorid"
      }
    ]
  }
}
```

Weitere Informationen finden Sie [DescribeInput](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [DescribeInput](#) in der AWS CLI Befehlsreferenz.

describe-logging-options

Das folgende Codebeispiel zeigt die Verwendung `describe-logging-options`.

AWS CLI

Um Informationen zu den Protokollierungseinstellungen zu erhalten

Im folgenden `describe-logging-options` Beispiel werden die aktuellen Protokollierungsoptionen für AWS IoT Events abgerufen.

```
aws iotevents describe-logging-options
```

Ausgabe:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "enabled": false,
    "level": "ERROR"
  }
}
```

```
}  
}
```

Weitere Informationen finden Sie [DescribeLoggingOptions](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [DescribeLoggingOptions](#) in der AWS CLI Befehlsreferenz.

list-detector-model-versions

Das folgende Codebeispiel zeigt die Verwendung `list-detector-model-versions`.

AWS CLI

Um Informationen über Versionen eines Detektormodells zu erhalten

Das folgende `list-detector-model-versions` Beispiel listet alle Versionen eines Detektormodells auf. Es werden nur die Metadaten für die jeweilige Detektormodellversion zurückgegeben.

```
aws iotevents list-detector-model-versions \  
  --detector-model-name motorDetectorModel
```

Ausgabe:

```
{  
  "detectorModelVersionSummaries": [  
    {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-  
west-2:123456789012:detectorModel/motorDetectorModel",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ListDetectorModelVersions](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [ListDetectorModelVersions](#) in der AWS CLI Befehlsreferenz.

list-detector-models

Das folgende Codebeispiel zeigt die Verwendung `list-detector-models`.

AWS CLI

Um eine Liste Ihrer Detektormodelle zu erhalten

Das folgende `list-detector-models` Beispiel listet die Meldermodelle auf, die Sie erstellt haben. Es werden nur die Metadaten für das jeweilige Detektormodell zurückgegeben.

```
aws iotevents list-detector-models
```

Ausgabe:

```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

Weitere Informationen finden Sie [ListDetectorModels](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [ListDetectorModels](#) in der AWS CLI Befehlsreferenz.

list-detectors

Das folgende Codebeispiel zeigt die Verwendung `list-detectors`.

AWS CLI

Um eine Liste von Detektoren für ein Detektormodell zu erhalten

Das folgende `list-detectors` Beispiel listet Detektoren (die Instanzen eines Detektormodells) auf.

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

Ausgabe:

```
{
  "detectorSummaries": [
    {
      "lastUpdateTime": 1558129925.2,
      "creationTime": 1552073155.527,
      "state": {
        "stateName": "Normal"
      },
      "keyValue": "Fulton-A32",
      "detectorModelName": "motorDetectorModel",
      "detectorModelVersion": "1"
    }
  ]
}
```

Weitere Informationen finden Sie [ListDetectors](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [ListDetectors](#) in der AWS CLI Befehlsreferenz.

list-inputs

Das folgende Codebeispiel zeigt die Verwendung `list-inputs`.

AWS CLI

Um Eingaben aufzulisten

Das folgende `list-inputs` Beispiel listet die Eingaben auf, die Sie erstellt haben.

```
aws iotevents list-inputs
```

Ausgabe:

```
{
  "status": "ACTIVE",
  "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
}
```

```
"lastUpdateTime": 1551742986.768,  
"creationTime": 1551742986.768,  
"inputName": "PressureInput",  
"inputDescription": "Pressure readings from a motor"  
}
```

Weitere Informationen finden Sie [ListInputs](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [ListInputs](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags aufzulisten, die einer Ressource zugewiesen sind

Im folgenden `list-tags-for-resource` Beispiel werden die Tags (Metadaten) aufgeführt, die Sie der Ressource zugewiesen haben.

```
aws iotevents list-tags-for-resource \  
--resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Ausgabe:

```
{  
  "tags": [  
    {  
      "value": "motor",  
      "key": "deviceType"  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ListTagsForResource](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-logging-options

Das folgende Codebeispiel zeigt die Verwendung `put-logging-options`.

AWS CLI

So legen Sie Protokollierungsoptionen fest

Im folgenden `list-tags-for-resource` Beispiel werden die Protokollierungsoptionen für AWS IoT Events festgelegt oder aktualisiert. Wenn Sie den Wert eines `loggingOptions` Felds aktualisieren, dauert es bis zu einer Minute, bis die Änderung wirksam wird. Wenn Sie außerdem die Richtlinie ändern, die der Rolle zugeordnet ist, die Sie im `roleArn` Feld angegeben haben (z. B. um eine ungültige Richtlinie zu korrigieren), dauert es bis zu fünf Minuten, bis diese Änderung wirksam wird.

```
aws iotevents put-logging-options \  
  --cli-input-json file://logging-options.json
```

Inhalt von `logging-options.json`:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",  
    "enabled": true,  
    "detectorDebugOptions": [  
      {  
        "detectorModelName": "motorDetectorModel",  
        "keyValue": "Fulton-A32"  
      }  
    ]  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [PutLoggingOptions](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [PutLoggingOptions](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer Ressource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel werden die Tags der angegebenen Ressource hinzugefügt oder geändert. Tags sind Metadaten, die zur Verwaltung einer Ressource verwendet werden können.

```
aws iotevents tag-resource \  
  --cli-input-json file://pressureInput.tag.json
```

Inhalt von `pressureInput.tag.json`:

```
{  
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
  "tags": [  
    {  
      "key": "deviceType",  
      "value": "motor"  
    }  
  ]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [TagResource](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden die angegebenen Tags aus der Ressource entfernt.

```
aws iotevents untag-resource \  
  --cli-input-json file://pressureInput.untag.json
```

Inhalt von `pressureInput.untag.json`:

```
{
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "tagKeys": [
    "deviceType"
  ]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [UntagResource](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-detector-model

Das folgende Codebeispiel zeigt die Verwendung `update-detector-model`.

AWS CLI

Um ein Detektormodell zu aktualisieren

Im folgenden `update-detector-model` Beispiel wird ein Detektormodell aktualisiert. Von der vorherigen Version erzeugte Detektoren (Instanzen) werden gelöscht und dann neu erstellt, sobald neue Eingaben eintreffen.

```
aws iotevents update-detector-model \
  --cli-input-json file://motorDetectorModel.update.json
```

Inhalt von `.update.json` `motorDetectorModel`:

```
{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
```

```
    "actions": [
      {
        "setVariable": {
          "variableName": "pressureThresholdBreach",
          "value": "0"
        }
      }
    ]
  },
  "onInput": {
    "transitionEvents": [
      {
        "eventName": "Overpressurized",
        "condition": "$input.PressureInput.sensorData.pressure > 70",
        "actions": [
          {
            "setVariable": {
              "variableName": "pressureThresholdBreach",
              "value": "$variable.pressureThresholdBreach + 3"
            }
          }
        ],
        "nextState": "Dangerous"
      }
    ]
  },
  {
    "stateName": "Dangerous",
    "onEnter": {
      "events": [
        {
          "eventName": "Pressure Threshold Breached",
          "condition": "$variable.pressureThresholdBreach > 1",
          "actions": [
            {
              "sns": {
                "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
              }
            }
          ]
        }
      ]
    }
  }
]
```

```
    }
  ]
},
"onInput": {
  "events": [
    {
      "eventName": "Overpressurized",
      "condition": "$input.PressureInput.sensorData.pressure > 70",
      "actions": [
        {
          "setVariable": {
            "variableName": "pressureThresholdBreach",
            "value": "3"
          }
        }
      ]
    },
    {
      "eventName": "Pressure Okay",
      "condition": "$input.PressureInput.sensorData.pressure <= 70",
      "actions": [
        {
          "setVariable": {
            "variableName": "pressureThresholdBreach",
            "value": "$variable.pressureThresholdBreach - 1"
          }
        }
      ]
    }
  ],
  "transitionEvents": [
    {
      "eventName": "BackToNormal",
      "condition": "$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreach <= 1",
      "nextState": "Normal"
    }
  ]
},
"onExit": {
  "events": [
    {
      "eventName": "Normal Pressure Restored",
      "condition": "true",
```



```

        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                }
            }
        ]
    }
}
],
"initialStateName": "Normal"
},
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

Ausgabe:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560799387.719,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560799387.719,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "2"
  }
}

```

Weitere Informationen finden Sie [UpdateDetectorModel](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [UpdateDetectorModel](#) in der AWS CLI Befehlsreferenz.

update-input

Das folgende Codebeispiel zeigt die Verwendung `update-input`.

AWS CLI

Um eine Eingabe zu aktualisieren

Das folgende `update-input` Beispiel aktualisiert eine Eingabe.

```
aws iotevents update-input \  
  --cli-input-json file://pressureInput.json
```

Inhalt von `pressureInput.json`:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "inputConfiguration": {  
    "status": "ACTIVE",  
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
    "lastUpdateTime": 1560795976.458,  
    "creationTime": 1560795312.542,  
    "inputName": "PressureInput",  
    "inputDescription": "Pressure readings from a motor"  
  }  
}
```

Weitere Informationen finden Sie [UpdateInput](#) im AWS IoT Events Developer Guide*.

- API-Einheiten finden Sie [UpdateInput](#) in der AWS CLI Befehlsreferenz.

AWS IoT Greengrass Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Greengrass.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-role-to-group

Das folgende Codebeispiel zeigt die Verwendung `associate-role-to-group`.

AWS CLI

Um eine Rolle mit einer Greengrass-Gruppe zu verknüpfen

Im folgenden `associate-role-to-group` Beispiel wird die angegebene IAM Rolle einer Greengrass-Gruppe zugeordnet. Die Gruppenrolle wird von lokalen Lambda-Funktionen und Konnektoren für den Zugriff auf AWS Dienste verwendet. Beispielsweise kann Ihre Gruppenrolle die für die CloudWatch Logs-Integration erforderlichen Berechtigungen gewähren.

```
aws greengrass associate-role-to-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \  
  --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

Ausgabe:

```
{  
  "AssociatedAt": "2019-09-10T20:03:30Z"  
}
```

Weitere Informationen finden Sie unter [Konfiguration der Gruppenrolle](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [AssociateRoleToGroup](#) in der AWS CLI Befehlsreferenz.

associate-service-role-to-account

Das folgende Codebeispiel zeigt die Verwendung `associate-service-role-to-account`.

AWS CLI

Um Ihrem AWS Konto eine Servicerolle zuzuordnen

Das folgende `associate-service-role-to-account` Beispiel ordnet eine durch ihre ARN angegebene IAM Servicerolle AWS IoT Greengrass in Ihrem AWS Konto zu. Sie müssen die Servicerolle zuvor in erstellt haben und ihr ein Richtliniendokument zuordnen IAM, das es AWS IoT Greengrass ermöglicht, diese Rolle zu übernehmen.

```
aws greengrass associate-service-role-to-account \
  --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

Ausgabe:

```
{
  "AssociatedAt": "2019-06-25T18:12:45Z"
}
```

Weitere Informationen finden Sie unter [Greengrass Service Role](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [AssociateServiceRoleToAccount](#) in der AWS CLI Befehlsreferenz.

create-connector-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-connector-definition-version`.

AWS CLI

Um eine Connector-Definitionsversion zu erstellen

Im folgenden `create-connector-definition-version` Beispiel wird eine Konnektordefinitionsversion erstellt und sie der angegebenen Konnektordefinition zugeordnet. Alle Stecker in einer Version definieren Werte für ihre Parameter.

```
aws greengrass create-connector-definition-version \
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
  --connectors "[{"Id": "MyTwilioNotificationsConnector",
  "ConnectorArn": "arn:aws:greengrass:us-west-2:/:connectors/
  TwilioNotifications/versions/2", "Parameters": {"TWILIO_ACCOUNT_SID
  ": "AC1a8d4204890840d7fc482aab38090d57", "TwilioAuthTokenSecretArn":
  "arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-
  ntS1p6", "TwilioAuthTokenSecretArn-ResourceId": "TwilioAuthToken",
  "DefaultFromPhoneNumber": "4254492999"}]]"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
  connectors/55d0052b-0d7d-44d6-b56f-21867215e118/versions/33f709a0-c825-49cb-9eea-
  dc8964fbd635",
  "CreationTimestamp": "2019-06-24T20:46:30.134Z",
  "Id": "55d0052b-0d7d-44d6-b56f-21867215e118",
  "Version": "33f709a0-c825-49cb-9eea-dc8964fbd635"
}
```

- API-Einheiten finden Sie [CreateConnectorDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

create-connector-definition

Das folgende Codebeispiel zeigt die Verwendung `create-connector-definition`.

AWS CLI

Um eine Konnektordefinition zu erstellen

Im folgenden `create-connector-definition` Beispiel werden eine Konnektordefinition und eine erste Version der Konnektordefinition erstellt. Die erste Version enthält einen Konnektor. Alle Konnektoren in einer Version definieren Werte für ihre Parameter.

```
aws greengrass create-connector-definition \
  --name MySNSConnector \
```

```
--initial-version "{\"Connectors\": [{\"Id\": \"MySNSConnector\", \"ConnectorArn\": \"arn:aws:greengrass:us-west-2:/connectors/SNS/versions/1\", \"Parameters\": {\"DefaultSNSArn\": \"arn:aws:sns:us-west-2:123456789012:GGConnectorTopic\"}}]}
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
  "CreationTimestamp": "2019-06-19T19:30:01.300Z",
  "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
  "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
  "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",
  "Name": "MySNSConnector"
}
```

Weitere Informationen finden Sie unter [Getting Started with Greengrass Connectors \(CLI\)](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateConnectorDefinition](#) in der AWS CLI Befehlsreferenz.

create-core-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-core-definition-version`.

AWS CLI

Um eine Core-Definitionsversion zu erstellen

Im folgenden `create-core-definition-version` Beispiel wird eine Version der Kerndefinition erstellt und mit der angegebenen Kerndefinition verknüpft. Die Version kann nur einen Kern enthalten. Bevor Sie einen Core erstellen können, müssen Sie zunächst das entsprechende AWS IoT-Ding erstellen und bereitstellen. Dieser Prozess umfasst die folgenden `iot` Befehle, die das `ThingArn` und, was für den `create-core-definition-version` Befehl `CertificateArn` erforderlich ist, zurückgeben.

Erstellen Sie das AWS IoT-Ding, das dem Kerngerät entspricht:

```
aws iot create-thing \
```

```
--thing-name "MyCoreDevice"
```

Ausgabe:

```
{
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
  "thingName": "MyCoreDevice",
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

Erstellen Sie öffentliche und private Schlüssel und das Kerngerätezertifikat für das Ding. Dieses Beispiel verwendet den `create-keys-and-certificate` Befehl und erfordert Schreibberechtigungen für das aktuelle Verzeichnis. Alternativ können Sie den `create-certificate-from-csr` Befehl verwenden.

```
aws iot create-keys-and-certificate \
  --set-as-active \
  --certificate-pem-outfile "myCore.cert.pem" \
  --public-key-outfile "myCore.public.key" \
  --private-key-outfile "myCore.private.key"
```

Ausgabe:

```
{
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
  "certificatePem": "-----BEGIN CERTIFICATE-----\nMIIDWTCakGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEaqKpRgnn6yq26U3y...wIDAQAB\n-----END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIABAKCAQEaqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcNOK4rMCxDR...fvY4+te\n-----END RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

Erstellen Sie eine AWS IoT-Richtlinie, die erlaubt `iot` und `greengrass` Maßnahmen ergreift. Der Einfachheit halber erlaubt die folgende Richtlinie Aktionen für alle Ressourcen, Ihre Richtlinie sollte jedoch restriktiver sein.

```
aws iot create-policy \
  --policy-name "Core_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource\": [\"*\"]} ]}"
```

Ausgabe:

```
{
  "policyName": "Core_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
  "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource\": [\"*\"]} ]}",
  "policyVersionId": "1"
}
```

Hängen Sie die Richtlinie an das Zertifikat an:

```
aws iot attach-policy \
  --policy-name "Core_Devices" \
  --target "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Hängen Sie das Ding an das Zertifikat an:

```
aws iot attach-thing-principal \
  --thing-name "MyCoreDevice" \
```



```
--principal "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Erstellen Sie die Version der Kerndefinition:

```
aws greengrass create-core-definition-version \
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
  --cores "[{"Id\":\"MyCoreDevice\"}, {"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice\"}, {"CertificateArn\":\"arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz\"}, {"SyncShadow\":\"true\"}]"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/3fdc1190-2ce5-44de-b98b-eec8f9571014",
  "Version": "3fdc1190-2ce5-44de-b98b-eec8f9571014",
  "CreationTimestamp": "2019-09-18T00:15:09.838Z",
  "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

Weitere Informationen finden [Sie unter Configure the AWS IoT Greengrass Core](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateCoreDefinitionVersion](#) in der AWS CLI Befehlsreferenz.

create-core-definition

Das folgende Codebeispiel zeigt die Verwendung `create-core-definition`.

AWS CLI

Beispiel 1: Um eine leere Kerndefinition zu erstellen

Das folgende `create-core-definition` Beispiel erstellt eine leere Greengrass-Core-Definition (keine ursprüngliche Version). Bevor der Kern verwendet werden kann, müssen Sie den `create-core-definition-version` Befehl verwenden, um die anderen Parameter für den Kern bereitzustellen.

```
aws greengrass create-core-definition \  
  --name cliGroup_Core
```

Ausgabe:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/  
b5c08008-54cb-44bd-9eec-c121b04283b5",  
  "CreationTimestamp": "2019-06-25T18:23:22.106Z",  
  "Id": "b5c08008-54cb-44bd-9eec-c121b04283b5",  
  "LastUpdatedTimestamp": "2019-06-25T18:23:22.106Z",  
  "Name": "cliGroup_Core"  
}
```

Beispiel 2: Um eine Kerndefinition mit einer ersten Version zu erstellen

Im folgenden `create-core-definition` Beispiel wird eine Kerndefinition erstellt, die eine erste Version der Kerndefinition enthält. Die Version kann nur einen Kern enthalten. Bevor Sie einen Core erstellen können, müssen Sie zunächst das entsprechende AWS IoT-Ding erstellen und bereitstellen. Dieser Prozess umfasst die folgenden `iot` Befehle, die das `ThingArn` und, was für den `create-core-definition` Befehl `CertificateArn` erforderlich ist, zurückgeben.

Erstellen Sie das AWS IoT-Ding, das dem Kerngerät entspricht:

```
aws iot create-thing \  
  --thing-name "MyCoreDevice"
```

Ausgabe:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",  
  "thingName": "MyCoreDevice",  
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"  
}
```

Erstellen Sie öffentliche und private Schlüssel und das Kerngerätezertifikat für das Ding. Dieses Beispiel verwendet den `create-keys-and-certificate` Befehl und erfordert Schreibberechtigungen für das aktuelle Verzeichnis. Alternativ können Sie den `create-certificate-from-csr` Befehl verwenden.

```
aws iot create-keys-and-certificate \
  --set-as-active \
  --certificate-pem-outfile "myCore.cert.pem" \
  --public-key-outfile "myCore.public.key" \
  --private-key-outfile "myCore.private.key"
```

Ausgabe:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCakGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDcN0K4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

Erstellen Sie eine AWS IoT-Richtlinie, die erlaubt `iot` und `greengrass` Maßnahmen ergreift. Der Einfachheit halber erlaubt die folgende Richtlinie Aktionen für alle Ressourcen, Ihre Richtlinie sollte jedoch restriktiver sein.

```
aws iot create-policy \
  --policy-name "Core_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect
\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\":
[\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"],
\"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource
\": [\"*\"]}]}\""
```

Ausgabe:

```
{
  "policyName": "Core_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\": \"Allow\", \"Action\": [ \"iot:Publish\", \"iot:Subscribe\", \"iot:Connect\", \"iot:Receive\" ], \"Resource\": [ \"*\" ] }, { \"Effect\": \"Allow\", \"Action\": [ \"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\" ], \"Resource\": [ \"*\" ] }, { \"Effect\": \"Allow\", \"Action\": [ \"greengrass:*\" ], \"Resource\": [ \"*\" ] } ] }",
  "policyVersionId": "1"
}
```

Hängen Sie die Richtlinie an das Zertifikat an:

```
aws iot attach-policy \
  --policy-name "Core_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Hängen Sie das Ding an das Zertifikat an:

```
aws iot attach-thing-principal \
  --thing-name "MyCoreDevice" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Erstellen Sie die Kerndefinition:

```
aws greengrass create-core-definition \
  --name "MyCores" \
  --initial-version "{ \"Cores\": [{ \"Id\": \"MyCoreDevice\", \"ThingArn\":
\"arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice\", \"CertificateArn\":
\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\", \"SyncShadow\": true } ] }"
```

Ausgabe:

```
{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
  "Name": "MyCores",
  "LastUpdatedTimestamp": "2019-09-18T00:11:06.197Z",
  "LatestVersion": "cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
  "CreationTimestamp": "2019-09-18T00:11:06.197Z",
  "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12",
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

Weitere Informationen finden [Sie unter Configure the AWS IoT Greengrass Core](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateCoreDefinition](#) in der AWS CLI Befehlsreferenz.

create-deployment

Das folgende Codebeispiel zeigt die Verwendung `create-deployment`.

AWS CLI

Um ein Deployment für eine Version einer Greengrass-Gruppe zu erstellen

Im folgenden `create-deployment` Beispiel wird die angegebene Version einer Greengrass-Gruppe bereitgestellt.

```
aws greengrass create-deployment \
  --deployment-type NewDeployment \
  --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
  --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

Ausgabe:

```
{
  "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/deployments/bfceb608-4e97-45bc-af5c-460144270308",
  "DeploymentId": "bfceb608-4e97-45bc-af5c-460144270308"
}
```

```
}
```

Weitere Informationen finden Sie unter [Getting Started with Connectors \(CLI\)](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateDeployment](#) in der AWS CLI Befehlsreferenz.

create-device-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-device-definition-version`.

AWS CLI

Um eine Gerätedefinitionsversion zu erstellen

Das folgende `create-device-definition-version` Beispiel erstellt eine Gerätedefinitionsversion und ordnet sie der angegebenen Gerätedefinition zu. Die Version definiert zwei Geräte. Bevor Sie ein Greengrass-Gerät erstellen können, müssen Sie zunächst das entsprechende AWS IoT-Ding erstellen und bereitstellen. Dieser Prozess umfasst die folgenden `iot` Befehle, die Sie ausführen müssen, um die erforderlichen Informationen für den Befehl Greengrass abzurufen:

Erstellen Sie das AWS IoT-Ding, das dem Gerät entspricht:

```
aws iot create-thing \  
  --thing-name "InteriorTherm"
```

Ausgabe:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",  
  "thingName": "InteriorTherm",  
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"  
}
```

Erstellen Sie öffentliche und private Schlüssel und das Gerätezertifikat für das Ding. Dieses Beispiel verwendet den `create-keys-and-certificate` Befehl und erfordert Schreibberechtigungen für das aktuelle Verzeichnis. Alternativ können Sie den `create-certificate-from-csr` folgenden Befehl verwenden:

```
aws iot create-keys-and-certificate \  
  --thing-name "InteriorTherm"
```

```
--set-as-active \
--certificate-pem-outfile "myDevice.cert.pem" \
--public-key-outfile "myDevice.public.key" \
--private-key-outfile "myDevice.private.key"
```

Ausgabe:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCaKgAwIBATgIUCGq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDCnOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

Erstellen Sie eine AWS IoT-Richtlinie, die erlaubt `iot` und `greengrass` Maßnahmen ergreift. Der Einfachheit halber erlaubt die folgende Richtlinie Aktionen für alle Ressourcen, Ihre Richtlinie kann jedoch restriktiver sein:

```
aws iot create-policy \
  --policy-name "GG_Devices" \
  --policy-document "{\n\"Version\":\n\"2012-10-17\", \"Statement\": [{\n\"Effect
\":\n\"Allow\", \"Action\": [\n\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect
\", \"iot:Receive\"], \"Resource\": [\n\"*\n\"]}, {\n\"Effect\":\n\"Allow\", \"Action\":
[\n\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"],
\n\"Resource\": [\n\"*\n\"]}, {\n\"Effect\":\n\"Allow\", \"Action\": [\n\"greengrass:*\"], \"Resource
\": [\n\"*\n\"]}]}"
```

Ausgabe:

```
{
  "policyName": "GG_Devices",
```

```

    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":"
    "\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect\","
    "\":\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":"
    "\":[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],\":"
    "\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource\":"
    "\":[\"*\"]}]}",
    "policyVersionId": "1"
  }

```

Hängen Sie die Richtlinie an das Zertifikat an:

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Hängen Sie das Ding an das Zertifikat an

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Nachdem Sie das IoT-Ding wie oben gezeigt erstellt und konfiguriert haben, verwenden Sie im folgenden Beispiel die Befehle ThingArn und CertificateArn aus den ersten beiden Befehlen.

```

aws greengrass create-device-definition-version \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
  --devices "[{\"Id\":\"InteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/InteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\"},\":"
    "\":[\"SyncShadow\":true},{\"Id\":\"ExteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\"},\":"
    "\":[\"SyncShadow\":true]}]"

```

Ausgabe:

```
{
```



```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
    "CreationTimestamp": "2019-09-11T00:15:09.838Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}

```

- API-Einheiten finden Sie [CreateDeviceDefinitionVersion](#) in der AWS CLI Befehlsreferenz.

create-device-definition

Das folgende Codebeispiel zeigt die Verwendung `create-device-definition`.

AWS CLI

Um eine Gerätedefinition zu erstellen

Im folgenden `create-device-definition` Beispiel wird eine Gerätedefinition erstellt, die eine erste Version der Gerätedefinition enthält. Die erste Version definiert zwei Geräte. Bevor Sie ein Greengrass-Gerät erstellen können, müssen Sie zunächst das entsprechende AWS IoT-Ding erstellen und bereitstellen. Dieser Prozess umfasst die folgenden `iot` Befehle, die Sie ausführen müssen, um die erforderlichen Informationen für den Befehl `Greengrass` abzurufen:

Erstellen Sie das AWS IoT-Ding, das dem Gerät entspricht:

```

aws iot create-thing \
  --thing-name "InteriorTherm"

```

Ausgabe:

```

{
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
  "thingName": "InteriorTherm",
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}

```

Erstellen Sie öffentliche und private Schlüssel und das Gerätezertifikat für das Ding. Dieses Beispiel verwendet den `create-keys-and-certificate` Befehl und erfordert Schreibberechtigungen für das aktuelle Verzeichnis. Alternativ können Sie den `create-certificate-from-csr` folgenden Befehl verwenden:

```
aws iot create-keys-and-certificate \
  --set-as-active \
  --certificate-pem-outfile "myDevice.cert.pem" \
  --public-key-outfile "myDevice.public.key" \
  --private-key-outfile "myDevice.private.key"
```

Ausgabe:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCaKgAwIBATgIUCGq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDcNOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

Erstellen Sie eine AWS IoT-Richtlinie, die erlaubt `iot` und `greengrass` Maßnahmen ergreift. Der Einfachheit halber erlaubt die folgende Richtlinie Aktionen für alle Ressourcen, Ihre Richtlinie kann jedoch restriktiver sein:

```
aws iot create-policy \
  --policy-name "GG_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect
\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\":
[\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"],
\"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource
\": [\"*\"]}]}"
```

Ausgabe:

```
{
  "policyName": "GG_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
  "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource\":[\"*\"]}]}",
  "policyVersionId": "1"
}
```

Hängen Sie die Richtlinie an das Zertifikat an:

```
aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Hängen Sie das Ding an das Zertifikat an

```
aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Nachdem Sie das IoT-Ding wie oben gezeigt erstellt und konfiguriert haben, verwenden Sie im folgenden Beispiel die Befehle ThingArn und CertificateArn aus den ersten beiden Befehlen.

```
aws greengrass create-device-definition \
  --name "Sensors" \
  --initial-version "{\"Devices\":[{\"Id\":\"InteriorTherm
\", \"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/
InteriorTherm\", \"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\"},
{ \"SyncShadow\":true}], {\"Id\":\"ExteriorTherm\", \"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\", \"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\"},
{ \"SyncShadow\":true}]}"
```

Ausgabe:

```
{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795",
  "Name": "Sensors",
  "LastUpdatedTimestamp": "2019-09-11T00:11:06.197Z",
  "LatestVersion": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
  "CreationTimestamp": "2019-09-11T00:11:06.197Z",
  "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- API-Einheiten finden Sie [CreateDeviceDefinition](#) in der AWS CLI Befehlsreferenz.

create-function-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-function-definition-version`.

AWS CLI

Um eine Version der Funktionsdefinition zu erstellen

Im folgenden `create-function-definition-version` Beispiel wird eine neue Version der angegebenen Funktionsdefinition erstellt. Diese Version spezifiziert eine einzelne Funktion `Hello-World-function`, deren ID den Zugriff auf das Dateisystem ermöglicht, und gibt eine maximale Speichergröße und einen Timeoutzeitraum an.

```
aws greengrass create-function-definition-version \
  --cli-input-json '{"FunctionDefinitionId": "e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3","Functions": [{"Id": "Hello-World-function", "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:Greengrass_HelloWorld_Counter:gghw-alias"}, {"FunctionConfiguration": {"Environment": {"AccessSysfs": true}, "Executable": "greengrassHelloWorldCounter.function_handler", "MemorySize": 16000, "Pinned": false, "Timeout": 25}}]}'
```

Ausgabe:

```
{
```

```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3/
versions/74abd1cc-637e-4abe-8684-9a67890f4043",
    "CreationTimestamp": "2019-06-25T22:03:43.376Z",
    "Id": "e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3",
    "Version": "74abd1cc-637e-4abe-8684-9a67890f4043"
}

```

- API-Einheiten finden Sie [CreateFunctionDefinitionVersion](#) in der AWS CLI Befehlsreferenz.

create-function-definition

Das folgende Codebeispiel zeigt die Verwendung `create-function-definition`.

AWS CLI

Um eine Lambda-Funktionsdefinition zu erstellen

Das folgende `create-function-definition` Beispiel erstellt eine Lambda-Funktionsdefinition und eine erste Version, indem es eine Liste von Lambda-Funktionen (in diesem Fall eine Liste mit nur einer benannten Funktion `TempMonitorFunction`) und deren Konfigurationen bereitstellt. Bevor Sie die Funktionsdefinition erstellen können, benötigen Sie die Lambda-FunktionARN. Verwenden Sie `Lambdas` und Befehle, um die Funktion und ihren Alias zu erstellen. `create-function publish-version` Der `create-function` Befehl ARN von Lambda erfordert die Ausführungsrolle, obwohl AWS IoT Greengrass diese Rolle nicht verwendet, da die Berechtigungen in der Greengrass-Gruppenrolle angegeben sind. Sie können den IAM `create-role` Befehl verwenden, um eine leere Rolle zu erstellen, um sie mit Lambdas ARN zu verwenden, `create-function` oder Sie können eine vorhandene Ausführungsrolle verwenden.

```

aws greengrass create-function-definition \
  --name MyGreengrassFunctions \
  --initial-version "{\"Functions\": [{\"Id\": \"TempMonitorFunction\",
  \"FunctionArn\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"FunctionConfiguration
\": {\"Executable\": \"temp_monitor.function_handler\", \"MemorySize\": 16000,
  \"Timeout\": 5}}]}"

```

Ausgabe:

```
{
```

```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/3b0d0080-87e7-48c6-b182-503ec743a08b",
    "CreationTimestamp": "2019-06-19T22:24:44.585Z",
    "Id": "3b0d0080-87e7-48c6-b182-503ec743a08b",
    "LastUpdatedTimestamp": "2019-06-19T22:24:44.585Z",
    "LatestVersion": "67f918b9-efb4-40b0-b87c-de8c9faf085b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/versions/67f918b9-
efb4-40b0-b87c-de8c9faf085b",
    "Name": "MyGreengrassFunctions"
}

```

Weitere Informationen finden Sie unter [How to Configure Local Resource Access Using the AWS Command Line Interface](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [CreateFunctionDefinition AWS CLI Befehlsreferenz](#).

create-group-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `create-group-certificate-authority`.

AWS CLI

Um eine Zertifizierungsstelle (CA) für eine Gruppe zu erstellen

Im folgenden `create-group-certificate-authority` Beispiel wird eine Zertifizierungsstelle für die angegebene Gruppe erstellt oder rotiert.

```

aws greengrass create-group-certificate-authority \
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"

```

Ausgabe:

```

{
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEecc5"
}

```

Weitere Informationen finden Sie unter [AWS IoT Greengrass Security](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateGroupCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

create-group-version

Das folgende Codebeispiel zeigt die Verwendung `create-group-version`.

AWS CLI

Um eine Version einer Greengrass-Gruppe zu erstellen

Das folgende `create-group-version` Beispiel erstellt eine Gruppenversion und ordnet sie der angegebenen Gruppe zu. Die Version verweist auf die Core-, Resource-, Connector-, Funktions- und Abonnementversionen, die die Entitäten enthalten, die in diese Gruppenversion aufgenommen werden sollen. Sie müssen diese Entitäten erstellen, bevor Sie die Gruppenversion erstellen können.

Um eine Ressourcendefinition mit einer ersten Version zu erstellen, verwenden Sie den `create-resource-definition` Befehl. Um eine Connectordefinition mit einer ersten Version zu erstellen, verwenden Sie den `create-connector-definition` Befehl. Um eine Funktionsdefinition mit einer ersten Version zu erstellen, verwenden Sie den `create-function-definition` Befehl. Um eine Abonnementdefinition mit einer ersten Version zu erstellen, verwenden Sie den Befehl `create-subscription-definition`. Um die neueste Version ARN der Kerndefinition abzurufen, verwenden Sie den `get-group-version` Befehl und geben Sie die ID der neuesten Gruppenversion an.

```
aws greengrass create-group-version \
  --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
  --core-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/6a630442-8708-4838-ad36-eb98849d975e/
versions/6c87151b-1fb4-4cb2-8b31-6ee715d8f8ba" \
  --resource-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1" \
  --connector-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/55d0052b-0d7d-44d6-b56f-21867215e118/
versions/78a3331b-895d-489b-8823-17b4f9f418a0" \
  --function-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/
versions/67f918b9-efb4-40b0-b87c-de8c9faf085b" \
  --subscription-definition-version-arn "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/definition/subscriptions/9d611d57-5d5d-44bd-a3b4-
fecbbdd69112/versions/aa645c47-ac90-420d-9091-8c7ffa4f103f"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/
ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/versions/e10b0459-4345-4a09-88a4-1af1f5d34638",
  "CreationTimestamp": "2019-06-20T18:42:47.020Z",
  "Id": "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca",
  "Version": "e10b0459-4345-4a09-88a4-1af1f5d34638"
}
```

Weitere Informationen finden Sie unter [Überblick über das AWS IoT Greengrass Group Object Model](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateGroupVersion](#) in der AWS CLI Befehlsreferenz.

create-group

Das folgende Codebeispiel zeigt die Verwendung `create-group`.

AWS CLI

Um eine Greengrass-Gruppe zu erstellen

Im folgenden `create-group` Beispiel wird eine Gruppe mit dem Namen erstellt. `cli-created-group`

```
aws greengrass create-group \
  --name cli-created-group
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/4e22bd92-898c-436b-ade5-434d883ff749",
  "CreationTimestamp": "2019-06-25T18:07:17.688Z",
  "Id": "4e22bd92-898c-436b-ade5-434d883ff749",
  "LastUpdatedTimestamp": "2019-06-25T18:07:17.688Z",
  "Name": "cli-created-group"
}
```

Weitere Informationen finden Sie unter [Überblick über das AWS IoT Greengrass Group Object Model](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateGroup](#) in der AWS CLI Befehlsreferenz.

create-logger-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-logger-definition-version`.

AWS CLI

Um eine Logger-Definitionsversion zu erstellen

Das folgende `create-logger-definition-version` Beispiel erstellt eine Logger-Definitionsversion und ordnet sie einer Logger-Definition zu. Die Version definiert vier Protokollierungskonfigurationen: 1) Systemkomponentenprotokolle im Dateisystem des Kerngeräts, 2) benutzerdefinierte Lambda-Funktionsprotokolle im Dateisystem des Kerngeräts, 3) Systemkomponentenprotokolle in Amazon CloudWatch Logs und 4) benutzerdefinierte Lambda-Funktionsprotokolle in Amazon Logs. CloudWatch Hinweis: Für die CloudWatch Logs-Integration muss Ihre Gruppenrolle die entsprechenden Berechtigungen gewähren.

```
aws greengrass create-logger-definition-version \
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
  --loggers "[{"Id":"1","Component":"GreengrassSystem","Level":"ERROR",
  "Space":10240,"Type":"FileSystem"}, {"Id":"2","Component":"Lambda",
  "Level":"INFO","Space":10240,"Type":"FileSystem"}, {"Id":"3",
  "Component":"GreengrassSystem","Level":"WARN","Type":"AWSCloudWatch"},
  {"Id":"4","Component":"Lambda","Level":"INFO","Type":"AWSCloudWatch"}]"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/49aedb1e-01a3-4d39-9871-3a052573f1ea",
  "Version": "49aedb1e-01a3-4d39-9871-3a052573f1ea",
  "CreationTimestamp": "2019-07-24T00:04:48.523Z",
  "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
}
```

Weitere Informationen finden Sie unter [Monitoring with AWS IoT Greengrass Logs](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateLoggerDefinitionVersion](#) in der AWS CLI Befehlsreferenz.

create-logger-definition

Das folgende Codebeispiel zeigt die Verwendung `create-logger-definition`.

AWS CLI

Um eine Logger-Definition zu erstellen

Im folgenden `create-logger-definition` Beispiel wird eine Logger-Definition erstellt, die eine erste Logger-Definitionsversion enthält. Die erste Version definiert drei Protokollierungskonfigurationen: 1) Systemkomponentenprotokolle im Dateisystem des Kerngeräts, 2) benutzerdefinierte Lambda-Funktionsprotokolle im Dateisystem des Kerngeräts und 3) benutzerdefinierte Lambda-Funktionsprotokolle in Amazon Logs. CloudWatch Hinweis: Für die CloudWatch Logs-Integration muss Ihre Gruppenrolle die entsprechenden Berechtigungen gewähren.

```
aws greengrass create-logger-definition \
  --name "LoggingConfigs" \
  --initial-version '{"Loggers":[{"Id":"1","Component":"GreengrassSystem",
  "Level":"ERROR","Space":10240,"Type":"FileSystem"}, {"Id":"2",
  "Component":"Lambda","Level":"INFO","Space":10240,"Type":"FileSystem"}, {"Id":"3",
  "Component":"Lambda","Level":"INFO","Type":"AWSCloudWatch"}]}'
```

Ausgabe:

```
{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/de1d9854-1588-4525-b25e-b378f60f2322",
  "Name": "LoggingConfigs",
  "LastUpdatedTimestamp": "2019-07-23T23:52:17.165Z",
  "LatestVersion": "de1d9854-1588-4525-b25e-b378f60f2322",
  "CreationTimestamp": "2019-07-23T23:52:17.165Z",
  "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0",
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
}
```

Weitere Informationen finden Sie unter [Monitoring with AWS IoT Greengrass Logs](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateLoggerDefinition](#) in der AWS CLI Befehlsreferenz.

create-resource-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-resource-definition-version`.

AWS CLI

Um eine Version einer Ressourcendefinition zu erstellen

Das folgende `create-resource-definition-version` Beispiel erstellt eine neue Version von `TwilioAuthToken`.

```
aws greengrass create-resource-definition-version \
  --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \
  --resources "[{"Id": "TwilioAuthToken"}, {"Name": "MyTwilioAuthToken", "ResourceDataContainer": {"SecretsManagerSecretResourceData": {"ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:greenrass-TwilioAuthToken-ntS1p6"}}}]"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/b3bcada0-5fb6-42df-bf0b-1ee4f15e769e",
  "CreationTimestamp": "2019-06-24T21:17:25.623Z",
  "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
  "Version": "b3bcada0-5fb6-42df-bf0b-1ee4f15e769e"
}
```

- API-Einheiten finden Sie [CreateResourceDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

create-resource-definition

Das folgende Codebeispiel zeigt die Verwendung `create-resource-definition`.

AWS CLI

Um eine Ressourcendefinition zu erstellen

Im folgenden `create-resource-definition` Beispiel wird eine Ressourcendefinition erstellt, die eine Liste von Ressourcen enthält, die in einer Greengrass-Gruppe verwendet werden sollen. In diesem Beispiel ist eine erste Version der Ressourcendefinition enthalten, indem eine Liste von Ressourcen bereitgestellt wird. Die Liste enthält eine Ressource für ein Twilio-Autorisierungstoken und die ARN für ein in AWS Secrets Manager gespeichertes Geheimnis. Sie müssen das Geheimnis erstellen, bevor Sie die Ressourcendefinition erstellen können.

```
aws greengrass create-resource-definition \
  --name MyGreengrassResources \
  --initial-version "{\"Resources\": [{\"Id\": \"TwilioAuthToken
\", \"Name\": \"MyTwilioAuthToken\", \"ResourceDataContainer\":
 {\"SecretsManagerSecretResourceData\": {\"ARN\": \"arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-TwilioAuthToken-ntSlp6\"}}}]}"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
  "CreationTimestamp": "2019-06-19T21:51:28.212Z",
  "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
  "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
  "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/a5f94d0b-
f6bc-40f4-bb78-7a1c5fe13ba1",
  "Name": "MyGreengrassResources"
}
```

Weitere Informationen finden Sie unter [How to Configure Local Resource Access Using the AWS Command Line Interface](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [CreateResourceDefinition AWS CLI Befehlsreferenz](#).

create-software-update-job

Das folgende Codebeispiel zeigt die Verwendung `create-software-update-job`.

AWS CLI

Um einen Softwareupdate-Job für einen Core zu erstellen

Im folgenden `create-software-update-job` Beispiel wird ein Aktualisierungsjob *over-the-air* (OTA) erstellt, um die AWS IoT Greengrass Core-Software auf dem Core zu aktualisieren, dessen Name lautet `MyFirstGroup_Core`. Für diesen Befehl ist eine IAM Rolle erforderlich, die den Zugriff auf Softwareupdatepakete in Amazon S3 ermöglicht und `iot.amazonaws.com` als vertrauenswürdige Entität enthalten ist.

```
aws greengrass create-software-update-job \
  --update-targets-architecture armv7l \
  --update-targets ["arn:aws:iot:us-west-2:123456789012:thing/MyFirstGroup_Core
  \"] \
  --update-targets-operating-system raspbian \
  --software-to-update core \
  --s3-url-signer-role arn:aws:iam::123456789012:role/OTA_signer_role \
  --update-agent-log-level WARN
```

Ausgabe:

```
{
  "IotJobId": "GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",
  "IotJobArn": "arn:aws:iot:us-west-2:123456789012:job/
  GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",
  "PlatformSoftwareVersion": "1.9.3"
}
```

Weitere Informationen finden Sie unter [OTAUpdates der AWS IoT Greengrass Core Software](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateSoftwareUpdateJob](#) in der AWS CLI Befehlsreferenz.

create-subscription-definition-version

Das folgende Codebeispiel zeigt die Verwendung `create-subscription-definition-version`.

AWS CLI

Um eine neue Version einer Abonnementdefinition zu erstellen

Im folgenden `create-subscription-definition-version` Beispiel wird eine neue Version einer Abonnementdefinition erstellt, die drei Abonnements enthält: eine Triggerbenachrichtigung, eine Temperatureingabe und einen Ausgabestatus.

```
aws greengrass create-subscription-definition-version \
  --subscription-definition-id "9d611d57-5d5d-44bd-a3b4-feccbdd69112" \
  --subscriptions "[{\\"Id\\": \\"TriggerNotification\\", \\"Source\\":
  \\"arn:aws:lambda:us-west-2:123456789012:function:TempMonitor:GG_TempMonitor
  \", \\"Subject\\": \\"twilio/txt\\", \\"Target\\": \\"arn:aws:greengrass:us-west-2:/:
  connectors/TwilioNotifications/versions/1\\"},{\\"Id\\": \\"TemperatureInput\\", \\"Source
  \": \\"cloud\\", \\"Subject\\": \\"temperature/input\\", \\"Target\\": \\"arn:aws:lambda:us-
  west-2:123456789012:function:TempMonitor:GG_TempMonitor\\"},{\\"Id\\": \\"OutputStatus
  \", \\"Source\\": \\"arn:aws:greengrass:us-west-2:/:connectors/TwilioNotifications/
  versions/1\\", \\"Subject\\": \\"twilio/message/status\\", \\"Target\\": \\"cloud\\"}]"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
  subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/7b65dfae-50b6-4d0f-
  b3e0-27728bfb0620",
  "CreationTimestamp": "2019-06-24T21:21:33.837Z",
  "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
  "Version": "7b65dfae-50b6-4d0f-b3e0-27728bfb0620"
}
```

- API-Einheiten finden Sie [CreateSubscriptionDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

create-subscription-definition

Das folgende Codebeispiel zeigt die Verwendung `create-subscription-definition`.

AWS CLI

Um eine Abonnementdefinition zu erstellen

Im folgenden `create-subscription-definition` Beispiel wird eine Abonnementdefinition erstellt und ihre ursprüngliche Version angegeben. Die erste Version enthält drei Abonnements: eines für das MQTT Thema, das der Connector abonniert, eines, das es einer Funktion ermöglicht, Temperaturwerte vom AWS IoT zu empfangen, und eines, das es AWS IoT ermöglicht, Statusinformationen vom Connector zu empfangen. Das Beispiel stellt den Alias ARN für die Lambda-Funktion bereit, der zuvor mithilfe des `create-alias` Lambda-Befehls erstellt wurde.

```
aws greengrass create-subscription-definition \
```

```
--initial-version "{\"Subscriptions\": [{\"Id\":
  \"TriggerNotification\", \"Source\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"Subject\":
  \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2:/connectors/
TwilioNotifications/versions/1\"},{\"Id\": \"TemperatureInput\", \"Source\":
  \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\"},{\"Id\": \"OutputStatus
\", \"Source\": \"arn:aws:greengrass:us-west-2:/connectors/TwilioNotifications/
versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]}"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112",
  "CreationTimestamp": "2019-06-19T22:34:26.677Z",
  "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
  "LastUpdatedTimestamp": "2019-06-19T22:34:26.677Z",
  "LatestVersion": "aa645c47-ac90-420d-9091-8c7ffa4f103f",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/aa645c47-
ac90-420d-9091-8c7ffa4f103f"
}
```

Weitere Informationen finden Sie unter [Getting Started with Connectors \(CLI\)](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [CreateSubscriptionDefinition](#) in der AWS CLI Befehlsreferenz.

delete-connector-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-connector-definition`.

AWS CLI

Um eine Konnektordefinition zu löschen

Im folgenden `delete-connector-definition` Beispiel wird die angegebene Greengrass-Connectordefinition gelöscht. Wenn Sie eine Connectordefinition löschen, die von einer Gruppe verwendet wird, kann diese Gruppe nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-connector-definition \
```

```
--connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteConnectorDefinition](#) unter AWS CLI Befehlsreferenz.

delete-core-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-core-definition`.

AWS CLI

Um eine Kerndefinition zu löschen

Das folgende `delete-core-definition` Beispiel löscht die angegebene Greengrass-Core-Definition, einschließlich aller Versionen. Wenn Sie einen Core löschen, der mit einer Greengrass-Gruppe verknüpft ist, kann diese Gruppe nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-core-definition \  
--core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteCoreDefinition](#) in der AWS CLI Befehlsreferenz.

delete-device-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-device-definition`.

AWS CLI

Um eine Gerätedefinition zu löschen

Im folgenden `delete-device-definition` Beispiel wird die angegebene Gerätedefinition einschließlich aller Versionen gelöscht. Wenn Sie eine Gerätedefinitionsversion löschen, die von einer Gruppenversion verwendet wird, kann die Gruppenversion nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-device-definition \  
--device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteDeviceDefinition](#) in der AWS CLI Befehlsreferenz.

delete-function-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-function-definition`.

AWS CLI

Um eine Funktionsdefinition zu löschen

Im folgenden `delete-function-definition` Beispiel wird die angegebene Greengrass-Funktionsdefinition gelöscht. Wenn Sie eine Funktionsdefinition löschen, die von einer Gruppe verwendet wird, kann diese Gruppe nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-function-definition \  
  --function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteFunctionDefinition](#) unter AWS CLI Befehlsreferenz.

delete-group

Das folgende Codebeispiel zeigt die Verwendung `delete-group`.

AWS CLI

Um eine Gruppe zu löschen

Im folgenden `delete-group` Beispiel wird die angegebene Greengrass-Gruppe gelöscht.

```
aws greengrass delete-group \  
  --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteGroup AWS CLI](#) Befehlsreferenz.

delete-logger-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-logger-definition`.

AWS CLI

Um eine Logger-Definition zu löschen

Das folgende `delete-logger-definition` Beispiel löscht die angegebene Logger-Definition, einschließlich aller Logger-Definitionsversionen. Wenn Sie eine Logger-Definitionsversion löschen, die von einer Gruppenversion verwendet wird, kann die Gruppenversion nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Monitoring with AWS IoT Greengrass Logs](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [DeleteLoggerDefinition](#) in der AWS CLI Befehlsreferenz.

`delete-resource-definition`

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-definition`.

AWS CLI

Um eine Ressourcendefinition zu löschen

Im folgenden `delete-resource-definition` Beispiel wird die angegebene Ressourcendefinition einschließlich aller Ressourcenversionen gelöscht. Wenn Sie eine Ressourcendefinition löschen, die von einer Gruppe verwendet wird, kann diese Gruppe nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-resource-definition \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteResourceDefinition](#) unter AWS CLI Befehlsreferenz.

`delete-subscription-definition`

Das folgende Codebeispiel zeigt die Verwendung `delete-subscription-definition`.

AWS CLI

Um eine Abonnementdefinition zu löschen

Im folgenden `delete-subscription-definition` Beispiel wird die angegebene Greengrass-Abonnementdefinition gelöscht. Wenn Sie ein Abonnement löschen, das von einer Gruppe verwendet wird, kann diese Gruppe nicht erfolgreich bereitgestellt werden.

```
aws greengrass delete-subscription-definition \  
  --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteSubscriptionDefinition](#) in der AWS CLI Befehlsreferenz.

disassociate-role-from-group

Das folgende Codebeispiel zeigt die Verwendung `disassociate-role-from-group`.

AWS CLI

Um die Rolle von einer Greengrass-Gruppe zu trennen

Im folgenden `disassociate-role-from-group` Beispiel wird die Zuordnung der IAM Rolle zur angegebenen Greengrass-Gruppe aufgehoben.

```
aws greengrass disassociate-role-from-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Ausgabe:

```
{  
  "DisassociatedAt": "2019-09-10T20:05:49Z"  
}
```

Weitere Informationen finden Sie unter [Konfiguration der Gruppenrolle](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [DisassociateRoleFromGroup](#) in der AWS CLI Befehlsreferenz.

disassociate-service-role-from-account

Das folgende Codebeispiel zeigt die Verwendung `disassociate-service-role-from-account`.

AWS CLI

Um eine Servicerolle von Ihrem AWS Konto zu trennen

Im folgenden `disassociate-service-role-from-account` Beispiel wird die Ihrem AWS Konto zugeordnete Servicerolle entfernt. Wenn Sie die Servicerolle in keiner AWS Region verwenden, verwenden Sie den `delete-role-policy` Befehl, um die `AWSGreengrassResourceAccessRolePolicy` verwaltete Richtlinie von der Rolle zu trennen, und verwenden Sie dann den `delete-role` Befehl, um die Rolle zu löschen.

```
aws greengrass disassociate-service-role-from-account
```

Ausgabe:

```
{
  "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

Weitere Informationen finden Sie unter [Greengrass Service Role](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [DisassociateServiceRoleFromAccount](#) in der AWS CLI Befehlsreferenz.

get-associated-role

Das folgende Codebeispiel zeigt die Verwendung `get-associated-role`.

AWS CLI

Um die Rolle einer Greengrass-Gruppe zuzuordnen

Im folgenden `get-associated-role` Beispiel wird die IAM Rolle abgerufen, die der angegebenen Greengrass-Gruppe zugeordnet ist. Die Gruppenrolle wird von lokalen Lambda-Funktionen und Konnektoren für den Zugriff auf AWS Dienste verwendet.

```
aws greengrass get-associated-role \
```

```
--group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Ausgabe:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",
  "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

Weitere Informationen finden [Sie unter Konfiguration der Gruppenrolle](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [GetAssociatedRole](#) in der AWS CLI Befehlsreferenz.

get-bulk-deployment-status

Das folgende Codebeispiel zeigt die Verwendung von `get-bulk-deployment-status`.

AWS CLI

Um den Status Ihrer Massenbereitstellung zu überprüfen

Im folgenden `get-bulk-deployment-status` Beispiel werden Statusinformationen für den angegebenen Massenbereitstellungsvorgang abgerufen. In diesem Beispiel enthält die Datei, in der die bereitzustellenden Gruppen angegeben wurden, einen ungültigen Eingabedatensatz.

```
aws greengrass get-bulk-deployment-status \
  --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Ausgabe:

```
{
  "BulkDeploymentMetrics": {
    "InvalidInputRecords": 1,
    "RecordsProcessed": 1,
    "RetryAttempts": 0
  },
  "BulkDeploymentStatus": "Completed",
  "CreatedAt": "2019-06-25T16:11:33.265Z",
  "tags": {}
}
```

Weitere Informationen finden Sie unter [Create Bulk Deployments for Groups](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [GetBulkDeploymentStatus AWS CLI](#) Befehlsreferenz.

get-connectivity-info

Das folgende Codebeispiel zeigt die Verwendung `get-connectivity-info`.

AWS CLI

Um die Konnektivitätsinformationen für einen Greengrass-Kern abzurufen

Das folgende `get-connectivity-info` Beispiel zeigt die Endpunkte, über die Geräte eine Verbindung zum angegebenen Greengrass-Core herstellen können. Bei den Konnektivitätsinformationen handelt es sich um eine Liste von IP-Adressen oder Domainnamen mit den entsprechenden Portnummern und optionalen, vom Kunden definierten Metadaten.

```
aws greengrass get-connectivity-info \  
  --thing-name "MyGroup_Core"
```

Ausgabe:

```
{  
  "ConnectivityInfo": [  
    {  
      "Metadata": "",  
      "PortNumber": 8883,  
      "HostAddress": "127.0.0.1",  
      "Id": "AUTOIP_127.0.0.1_0"  
    },  
    {  
      "Metadata": "",  
      "PortNumber": 8883,  
      "HostAddress": "192.168.1.3",  
      "Id": "AUTOIP_192.168.1.3_1"  
    },  
    {  
      "Metadata": "",  
      "PortNumber": 8883,  
      "HostAddress": ":::1",  
      "Id": "AUTOIP_:::1_2"  
    }  
  ]  
}
```

```

    },
    {
      "Metadata": "",
      "PortNumber": 8883,
      "HostAddress": "fe80::1e69:ed93:f5b:f6d",
      "Id": "AUTOIP_fe80::1e69:ed93:f5b:f6d_3"
    }
  ]
}

```

- API-Einheiten finden Sie unter [GetConnectivityInfo AWS CLI](#) Befehlsreferenz.

get-connector-definition-version

Das folgende Codebeispiel zeigt die Verwendung von `get-connector-definition-version`.

AWS CLI

Um Informationen über eine bestimmte Version einer Connectordefinition abzurufen

Im folgenden `get-connector-definition-version` Beispiel werden Informationen über die angegebene Version der angegebenen Connectordefinition abgerufen. Verwenden Sie den `list-connector-definition-versions` Befehl, um die IDs aller Versionen der Connectordefinition abzurufen. Um die ID der letzten Version abzurufen, die der Connectordefinition hinzugefügt wurde, verwenden Sie den `get-connector-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```

aws greengrass get-connector-definition-version \
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \
  --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"

```

Ausgabe:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",
  "CreationTimestamp": "2019-06-19T19:30:01.300Z",
  "Definition": {
    "Connectors": [
      {

```

```

        "ConnectorArn": "arn:aws:greengrass:us-west-2::/connectors/SNS/
versions/1",
        "Id": "MySNSConnector",
        "Parameters": {
            "DefaultSNSArn": "arn:aws:sns:us-
west-2:123456789012:GGConnectorTopic"
        }
    ]
},
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
}

```

Weitere Informationen finden Sie unter [Integration mit Diensten und Protokollen mithilfe von Greengrass Connectors](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [GetConnectorDefinitionVersion](#) in der AWS CLI Befehlsreferenz.

get-connector-definition

Das folgende Codebeispiel zeigt die Verwendung `get-connector-definition`.

AWS CLI

Um Informationen über eine Konnektordefinition abzurufen

Im folgenden `get-connector-definition` Beispiel werden Informationen über die angegebene Konnektordefinition abgerufen. Verwenden Sie den IDs `list-connector-definitions` Befehl, um Ihre Konnektordefinitionen abzurufen.

```

aws greengrass get-connector-definition \
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"

```

Ausgabe:

```

{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",

```



```

    "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-
c7c2-4a26-a7e2-7bf478ea2623",
    "Name": "MySNSConnector",
    "tags": {}
}

```

Weitere Informationen finden Sie unter [Integration mit Diensten und Protokollen mithilfe von Greengrass Connectors](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [GetConnectorDefinition](#) in der AWS CLI Befehlsreferenz.

get-core-definition-version

Das folgende Codebeispiel zeigt die Verwendung `get-core-definition-version`.

AWS CLI

Um Details zu einer bestimmten Version der Greengrass-Kerndefinition abzurufen

Im folgenden `get-core-definition-version` Beispiel werden Informationen über die angegebene Version der angegebenen Kerndefinition abgerufen. Verwenden Sie den `list-core-definition-versions` Befehl, um die IDs aller Versionen der Kerndefinition abzurufen. Um die ID der letzten Version abzurufen, die der Kerndefinition hinzugefügt wurde, verwenden Sie den `get-core-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```

aws greengrass get-core-definition-version \
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46" \
  --core-definition-version-id "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"

```

Ausgabe:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeaac3-fd9d-4312-a8fd-ffa9404a20e0",
  "CreationTimestamp": "2019-06-18T16:21:21.351Z",
  "Definition": {
    "Cores": [
      {
        "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/928dea7b82331b47c3ff77b0e763fc5e64e2f7c884e6ef391baed9b6b8e21b45",

```

```

        "Id": "1a39aac7-0885-4417-91f6-23e4cea6c511",
        "SyncShadow": false,
        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
GGGroup4Pi3_Core"
    }
]
},
"Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
"Version": "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"
}

```

- API-Einheiten finden Sie [GetCoreDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

get-core-definition

Das folgende Codebeispiel zeigt die Verwendung `get-core-definition`.

AWS CLI

Um Details für eine Greengrass-Kerndefinition abzurufen

Im folgenden `get-core-definition` Beispiel werden Informationen über die angegebene Kerndefinition abgerufen. Verwenden Sie den IDs `list-core-definitions` Befehl, um Ihre Kerndefinitionen abzurufen.

```

aws greengrass get-core-definition \
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"

```

Ausgabe:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd",
  "CreationTimestamp": "2018-10-18T04:47:06.721Z",
  "Id": "237d6916-27cf-457f-ba0c-e86cfb5d25cd",
  "LastUpdatedTimestamp": "2018-10-18T04:47:06.721Z",
  "LatestVersion": "bd2cd6d4-2bc5-468a-8962-39e071e34b68",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd/versions/
bd2cd6d4-2bc5-468a-8962-39e071e34b68",
  "tags": {}
}

```

- API-Einheiten finden Sie [GetCoreDefinition](#) in der AWS CLI Befehlsreferenz.

get-deployment-status

Das folgende Codebeispiel zeigt die Verwendung `get-deployment-status`.

AWS CLI

Um den Status einer Bereitstellung abzurufen

Im folgenden `get-deployment-status` Beispiel wird der Status für die angegebene Bereitstellung der angegebenen Greengrass-Gruppe abgerufen. Um die Bereitstellungs-ID abzurufen, verwenden Sie den `list-deployments` Befehl und geben Sie die Gruppen-ID an.

```
aws greengrass get-deployment-status \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
  --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

Ausgabe:

```
{
  "DeploymentStatus": "Success",
  "DeploymentType": "NewDeployment",
  "UpdatedAt": "2019-06-18T17:04:44.761Z"
}
```

- API-Einheiten finden Sie [GetDeploymentStatus](#) in der AWS CLI Befehlsreferenz.

get-device-definition-version

Das folgende Codebeispiel zeigt die Verwendung `get-device-definition-version`.

AWS CLI

Um eine Version der Gerätedefinition zu erhalten

Im folgenden `get-device-definition-version` Beispiel werden Informationen über die angegebene Version der angegebenen Gerätedefinition abgerufen. Verwenden Sie den `list-device-definition-versions` Befehl, um die IDs aller Versionen der Gerätedefinition abzurufen. Um die ID der letzten Version abzurufen, die der Gerätedefinition hinzugefügt wurde,

verwenden Sie den `get-device-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```
aws greengrass get-device-definition-version \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
  --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"
```

Ausgabe:

```
{
  "Definition": {
    "Devices": [
      {
        "CertificateArn": "arn:aws:iot:us-west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02",
        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/ExteriorTherm",
        "SyncShadow": true,
        "Id": "ExteriorTherm"
      },
      {
        "CertificateArn": "arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
        "SyncShadow": true,
        "Id": "InteriorTherm"
      }
    ]
  },
  "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
  "CreationTimestamp": "2019-09-11T00:15:09.838Z",
  "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
}
```

- API-Einheiten finden Sie [GetDeviceDefinitionVersion](#) in der AWS CLI Befehlsreferenz.

get-device-definition

Das folgende Codebeispiel zeigt die Verwendung `get-device-definition`.

AWS CLI

Um eine Gerätedefinition zu erhalten

Im folgenden `get-device-definition` Beispiel werden Informationen über die angegebene Gerätedefinition abgerufen. Verwenden Sie den IDs `list-device-definitions` Befehl, um Ihre Gerätedefinitionen abzurufen.

```
aws greengrass get-device-definition \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Ausgabe:

```
{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
  "Name": "TemperatureSensors",
  "tags": {},
  "LastUpdatedTimestamp": "2019-09-11T00:19:03.698Z",
  "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
  "CreationTimestamp": "2019-09-11T00:11:06.197Z",
  "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- API-Einheiten finden Sie [GetDeviceDefinition](#) in der AWS CLI Befehlsreferenz.

get-function-definition-version

Das folgende Codebeispiel zeigt die Verwendung `get-function-definition-version`.

AWS CLI

Um Details zu einer bestimmten Version einer Lambda-Funktion abzurufen

Im Folgenden werden Informationen über die angegebene Version der angegebenen Funktionsdefinition `get-function-definition-version` abgerufen. Verwenden Sie den `list-function-definition-versions` Befehl, um die IDs aller Versionen der Funktionsdefinition abzurufen. Um die ID der letzten Version abzurufen, die der Funktionsdefinition hinzugefügt wurde, verwenden Sie den `get-function-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```
aws greengrass get-function-definition-version \  
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \  
  --function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"
```

Ausgabe:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/9748fda7-1589-4fcc-ac94-  
f5559e88678b",  
  "CreationTimestamp": "2019-06-18T17:04:30.776Z",  
  "Definition": {  
    "Functions": [  
      {  
        "FunctionArn": "arn:aws:lambda::function:GGIPDetector:1",  
        "FunctionConfiguration": {  
          "Environment": {},  
          "MemorySize": 32768,  
          "Pinned": true,  
          "Timeout": 3  
        },  
        "Id": "26b69bdb-e547-46bc-9812-84ec04b6cc8c"  
      },  
      {  
        "FunctionArn": "arn:aws:lambda:us-  
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",  
        "FunctionConfiguration": {  
          "EncodingType": "json",  
          "Environment": {  
            "Variables": {}  
          },  
          "MemorySize": 16384,  
          "Pinned": true,  
          "Timeout": 25  
        },  
      },  
    ],  
  },  
}
```

```

        "Id": "384465a8-eedf-48c6-b793-4c35f7bfae9b"
      }
    ]
  },
  "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
  "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
}

```

- API-Einheiten finden Sie [GetFunctionDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

get-function-definition

Das folgende Codebeispiel zeigt die Verwendung `get-function-definition`.

AWS CLI

Um eine Funktionsdefinition abzurufen

Im folgenden `get-function-definition` Beispiel werden Details für die angegebene Funktionsdefinition angezeigt. Verwenden Sie den IDs `list-function-definitions` Befehl, um Ihre Funktionsdefinitionen abzurufen.

```

aws greengrass get-function-definition \
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"

```

Ausgabe:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
  "CreationTimestamp": "2019-06-18T16:21:21.431Z",
  "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
  "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
  "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
  "tags": {}
}

```

- API-Einheiten finden Sie [GetFunctionDefinition](#) in der AWS CLI Befehlsreferenz.

get-group-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `get-group-certificate-authority`.

AWS CLI

Um die mit einer Greengrass-Gruppe verknüpfte CA abzurufen

Im folgenden `get-group-certificate-authority` Beispiel wird die Zertifizierungsstelle (CA) abgerufen, die der angegebenen Greengrass-Gruppe zugeordnet ist. Um die Zertifizierungsstellen-ID abzurufen, verwenden Sie den `list-group-certificate-authorities` Befehl und geben Sie die Gruppen-ID an.

```
aws greengrass get-group-certificate-authority \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
  --certificate-authority-
id "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
```

Ausgabe:

```
{
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/certificateauthorities/f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
  "GroupCertificateAuthorityId":
  "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
  "PemEncodedCertificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAFICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBWEXAMPLEGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDEXAMPLEEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAEXAMPLESDB25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jEXAMPLENMTewNDI1MjA0NTIxWhcN
MTIwNDI0MjA0EXAMPLEBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWEXAMPLEDASBgNVBAwTC01BTSDB25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWEXAMPLEGkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5EXAMPLE8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CEXAMPLE93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswYEXAMPLEEgpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKEXAMPLEAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n"
```



```
}
```

- API-Einheiten finden Sie [GetGroupCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

get-group-certificate-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-group-certificate-configuration`.

AWS CLI

Um die Konfiguration für die Zertifizierungsstelle abzurufen, die von der Greengrass-Gruppe verwendet wird

Im folgenden `get-group-certificate-configuration` Beispiel wird die Konfiguration für die Zertifizierungsstelle (CA) abgerufen, die von der angegebenen Greengrass-Gruppe verwendet wird.

```
aws greengrass get-group-certificate-configuration \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Ausgabe:

```
{  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "CertificateExpiryInMilliseconds": 604800000,  
  "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"  
}
```

- API-Einheiten finden Sie unter [GetGroupCertificateConfiguration AWS CLI](#) Befehlsreferenz.

get-group-version

Das folgende Codebeispiel zeigt die Verwendung `get-group-version`.

AWS CLI

Um Informationen über eine Version einer Greengrass-Gruppe abzurufen

Im folgenden `get-group-version` Beispiel werden Informationen über die angegebene Version der angegebenen Gruppe abgerufen. Verwenden Sie den `list-group-versions` Befehl, um die IDs aller Versionen der Gruppe abzurufen. Um die ID der letzten Version abzurufen, die

der Gruppe hinzugefügt wurde, verwenden Sie den `get-group` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```
aws greengrass get-group-version \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

Ausgabe:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-  
b77f-8741a4b00e5e",  
  "CreationTimestamp": "2019-06-18T17:04:30.915Z",  
  "Definition": {  
    "CoreDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-  
fd9d-4312-a8fd-ffa9404a20e0",  
    "FunctionDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/  
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",  
    "SubscriptionDefinitionVersionArn": "arn:aws:greengrass:us-  
west-2:123456789012:/greengrass/definition/subscriptions/70e49321-83d5-45d2-  
bc09-81f4917ae152/versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"  
  },  
  "Id": "1013db12-8b58-45ff-acc7-704248f66731",  
  "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"  
}
```

- API-Einheiten finden Sie [GetGroupVersion](#) unter AWS CLI Befehlsreferenz.

get-group

Das folgende Codebeispiel zeigt die Verwendung `get-group`.

AWS CLI

Um Informationen über eine Greengrass-Gruppe abzurufen

Im folgenden `get-group` Beispiel werden Informationen über die angegebene Greengrass-Gruppe abgerufen. Verwenden Sie den Befehl `IDs`, um die Ihrer Gruppen abzurufen. `list-groups`

```
aws greengrass get-group \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Ausgabe:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731",  
  "CreationTimestamp": "2019-06-18T16:21:21.457Z",  
  "Id": "1013db12-8b58-45ff-acc7-704248f66731",  
  "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",  
  "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-  
b77f-8741a4b00e5e",  
  "Name": "GGGroup4Pi3",  
  "tags": {}  
}
```

- API-Einheiten finden Sie [GetGroup](#) in der AWS CLI Befehlsreferenz.

get-logger-definition-version

Das folgende Codebeispiel zeigt die Verwendung `get-logger-definition-version`.

AWS CLI

Um Informationen über eine Version einer Logger-Definition abzurufen

Im folgenden `get-logger-definition-version` Beispiel werden Informationen über die angegebene Version der angegebenen Logger-Definition abgerufen. Verwenden Sie den `list-logger-definition-versions` Befehl, um die IDs aller Versionen der Logger-Definition abzurufen. Um die ID der letzten Version abzurufen, die der Logger-Definition hinzugefügt wurde, verwenden Sie den `get-logger-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```
aws greengrass get-logger-definition-version \  
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \  
  --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
  "CreationTimestamp": "2019-05-08T16:10:13.866Z",
  "Definition": {
    "Loggers": []
  },
  "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
  "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
}
```

- API-Einheiten finden Sie [GetLoggerDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

get-logger-definition

Das folgende Codebeispiel zeigt die Verwendung `get-logger-definition`.

AWS CLI

Um Informationen über eine Logger-Definition abzurufen

Im folgenden `get-logger-definition` Beispiel werden Informationen über die angegebene Logger-Definition abgerufen. Verwenden Sie den IDs `list-logger-definitions` Befehl, um Ihre Logger-Definitionen abzurufen.

```
aws greengrass get-logger-definition \
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
  "CreationTimestamp": "2019-05-08T16:10:13.809Z",
  "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
  "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
  "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
  "tags": {}
}
```

```
}

```

- API-Einheiten finden Sie [GetLoggerDefinition](#) in der AWS CLI Befehlsreferenz.

get-resource-definition-version

Das folgende Codebeispiel zeigt die Verwendung `get-resource-definition-version`.

AWS CLI

Um Informationen über eine bestimmte Version einer Ressourcendefinition abzurufen

Im folgenden `get-resource-definition-version` Beispiel werden Informationen über die angegebene Version der angegebenen Ressourcendefinition abgerufen. Verwenden Sie den `list-resource-definition-versions` Befehl, um die IDs aller Versionen der Ressourcendefinition abzurufen. Um die ID der letzten Version abzurufen, die der Ressourcendefinition hinzugefügt wurde, verwenden Sie den `get-resource-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```
aws greengrass get-resource-definition-version \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
  --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
  "CreationTimestamp": "2019-06-19T16:40:59.392Z",
  "Definition": {
    "Resources": [
      {
        "Id": "26ff3f7b-839a-4217-9fdc-a218308b3963",
        "Name": "usb-port",
        "ResourceDataContainer": {
          "LocalDeviceResourceData": {
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": false
            },
            "SourcePath": "/dev/bus/usb"
          }
        }
      }
    ]
  }
}
```

```

    }
  }
]
},
  "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
}

```

- API-Einheiten finden Sie [GetResourceDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

get-resource-definition

Das folgende Codebeispiel zeigt die Verwendung `get-resource-definition`.

AWS CLI

Um Informationen über eine Ressourcendefinition abzurufen

Im folgenden `get-resource-definition` Beispiel werden Informationen über die angegebene Ressourcendefinition abgerufen. Verwenden Sie den IDs `list-resource-definitions` Befehl, um Ihre Ressourcendefinitionen abzurufen.

```

aws greengrass get-resource-definition \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"

```

Ausgabe:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "CreationTimestamp": "2019-06-19T16:40:59.261Z",
  "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
  "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
  "tags": {}
}

```

- API-Einheiten finden Sie [GetResourceDefinition](#) in der AWS CLI Befehlsreferenz.

get-service-role-for-account

Das folgende Codebeispiel zeigt die Verwendung `get-service-role-for-account`.

AWS CLI

Um die Details für die Servicerolle abzurufen, die Ihrem Konto zugeordnet ist

Im folgenden `get-service-role-for-account` Beispiel werden Informationen über die Servicerolle abgerufen, die Ihrem AWS Konto zugeordnet ist.

```
aws greengrass get-service-role-for-account
```

Ausgabe:

```
{
  "AssociatedAt": "2018-10-18T15:59:20Z",
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Weitere Informationen finden Sie unter [Greengrass Service Role](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [GetServiceRoleForAccount](#) in der AWS CLI Befehlsreferenz.

get-subscription-definition-version

Das folgende Codebeispiel zeigt die Verwendung `get-subscription-definition-version`.

AWS CLI

Um Informationen zu einer bestimmten Version einer Abonnementdefinition abzurufen

Im folgenden `get-subscription-definition-version` Beispiel werden Informationen über die angegebene Version der angegebenen Abonnementdefinition abgerufen. Verwenden Sie den Befehl, um die IDs aller Versionen der Abonnementdefinition abzurufen. `list-subscription-definition-versions` Um die ID der letzten Version abzurufen, die der Abonnementdefinition hinzugefügt wurde, verwenden Sie den `get-subscription-definition` Befehl und überprüfen Sie die `LatestVersion` Eigenschaft.

```
aws greengrass get-subscription-definition-version \
  --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \
```

```
--subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

Ausgabe:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/88ae8699-12ac-4663-
ba3f-4d7f0519140b",
  "CreationTimestamp": "2019-06-18T17:03:52.499Z",
  "Definition": {
    "Subscriptions": [
      {
        "Id": "692c4484-d89f-4f64-8edd-1a041a65e5b6",
        "Source": "arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",
        "Subject": "hello/world",
        "Target": "cloud"
      }
    ]
  },
  "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
  "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
}
```

- API-Einheiten finden Sie [GetSubscriptionDefinitionVersion](#) unter AWS CLI Befehlsreferenz.

get-subscription-definition

Das folgende Codebeispiel zeigt die Verwendung `get-subscription-definition`.

AWS CLI

Um Informationen über eine Abonnementdefinition abzurufen

Im folgenden `get-subscription-definition` Beispiel werden Informationen über die angegebene Abonnementdefinition abgerufen. Verwenden Sie den IDs `list-subscription-definitions` Befehl, um Ihre Abonnementdefinitionen abzurufen.

```
aws greengrass get-subscription-definition \
--subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

Ausgabe:


```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
  "CreationTimestamp": "2019-06-18T17:03:52.392Z",
  "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
  "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
  "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",
  "tags": {}
}
```

- API-Einheiten finden Sie [GetSubscriptionDefinition](#) in der AWS CLI Befehlsreferenz.

get-thing-runtime-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-thing-runtime-configuration`.

AWS CLI

Um die Laufzeitkonfiguration eines Greengrass-Kerns abzurufen

Im folgenden `get-thing-runtime-configuration` Beispiel wird die Laufzeitkonfiguration eines Greengrass-Kerns abgerufen. Bevor Sie die Laufzeitkonfiguration abrufen können, müssen Sie den `update-thing-runtime-configuration` Befehl verwenden, um eine Laufzeitkonfiguration für den Core zu erstellen.

```
aws greengrass get-thing-runtime-configuration \
  --thing-name SampleGreengrassCore
```

Ausgabe:

```
{
  "RuntimeConfiguration": {
    "TelemetryConfiguration": {
      "ConfigurationSyncStatus": "OutOfSync",
      "Telemetry": "On"
    }
  }
}
```

Weitere Informationen finden Sie unter [Konfiguration der Telemetreeinstellungen](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [GetThingRuntimeConfiguration AWS CLI Befehlsreferenz](#).

list-bulk-deployment-detailed-reports

Das folgende Codebeispiel zeigt die Verwendung `list-bulk-deployment-detailed-reports`.

AWS CLI

Um Informationen zu einzelnen Bereitstellungen in einer Massenbereitstellung aufzulisten

Im folgenden `list-bulk-deployment-detailed-reports` Beispiel werden Informationen zu den einzelnen Bereitstellungen in einer Massenbereitstellung angezeigt, einschließlich des Status.

```
aws greengrass list-bulk-deployment-detailed-reports \
  --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

Ausgabe:

```
{
  "Deployments": [
    {
      "DeploymentType": "NewDeployment",
      "DeploymentStatus": "Success",
      "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/
deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/
versions/123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
      "CreatedAt": "2020-01-21T21:34:16.501Z"
    },
    {
      "DeploymentType": "NewDeployment",
      "DeploymentStatus": "InProgress",
      "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/
deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
        "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/versions/a1b2c3d4-5678-90ab-cdef-
EXAMPLE66666",
        "CreatedAt": "2020-01-21T21:34:16.486Z"
    },
    ...
]
}
```

Weitere Informationen finden Sie unter [Create Bulk Deployments for Groups](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [ListBulkDeploymentDetailedReports AWS CLI Befehlsreferenz](#).

list-bulk-deployments

Das folgende Codebeispiel zeigt die Verwendung `list-bulk-deployments`.

AWS CLI

Um Massenbereitstellungen aufzulisten

Das folgende `list-bulk-deployments` Beispiel listet alle Massenbereitstellungen auf.

```
aws greengrass list-bulk-deployments
```

Ausgabe:

```
{
  "BulkDeployments": [
    {
      "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
      "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
      "CreatedAt": "2019-06-25T16:11:33.265Z"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Create Bulk Deployments for Groups](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [ListBulkDeployments AWS CLI Befehlsreferenz](#).

list-connector-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-connector-definition-versions`.

AWS CLI

Um die Versionen aufzulisten, die für eine Connectordefinition verfügbar sind

Im folgenden `list-connector-definition-versions` Beispiel werden die Versionen aufgeführt, die für die angegebene Connectordefinition verfügbar sind. Verwenden Sie den `list-connector-definitions` Befehl, um die Connector-Definition-ID abzurufen.

```
aws greengrass list-connector-definition-versions \  
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Ausgabe:

```
{  
  "Versions": [  
    {  
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-  
c7c2-4a26-a7e2-7bf478ea2623",  
      "CreationTimestamp": "2019-06-19T19:30:01.300Z",  
      "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
      "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Integration mit Diensten und Protokollen mithilfe von Greengrass Connectors](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [ListConnectorDefinitionVersions](#) in der AWS CLI Befehlsreferenz.

list-connector-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-connector-definitions`.

AWS CLI

Um die definierten Greengrass-Konnektoren aufzulisten

Das folgende `list-connector-definitions` Beispiel listet alle Greengrass-Konnektoren auf, die für Ihr AWS Konto definiert sind.

```
aws greengrass list-connector-definitions
```

Ausgabe:

```
{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
      "CreationTimestamp": "2019-06-19T19:30:01.300Z",
      "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
      "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
      "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/
versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",
      "Name": "MySNSConnector"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Integration mit Diensten und Protokollen mithilfe von Greengrass Connectors](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [ListConnectorDefinitions](#) in der AWS CLI Befehlsreferenz.

list-core-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-core-definition-versions`.

AWS CLI

Um die Versionen einer Greengrass-Kerndefinition aufzulisten

Das folgende `list-core-definitions` Beispiel listet alle Versionen der angegebenen Greengrass-Core-Definition auf. Sie können den `list-core-definitions` Befehl verwenden, um die Versions-ID abzurufen.

```
aws greengrass list-core-definition-versions \
```

```
--core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"
```

Ausgabe:

```
{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-c5da-440c-
a97b-084e62593b4c",
      "CreationTimestamp": "2019-06-18T16:14:17.709Z",
      "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
      "Version": "467c36e4-c5da-440c-a97b-084e62593b4c"
    }
  ]
}
```

- API-Einheiten finden Sie [ListCoreDefinitionVersions](#) in der AWS CLI Befehlsreferenz.

list-core-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-core-definitions`.

AWS CLI

Um die Kerndefinitionen von Greengrass aufzulisten

Das folgende `list-core-definitions` Beispiel listet alle Greengrass-Kerndefinitionen für Ihr AWS Konto auf.

```
aws greengrass list-core-definitions
```

Ausgabe:

```
{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d",
      "CreationTimestamp": "2018-10-17T04:30:32.786Z",
```

```

    "Id": "0507843c-c1ef-4f06-b051-817030df7e7d",
    "LastUpdatedTimestamp": "2018-10-17T04:30:32.786Z",
    "LatestVersion": "bcd9e86-3793-491e-93af-3cdfbf4e22b7",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d/versions/bcd9e86-3793-491e-93af-3cdfbf4e22b7"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/31c22500-3509-4271-bafd-cf0655cda438",
    "CreationTimestamp": "2019-06-18T16:24:16.064Z",
    "Id": "31c22500-3509-4271-bafd-cf0655cda438",
    "LastUpdatedTimestamp": "2019-06-18T16:24:16.064Z",
    "LatestVersion": "2f350395-6d09-4c8a-8336-9ae5b57ace84",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/31c22500-3509-4271-bafd-cf0655cda438/versions/2f350395-6d09-4c8a-8336-9ae5b57ace84"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46",
    "CreationTimestamp": "2019-06-18T16:21:21.351Z",
    "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
    "LastUpdatedTimestamp": "2019-06-18T16:21:21.351Z",
    "LatestVersion": "42aeec3-fd9d-4312-a8fd-ffa9404a20e0",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeec3-fd9d-4312-a8fd-ffa9404a20e0"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7",
    "CreationTimestamp": "2019-06-18T16:14:17.709Z",
    "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
    "LastUpdatedTimestamp": "2019-06-18T16:14:17.709Z",
    "LatestVersion": "467c36e4-c5da-440c-a97b-084e62593b4c",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-c5da-440c-a97b-084e62593b4c"
  }
]
}

```

- API-Einheiten finden Sie [ListCoreDefinitions](#) in der AWS CLI Befehlsreferenz.

list-deployments

Das folgende Codebeispiel zeigt die Verwendung `list-deployments`.

AWS CLI

Um die Bereitstellungen für eine Greengrass-Gruppe aufzulisten

Das folgende `list-deployments` Beispiel listet die Bereitstellungen für die angegebene Greengrass-Gruppe auf. Sie können den `list-groups` Befehl verwenden, um Ihre Gruppen-ID nachzuschlagen.

```
aws greengrass list-deployments \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Ausgabe:

```
{
  "Deployments": [
    {
      "CreatedAt": "2019-06-18T17:04:32.702Z",
      "DeploymentId": "1065b8a0-812b-4f21-9d5d-e89b232a530f",
      "DeploymentType": "NewDeployment",
      "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e"
    }
  ]
}
```

- API-Einheiten finden Sie [ListDeployments](#) in der AWS CLI Befehlsreferenz.

list-device-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-device-definition-versions`.

AWS CLI

Um die Versionen einer Gerätedefinition aufzulisten

Im folgenden `list-device-definition-versions` Beispiel werden die Versionen der Gerätedefinitionen angezeigt, die der angegebenen Gerätedefinition zugeordnet sind.


```
aws greengrass list-device-definition-versions \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Ausgabe:

```
{
  "Versions": [
    {
      "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
      "CreationTimestamp": "2019-09-11T00:15:09.838Z",
      "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
    },
    {
      "Version": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
      "CreationTimestamp": "2019-09-11T00:11:06.197Z",
      "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795"
    }
  ]
}
```

- API-Einheiten finden Sie [ListDeviceDefinitionVersions](#) unter AWS CLI Befehlsreferenz.

list-device-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-device-definitions`.

AWS CLI

Um Ihre Gerätedefinitionen aufzulisten

Im folgenden `list-device-definitions` Beispiel werden Details zu den Gerätedefinitionen in Ihrem AWS Konto in der angegebenen AWS Region angezeigt.

```
aws greengrass list-device-definitions \
  --region us-west-2
```

Ausgabe:

```
{
  "Definitions": [
    {
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab/versions/c777b0f5-1059-449b-beaa-f003ebc56c34",
      "LastUpdatedTimestamp": "2019-06-14T15:42:09.059Z",
      "LatestVersion": "c777b0f5-1059-449b-beaa-f003ebc56c34",
      "CreationTimestamp": "2019-06-14T15:42:09.059Z",
      "Id": "50f3274c-3f0a-4f57-b114-6f46085281ab",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab"
    },
    {
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/e01951c9-6134-479a-969a-1a15cac11c40/versions/514d57aa-4ee6-401c-9fac-938a9f7a51e5",
      "Name": "TestDeviceDefinition",
      "LastUpdatedTimestamp": "2019-04-16T23:17:43.245Z",
      "LatestVersion": "514d57aa-4ee6-401c-9fac-938a9f7a51e5",
      "CreationTimestamp": "2019-04-16T23:17:43.245Z",
      "Id": "e01951c9-6134-479a-969a-1a15cac11c40",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/e01951c9-6134-479a-969a-1a15cac11c40"
    },
    {
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
      "Name": "TemperatureSensors",
      "LastUpdatedTimestamp": "2019-09-10T00:19:03.698Z",
      "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
      "CreationTimestamp": "2019-09-11T00:11:06.197Z",
      "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
    }
  ]
}
```

- APIEinzelheiten finden Sie [ListDeviceDefinitions](#) unter AWS CLI Befehlsreferenz.

list-function-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-function-definition-versions`.

AWS CLI

Um die Versionen einer Lambda-Funktion aufzulisten

Das folgende `list-function-definition-versions` Beispiel listet alle Versionen der angegebenen Lambda-Funktion auf. Sie können den `list-function-definitions` Befehl verwenden, um die ID abzurufen.

```
aws greengrass list-function-definition-versions \
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

Ausgabe:

```
{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
      "CreationTimestamp": "2019-06-18T17:04:30.776Z",
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/9b08df77-26f2-4c29-93d2-769715edcfec",
      "CreationTimestamp": "2019-06-18T17:02:44.087Z",
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "Version": "9b08df77-26f2-4c29-93d2-769715edcfec"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/4236239f-94f7-4b90-a2f8-2a24c829d21e",
      "CreationTimestamp": "2019-06-18T17:01:42.284Z",
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "Version": "4236239f-94f7-4b90-a2f8-2a24c829d21e"
    }
  ]
}
```

```

    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/343408bb-549a-4fbe-b043-853643179a39",
      "CreationTimestamp": "2019-06-18T16:21:21.431Z",
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "Version": "343408bb-549a-4fbe-b043-853643179a39"
    }
  ]
}

```

- API-Einheiten finden Sie [ListFunctionDefinitionVersions](#) in der AWS CLI Befehlsreferenz.

list-function-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-function-definitions`.

AWS CLI

Um Lambda-Funktionen aufzulisten

Das folgende `list-function-definitions` Beispiel listet alle Lambda-Funktionen auf, die für Ihr AWS Konto definiert sind.

```
aws greengrass list-function-definitions
```

Ausgabe:

```

{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960",
      "CreationTimestamp": "2018-10-17T04:30:32.884Z",
      "Id": "017970a5-8952-46dd-b1c1-020b3ae8e960",
      "LastUpdatedTimestamp": "2018-10-17T04:30:32.884Z",
      "LatestVersion": "4380b302-790d-4ed8-92bf-02e88afecb15",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960/versions/4380b302-790d-4ed8-92bf-02e88afecb15"
    }
  ]
}

```

```

    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "CreationTimestamp": "2019-06-18T16:21:21.431Z",
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
      "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/6598e653-a262-440c-9967-e2697f64da7b",
      "CreationTimestamp": "2019-06-18T16:24:16.123Z",
      "Id": "6598e653-a262-440c-9967-e2697f64da7b",
      "LastUpdatedTimestamp": "2019-06-18T16:24:16.123Z",
      "LatestVersion": "38bc6ccd-98a2-4ce7-997e-16c84748fae4",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/6598e653-a262-440c-9967-e2697f64da7b/
versions/38bc6ccd-98a2-4ce7-997e-16c84748fae4"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/c668df84-fad2-491b-95f4-655d2cad7885",
      "CreationTimestamp": "2019-06-18T16:14:17.784Z",
      "Id": "c668df84-fad2-491b-95f4-655d2cad7885",
      "LastUpdatedTimestamp": "2019-06-18T16:14:17.784Z",
      "LatestVersion": "37dd68c4-a64f-40ba-aa13-71fecc3ebded",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/c668df84-fad2-491b-95f4-655d2cad7885/
versions/37dd68c4-a64f-40ba-aa13-71fecc3ebded"
    }
  ]
}

```

- API-Einheiten finden Sie [ListFunctionDefinitions](#) in der AWS CLI Befehlsreferenz.

list-group-certificate-authorities

Das folgende Codebeispiel zeigt die Verwendung `list-group-certificate-authorities`.

AWS CLI

Um die aktuellen Daten CAs für eine Gruppe aufzulisten

Das folgende `list-group-certificate-authorities` Beispiel listet die aktuellen Zertifizierungsstellen (CAs) für die angegebene Greengrass-Gruppe auf.

```
aws greengrass list-group-certificate-authorities \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Ausgabe:

```
{  
  "GroupCertificateAuthorities": [  
    {  
      "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-  
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/  
certificateauthorities/  
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",  
      "GroupCertificateAuthorityId":  
      "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [ListGroupCertificateAuthorities AWS CLI Befehlsreferenz](#).

list-group-versions

Das folgende Codebeispiel zeigt die Verwendung `list-group-versions`.

AWS CLI

Um die Versionen einer Greengrass-Gruppe aufzulisten

Das folgende `list-group-versions` Beispiel listet die Versionen der angegebenen Greengrass-Gruppe auf.

```
aws greengrass list-group-versions \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Ausgabe:

```
{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
      "CreationTimestamp": "2019-06-18T17:04:30.915Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/4340669d-
d14d-44e3-920c-46c928750750",
      "CreationTimestamp": "2019-06-18T17:03:52.663Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "4340669d-d14d-44e3-920c-46c928750750"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
versions/1b06e099-2d5b-4f10-91b9-78c4e060f5da",
      "CreationTimestamp": "2019-06-18T17:02:44.189Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "1b06e099-2d5b-4f10-91b9-78c4e060f5da"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/2d3f27f1-3b43-4554-
ab7a-73ec30477efe",
      "CreationTimestamp": "2019-06-18T17:01:42.401Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "2d3f27f1-3b43-4554-ab7a-73ec30477efe"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/d20f7ae9-3444-4c1c-b025-
e2ede23cdd31",
      "CreationTimestamp": "2019-06-18T16:21:21.457Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "d20f7ae9-3444-4c1c-b025-e2ede23cdd31"
    }
  ]
}
```

```
]
}
```

- API-Einheiten finden Sie [ListGroupVersions](#) in der AWS CLI Befehlsreferenz.

list-groups

Das folgende Codebeispiel zeigt die Verwendung `list-groups`.

AWS CLI

Um die Greengrass-Gruppen aufzulisten

Das folgende `list-groups` Beispiel listet alle Greengrass-Gruppen auf, die in Ihrem AWS Konto definiert sind.

```
aws greengrass list-groups
```

Ausgabe:

```
{
  "Groups": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731",
      "CreationTimestamp": "2019-06-18T16:21:21.457Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",
      "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-b77f-8741a4b00e5e",
      "Name": "GGGroup4Pi3"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
      "CreationTimestamp": "2018-10-31T21:52:46.603Z",
      "Id": "1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
      "LastUpdatedTimestamp": "2018-10-31T21:52:46.603Z",
      "LatestVersion": "749af901-60ab-456f-a096-91b12d983c29",

```



```

    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/versions/749af901-60ab-456f-
a096-91b12d983c29",
    "Name": "MyTestGroup"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/504b5c8d-bbed-4635-aff1-48ec5b586db5",
    "CreationTimestamp": "2018-12-31T21:39:36.771Z",
    "Id": "504b5c8d-bbed-4635-aff1-48ec5b586db5",
    "LastUpdatedTimestamp": "2018-12-31T21:39:36.771Z",
    "LatestVersion": "46911e8e-f9bc-4898-8b63-59c7653636ec",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/504b5c8d-bbed-4635-aff1-48ec5b586db5/versions/46911e8e-
f9bc-4898-8b63-59c7653636ec",
    "Name": "smp-ggrass-group"
  }
]
}

```

- API-Einheiten finden Sie [ListGroups](#) in der AWS CLI Befehlsreferenz.

list-logger-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-logger-definition-versions`.

AWS CLI

Um eine Liste von Versionen einer Logger-Definition abzurufen

Im folgenden `list-logger-definition-versions` Beispiel wird eine Liste aller Versionen der angegebenen Logger-Definition abgerufen.

```

aws greengrass list-logger-definition-versions \
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"

```

Ausgabe:

```

{
  "Versions": [
    {

```

```

        "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
        "CreationTimestamp": "2019-05-08T16:10:13.866Z",
        "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
        "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
    },
    {
        "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/3ec6d3af-eb85-48f9-
a16d-1c795fe696d7",
        "CreationTimestamp": "2019-05-08T16:10:13.809Z",
        "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
        "Version": "3ec6d3af-eb85-48f9-a16d-1c795fe696d7"
    }
]
}

```

- API-Einheiten finden Sie [ListLoggerDefinitionVersions](#) in der AWS CLI Befehlsreferenz.

list-logger-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-logger-definitions`.

AWS CLI

Um eine Liste von Logger-Definitionen zu erhalten

Das folgende `list-logger-definitions` Beispiel listet alle Logger-Definitionen für Ihr AWS Konto auf.

```
aws greengrass list-logger-definitions
```

Ausgabe:

```

{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
      "CreationTimestamp": "2019-05-08T16:10:13.809Z",

```

```

    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
    "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/
versions/5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
  }
]
}

```

- API-Einheiten finden Sie [ListLoggerDefinitions](#) in der AWS CLI Befehlsreferenz.

list-resource-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-resource-definition-versions`.

AWS CLI

Um die Versionen einer Ressourcendefinition aufzulisten

Das folgende `list-resource-definition-versions` Beispiel listet die Versionen für die angegebene Greengrass-Ressource auf.

```

aws greengrass list-resource-definition-versions \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"

```

Ausgabe:

```

{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
      "CreationTimestamp": "2019-06-19T16:40:59.392Z",
      "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
      "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/432d92f6-12de-4ec9-a704-619a942a62aa",

```

```

    "CreationTimestamp": "2019-06-19T16:40:59.261Z",
    "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
    "Version": "432d92f6-12de-4ec9-a704-619a942a62aa"
  }
]
}

```

- API-Einheiten finden Sie [ListResourceDefinitionVersions](#) in der AWS CLI Befehlsreferenz.

list-resource-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-resource-definitions`.

AWS CLI

Um die definierten Ressourcen aufzulisten

Das folgende `list-resource-definitions` Beispiel listet die Ressourcen auf, die für die Verwendung von AWS IoT Greengrass definiert sind.

```
aws greengrass list-resource-definitions
```

Ausgabe:

```

{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
      "CreationTimestamp": "2019-06-19T16:40:59.261Z",
      "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
      "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
      "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
      "CreationTimestamp": "2019-06-19T21:51:28.212Z",

```

```

        "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
        "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
        "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
        "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
        "Name": "MyGreengrassResources"
    }
]
}

```

- API-Einheiten finden Sie [ListResourceDefinitions](#) in der AWS CLI Befehlsreferenz.

list-subscription-definition-versions

Das folgende Codebeispiel zeigt die Verwendung `list-subscription-definition-versions`.

AWS CLI

Um die Versionen einer Abonnementdefinition aufzulisten

Das folgende `list-subscription-definition-versions` Beispiel listet alle Versionen des angegebenen Abonnements auf. Sie können den `list-subscription-definitions` Befehl verwenden, um die Abonnement-ID nachzuschlagen.

```

aws greengrass list-subscription-definition-versions \
  --subscription-definition-id 70e49321-83d5-45d2-bc09-81f4917ae152

```

Ausgabe:

```

{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",
      "CreationTimestamp": "2019-06-18T17:03:52.499Z",
      "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
      "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
    },
    {

```

```

        "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/7e320ba3-
c369-4069-a2f0-90acb7f219d6",
        "CreationTimestamp": "2019-06-18T17:03:52.392Z",
        "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
        "Version": "7e320ba3-c369-4069-a2f0-90acb7f219d6"
    }
]
}

```

- API-Einheiten finden Sie [ListSubscriptionDefinitionVersions](#) in der AWS CLI Befehlsreferenz.

list-subscription-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-subscription-definitions`.

AWS CLI

Um eine Liste der Abonnementdefinitionen abzurufen

Das folgende `list-subscription-definitions` Beispiel listet alle AWS IoT Greengrass-Abonnements auf, die in Ihrem AWS Konto definiert sind.

```
aws greengrass list-subscription-definitions
```

Ausgabe:

```

{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
      "CreationTimestamp": "2019-06-18T17:03:52.392Z",
      "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
      "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
      "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
    },
    {

```

```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967",
    "CreationTimestamp": "2018-10-18T15:45:34.024Z",
    "Id": "cd6f1c37-d9a4-4e90-be94-01a7404f5967",
    "LastUpdatedTimestamp": "2018-10-18T15:45:34.024Z",
    "LatestVersion": "d1cf8fac-284f-4f6a-98fe-a2d36d089373",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967/versions/
d1cf8fac-284f-4f6a-98fe-a2d36d089373"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b",
    "CreationTimestamp": "2018-10-22T17:09:31.429Z",
    "Id": "fa81bc84-3f59-4377-a84b-5d0134da359b",
    "LastUpdatedTimestamp": "2018-10-22T17:09:31.429Z",
    "LatestVersion": "086d1b08-b25a-477c-a16f-6f9b3a9c295a",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b/
versions/086d1b08-b25a-477c-a16f-6f9b3a9c295a"
  }
]
}

```

- API-Einheiten finden Sie unter [ListSubscriptionDefinitions AWS CLI Befehlsreferenz](#).

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die an eine Ressource angehängten Tags aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags und ihre Werte auf, die an die angegebene Ressource angehängt sind.

```

aws greengrass list-tags-for-resource \
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"

```

Ausgabe:

```
{
  "tags": {
    "ResourceSubType": "USB",
    "ResourceType": "Device"
  }
}
```

Weitere Informationen finden Sie unter [Tagging Your Greengrass Resources](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in AWS CLI der Befehlsreferenz.

reset-deployments

Das folgende Codebeispiel zeigt die Verwendung `reset-deployments`.

AWS CLI

So bereinigen Sie die Bereitstellungsinformationen für eine Greengrass-Gruppe

Im folgenden `reset-deployments` Beispiel werden die Bereitstellungsinformationen für die angegebene Greengrass-Gruppe bereinigt. Wenn Sie die `hinzufügen--force` option, werden die Bereitstellungsinformationen zurückgesetzt, ohne auf die Antwort des Kerngeräts zu warten.

```
aws greengrass reset-deployments \
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
  --force
```

Ausgabe:

```
{
  "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/
deployments/7dd4e356-9882-46a3-9e28-6d21900c011a",
  "DeploymentId": "7dd4e356-9882-46a3-9e28-6d21900c011a"
}
```

Weitere Informationen finden Sie unter [Reset Deployments](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [ResetDeployments](#) in der AWS CLI Befehlsreferenz.

start-bulk-deployment

Das folgende Codebeispiel zeigt die Verwendung `start-bulk-deployment`.

AWS CLI

Um eine Massenbereitstellung zu starten

Im folgenden `start-bulk-deployment` Beispiel wird eine Massenbereitstellung gestartet, wobei eine in einem S3-Bucket gespeicherte Datei verwendet wird, um die bereitzustellenden Gruppen anzugeben.

```
aws greengrass start-bulk-deployment \  
  --cli-input-json "{\"InputFileUri\":\"https://gg-group-deployment1.s3-us-  
west-2.amazonaws.com/MyBulkDeploymentInputFile.txt\", \"ExecutionRoleArn\":  
\"arn:aws:iam::123456789012:role/ggCreateDeploymentRole\", \"AmznClientToken\":  
\"yourAmazonClientToken\"}"
```

Ausgabe:

```
{  
  "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",  
  "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"  
}
```

Weitere Informationen finden Sie unter [Create Bulk Deployments for Groups](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [StartBulkDeployment AWS CLI](#) Befehlsreferenz.

stop-bulk-deployment

Das folgende Codebeispiel zeigt die Verwendung `stop-bulk-deployment`.

AWS CLI

Um eine Massenbereitstellung zu beenden

Im folgenden `stop-bulk-deployment` Beispiel wird die angegebene Massenbereitstellung beendet. Wenn Sie versuchen, eine Massenbereitstellung zu beenden, die abgeschlossen ist,

wird eine Fehlermeldung angezeigt: `InvalidInputException: Cannot change state of finished execution.`

```
aws greengrass stop-bulk-deployment \  
  --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Create Bulk Deployments for Groups](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [StopBulkDeployment AWS CLI Befehlsreferenz](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags auf eine Ressource anzuwenden

Im folgenden `tag-resource` Beispiel werden zwei Tags, `ResourceType` und `ResourceSubType`, auf die angegebene Greengrass-Ressource angewendet. Dieser Vorgang kann sowohl neue Tags und Werte hinzufügen als auch den Wert vorhandener Tags aktualisieren. Verwenden Sie den `untag-resource` Befehl, um ein Tag zu entfernen.

```
aws greengrass tag-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
  definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658" \  
  --tags "ResourceType=Device,ResourceSubType=USB"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your Greengrass Resources](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag und seinen Wert aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag entfernt, dessen Schlüssel `Category` aus der angegebenen Greengrass-Gruppe stammt. Wenn der Schlüssel für die angegebene Ressource nicht `Category` existiert, wird kein Fehler zurückgegeben.

```
aws greengrass untag-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731" \  
  --tag-keys "Category"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your Greengrass Resources](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in AWS CLI der Befehlsreferenz.

update-connectivity-info

Das folgende Codebeispiel zeigt die Verwendung `update-connectivity-info`.

AWS CLI

Um die Konnektivitätsinformationen für einen Greengrass-Core zu aktualisieren

Das folgende `update-connectivity-info` Beispiel ändert die Endpunkte, über die Geräte eine Verbindung zum angegebenen Greengrass-Core herstellen können. Bei den Konnektivitätsinformationen handelt es sich um eine Liste von IP-Adressen oder Domainnamen mit den entsprechenden Portnummern und optionalen, vom Kunden definierten Metadaten. Möglicherweise müssen Sie die Konnektivitätsinformationen aktualisieren, wenn sich das lokale Netzwerk ändert.

```
aws greengrass update-connectivity-info \  
  --thing-name "MyGroup_Core" \  
  --connectivity-info "[{"Metadata":"","PortNumber":8883,"HostAddress":  
"127.0.0.1","Id":"localhost_127.0.0.1_0"}, {"Metadata":"","PortNumber  
":8883,"HostAddress":"192.168.1.3","Id":"localIP_192.168.1.3"}]"
```

Ausgabe:

```
{
  "Version": "312de337-59af-4cf9-a278-2a23bd39c300"
}
```

- API-Einheiten finden Sie [UpdateConnectivityInfo](#) in der AWS CLI Befehlsreferenz.

update-connector-definition

Das folgende Codebeispiel zeigt die Verwendung `update-connector-definition`.

AWS CLI

Um den Namen für eine Connectordefinition zu aktualisieren

Im folgenden `update-connector-definition` Beispiel wird der Name für die angegebene Connectordefinition aktualisiert. Wenn Sie die Details für den Connector aktualisieren möchten, verwenden Sie den `create-connector-definition-version` Befehl, um eine neue Version zu erstellen.

```
aws greengrass update-connector-definition \
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
  --name "GreengrassConnectors2019"
```

Weitere Informationen finden Sie unter [Integration mit Diensten und Protokollen mithilfe von Konnektoren](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UpdateConnectorDefinition](#) in der AWS CLI Befehlsreferenz.

update-core-definition

Das folgende Codebeispiel zeigt die Verwendung `update-core-definition`.

AWS CLI

Um eine Kerndefinition zu aktualisieren

Im folgenden `update-core-definition` Beispiel wird der Name der angegebenen Kerndefinition geändert. Sie können nur die `name` Eigenschaft einer Kerndefinition aktualisieren.

```
aws greengrass update-core-definition \  
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \  
  --name "MyCoreDevices"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Configure the AWS IoT Greengrass Core](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UpdateCoreDefinition](#) in der AWS CLI Befehlsreferenz.

update-device-definition

Das folgende Codebeispiel zeigt die Verwendung `update-device-definition`.

AWS CLI

Um eine Gerätedefinition zu aktualisieren

Im folgenden `update-device-definition` Beispiel wird der Name der angegebenen Gerätedefinition geändert. Sie können nur die `name` Eigenschaft einer Gerätedefinition aktualisieren.

```
aws greengrass update-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \  
  --name "TemperatureSensors"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UpdateDeviceDefinition](#) in der AWS CLI Befehlsreferenz.

update-function-definition

Das folgende Codebeispiel zeigt die Verwendung `update-function-definition`.

AWS CLI

Um den Namen für eine Funktionsdefinition zu aktualisieren

Im folgenden `update-function-definition` Beispiel wird der Name für die angegebene Funktionsdefinition aktualisiert. Wenn Sie die Details für die Funktion aktualisieren möchten,

verwenden Sie den `create-function-definition-version` Befehl, um eine neue Version zu erstellen.

```
aws greengrass update-function-definition \  
  --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \  
  --name ObsoleteFunction
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Run Local Lambda Functions](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UpdateFunctionDefinition](#) in der AWS CLI Befehlsreferenz.

update-group-certificate-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-group-certificate-configuration`.

AWS CLI

Um den Ablauf der Zertifikate einer Gruppe zu aktualisieren

Im folgenden `update-group-certificate-configuration` Beispiel wird ein Ablauf von 10 Tagen für die für die angegebene Gruppe generierten Zertifikate festgelegt.

```
aws greengrass update-group-certificate-configuration \  
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \  
  --certificate-expiry-in-milliseconds 864000000
```

Ausgabe:

```
{  
  "CertificateExpiryInMilliseconds": 864000000,  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"  
}
```

Weitere Informationen finden Sie unter [AWS IoT Greengrass Security](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UpdateGroupCertificateConfiguration](#) in der AWS CLI Befehlsreferenz.

update-group

Das folgende Codebeispiel zeigt die Verwendung `update-group`.

AWS CLI

Um den Gruppennamen zu aktualisieren

Im folgenden `update-group` Beispiel wird der Name der angegebenen Greengrass-Gruppe aktualisiert. Wenn Sie die Details für die Gruppe aktualisieren möchten, verwenden Sie den `create-group-version` Befehl, um eine neue Version zu erstellen.

```
aws greengrass update-group \  
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
  --name TestGroup4of6
```

Weitere Informationen finden [Sie unter Configure AWS IoT Greengrass on AWS IoT im AWS IoT Greengrass Developer Guide](#).

- API-Einheiten finden Sie [UpdateGroup](#) in der AWS CLI Befehlsreferenz.

update-logger-definition

Das folgende Codebeispiel zeigt die Verwendung `update-logger-definition`.

AWS CLI

Um eine Logger-Definition zu aktualisieren

Im folgenden `update-logger-definition` Beispiel wird der Name der angegebenen Logger-Definition geändert. Sie können nur die `name` Eigenschaft einer Logger-Definition aktualisieren.

```
aws greengrass update-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \  
  --name "LoggingConfigsForSensors"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Monitoring with AWS IoT Greengrass Logs](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UpdateLoggerDefinition](#) in der AWS CLI Befehlsreferenz.

update-resource-definition

Das folgende Codebeispiel zeigt die Verwendung `update-resource-definition`.

AWS CLI

Um den Namen für eine Ressourcendefinition zu aktualisieren

Im folgenden `update-resource-definition` Beispiel wird der Name für die angegebene Ressourcendefinition aktualisiert. Wenn Sie die Details für die Ressource ändern möchten, verwenden Sie den `create-resource-definition-version` Befehl, um eine neue Version zu erstellen.

```
aws greengrass update-resource-definition \  
  --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \  
  --name GreengrassConnectorResources
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Access Local Resources with Lambda Functions and Connectors](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie [UpdateResourceDefinition](#) in der AWS CLI Befehlsreferenz.

update-subscription-definition

Das folgende Codebeispiel zeigt die Verwendung `update-subscription-definition`.

AWS CLI

Um den Namen für eine Abonnementdefinition zu aktualisieren

Im folgenden `update-subscription-definition` Beispiel wird der Name für die angegebene Abonnementdefinition aktualisiert. Wenn Sie die Details für das Abonnement ändern möchten, verwenden Sie den `create-subscription-definition-version` Befehl, um eine neue Version zu erstellen.

```
aws greengrass update-subscription-definition \  
  --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \  
  --name ObsoleteSubscription
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im Titel des Handbuchs.

- API-Einheiten finden Sie [UpdateSubscriptionDefinition](#) in der AWS CLI Befehlsreferenz.

update-thing-runtime-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-thing-runtime-configuration`.

AWS CLI

Um Telemetrie in der Runtime-Konfiguration eines Greengrass-Cores einzuschalten

Das folgende `update-thing-runtime-configuration` Beispiel aktualisiert die Laufzeitkonfiguration eines Greengrass-Kerns, um Telemetrie zu aktivieren.

```
aws greengrass update-thing-runtime-configuration \  
  --thing-name SampleGreengrassCore \  
  --telemetry-configuration {"Telemetry\":"0n\"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfiguration der Telemetrieinstellungen](#) im AWS IoT Greengrass Developer Guide.

- API-Einheiten finden Sie unter [UpdateThingRuntimeConfiguration AWS CLI](#) Befehlsreferenz.

AWS IoT Greengrass V2 Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Greengrass V2.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-service-role-to-account

Das folgende Codebeispiel zeigt die Verwendung `associate-service-role-to-account`.

AWS CLI

Um die Greengrass-Servicerolle Ihrem AWS Konto zuzuordnen

Das folgende `associate-service-role-to-account` Beispiel ordnet AWS IoT Greengrass eine Servicerolle für Ihr AWS Konto zu.

```
aws greengrassv2 associate-service-role-to-account \  
  --role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

Ausgabe:

```
{  
  "associatedAt": "2022-01-19T19:21:53Z"  
}
```

Weitere Informationen finden Sie unter [Greengrass-Servicerolle](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [AssociateServiceRoleToAccount AWS CLI Befehlsreferenz](#).

batch-associate-client-device-with-core-device

Das folgende Codebeispiel zeigt die Verwendung `batch-associate-client-device-with-core-device`.

AWS CLI

Um Client-Geräte einem Core-Gerät zuzuordnen

Im folgenden `batch-associate-client-device-with-core-device` Beispiel werden zwei Client-Geräte einem Core-Gerät zugeordnet.

```
aws greengrassv2 batch-associate-client-device-with-core-device \  
  --core-device-thing-name MyGreengrassCore \  
  --client-device-thing-names MyGreengrassClient1 MyGreengrassClient2
```

```
--entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Ausgabe:

```
{  
  "errorEntries": []  
}
```

Weitere Informationen finden Sie unter [Interagieren mit lokalen IoT-Geräten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [BatchAssociateClientDeviceWithCoreDevice AWS CLI-Befehlsreferenz](#).

batch-disassociate-client-device-from-core-device

Das folgende Codebeispiel zeigt die Verwendung `batch-disassociate-client-device-from-core-device`.

AWS CLI

Um Client-Geräte von einem Core-Gerät zu trennen

Im folgenden `batch-disassociate-client-device-from-core-device` Beispiel wird die Verbindung zwischen zwei Client-Geräten und einem Kerngerät getrennt.

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \  
  --core-device-thing-name MyGreengrassCore \  
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Ausgabe:

```
{  
  "errorEntries": []  
}
```

Weitere Informationen finden Sie unter [Interagieren mit lokalen IoT-Geräten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [BatchDisassociateClientDeviceFromCoreDevice AWS CLI-Befehlsreferenz](#).

cancel-deployment

Das folgende Codebeispiel zeigt die Verwendung `cancel-deployment`.

AWS CLI

Um eine Bereitstellung abzurechnen

Das folgende `cancel-deployment` Beispiel beendet eine kontinuierliche Bereitstellung für eine Dinggruppe.

```
aws greengrassv2 cancel-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "message": "SUCCESS"  
}
```

Weitere Informationen finden Sie unter [Bereitstellungen stornieren](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [CancelDeployment AWS CLI](#) Befehlsreferenz.

create-component-version

Das folgende Codebeispiel zeigt die Verwendung `create-component-version`.

AWS CLI

Beispiel 1: Um eine Komponentenversion aus einem Rezept zu erstellen

Im folgenden `create-component-version` Beispiel wird eine Version einer Hello World-Komponente aus einer Rezeptdatei erstellt.

```
aws greengrassv2 create-component-version \  
  --inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

Inhalt von `com.example.HelloWorld-1.0.0.json`:

```
{
  "RecipeFormatVersion": "2020-01-25",
  "ComponentName": "com.example.HelloWorld",
  "ComponentVersion": "1.0.0",
  "ComponentDescription": "My first AWS IoT Greengrass component.",
  "ComponentPublisher": "Amazon",
  "ComponentConfiguration": {
    "DefaultConfiguration": {
      "Message": "world"
    }
  },
  "Manifests": [
    {
      "Platform": {
        "os": "linux"
      },
      "Lifecycle": {
        "Run": "echo 'Hello {configuration:/Message}'"
      }
    }
  ]
}
```

Ausgabe:

```
{
  "arn": "arn:aws:greengrass:us-west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
  "componentName": "com.example.HelloWorld",
  "componentVersion": "1.0.0",
  "creationTimestamp": "2021-01-07T16:24:33.650000-08:00",
  "status": {
    "componentState": "REQUESTED",
    "message": "NONE",
    "errors": {}
  }
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Komponenten erstellen](#) und [Komponenten zur Bereitstellung hochladen](#) im AWS IoT Greengrass V2 Developer Guide.

Beispiel 2: So erstellen Sie eine Komponentenversion aus einer AWS Lambda-Funktion

Im folgenden `create-component-version` Beispiel wird eine Version einer Hello World-Komponente aus einer AWS Lambda-Funktion erstellt.

```
aws greengrassv2 create-component-version \  
  --cli-input-json file://lambda-function-component.json
```

Inhalt von `lambda-function-component.json`:

```
{  
  "lambdaFunction": {  
    "lambdaArn": "arn:aws:lambda:us-  
west-2:123456789012:function:HelloWorldPythonLambda:1",  
    "componentName": "com.example.HelloWorld",  
    "componentVersion": "1.0.0",  
    "componentLambdaParameters": {  
      "eventSources": [  
        {  
          "topic": "hello/world/+",  
          "type": "IOT_CORE"  
        }  
      ]  
    }  
  }  
}
```

Ausgabe:

```
{  
  "arn": "arn:aws:greengrass:us-  
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",  
  "componentName": "com.example.HelloWorld",  
  "componentVersion": "1.0.0",  
  "creationTimestamp": "2021-01-07T17:05:27.347000-08:00",  
  "status": {  
    "componentState": "REQUESTED",  
    "message": "NONE",  
    "errors": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Ausführen von AWS Lambda-Funktionen](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [CreateComponentVersion AWS CLI-Befehlsreferenz](#).

create-deployment

Das folgende Codebeispiel zeigt die Verwendung `create-deployment`.

AWS CLI

Beispiel 1: Um ein Deployment zu erstellen

Im folgenden `create-deployment` Beispiel wird die AWS IoT Greengrass-Befehlszeilenschnittstelle auf einem Kerngerät bereitgestellt.

```
aws greengrassv2 create-deployment \  
  --cli-input-json file://cli-deployment.json
```

Inhalt von `cli-deployment.json`:

```
{  
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/MyGreengrassCore",  
  "deploymentName": "Deployment for MyGreengrassCore",  
  "components": {  
    "aws.greengrass.Cli": {  
      "componentVersion": "2.0.3"  
    }  
  },  
  "deploymentPolicies": {  
    "failureHandlingPolicy": "DO_NOTHING",  
    "componentUpdatePolicy": {  
      "timeoutInSeconds": 60,  
      "action": "NOTIFY_COMPONENTS"  
    },  
    "configurationValidationPolicy": {  
      "timeoutInSeconds": 60  
    }  
  },  
  "iotJobConfiguration": {}  
}
```

Ausgabe:

```
{
```

```
"deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Weitere Informationen finden Sie unter [Create Deployments](#) im AWS IoT Greengrass V2 Developer Guide.

Beispiel 2: So erstellen Sie eine Bereitstellung, die Komponentenkonfigurationen aktualisiert

Im folgenden `create-deployment` Beispiel wird die Nucleus-Komponente AWS IoT Greengrass für eine Gruppe von Kerngeräten bereitgestellt. Bei dieser Bereitstellung werden die folgenden Konfigurationsupdates für die Nucleus-Komponente angewendet:

Setzt die Proxyeinstellungen der Zielgeräte auf ihre Standardeinstellungen zurück (keine Proxyeinstellungen). Setzt die Einstellungen der Zielgeräte MQTT auf ihre Standardwerte zurück. Legt die JVM Optionen für den Nucleus fest. Legt die Protokollierungsebene für den JVM Nucleus fest.

```
aws greengrassv2 create-deployment \
  --cli-input-json file://nucleus-deployment.json
```

Inhalt von `nucleus-deployment.json`:

```
{
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
  "deploymentName": "Deployment for MyGreengrassCoreGroup",
  "components": {
    "aws.greengrass.Nucleus": {
      "componentVersion": "2.0.3",
      "configurationUpdate": {
        "reset": [
          "/networkProxy",
          "/mqtt"
        ],
        "merge": "{\"jvmOptions\": \"-Xmx64m\", \"logging\": {\"level\": \"WARN
\n\"}}\"
      }
    }
  },
  "deploymentPolicies": {
    "failureHandlingPolicy": "ROLLBACK",
    "componentUpdatePolicy": {
```



```
        "timeoutInSeconds": 60,
        "action": "NOTIFY_COMPONENTS"
    },
    "configurationValidationPolicy": {
        "timeoutInSeconds": 60
    }
},
"iotJobConfiguration": {}
}
```

Ausgabe:

```
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Weitere Informationen finden Sie unter [Bereitstellungen erstellen](#) und [Komponentenkonfigurationen aktualisieren](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [CreateDeployment AWS CLI Befehlsreferenz](#).

delete-component

Das folgende Codebeispiel zeigt die Verwendung `delete-component`.

AWS CLI

Um eine Komponentenversion zu löschen

Im folgenden `delete-component` Beispiel wird eine Hello World-Komponente gelöscht.

```
aws greengrassv2 delete-component \
  --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [DeleteComponent](#) in der AWS CLI Befehlsreferenz.

delete-core-device

Das folgende Codebeispiel zeigt die Verwendung `delete-core-device`.

AWS CLI

Um ein Core-Gerät zu löschen

Im folgenden `delete-core-device` Beispiel wird ein AWS IoT-Greengrass-Core-Gerät gelöscht.

```
aws greengrassv2 delete-core-device \  
  --core-device-thing-name MyGreengrassCore
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Deinstallieren der AWS IoT Greengrass Core-Software](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [DeleteCoreDevice](#) in der AWS CLI Befehlsreferenz.

describe-component

Das folgende Codebeispiel zeigt die Verwendung `describe-component`.

AWS CLI

Um eine Komponentenversion zu beschreiben

Das folgende `describe-component` Beispiel beschreibt eine Hello World-Komponente.

```
aws greengrassv2 describe-component \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

Ausgabe:

```
{
```

```
"arn": "arn:aws:greengrass:us-west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
"componentName": "com.example.HelloWorld",
"componentVersion": "1.0.0",
"creationTimestamp": "2021-01-07T17:12:11.133000-08:00",
"publisher": "Amazon",
"description": "My first AWS IoT Greengrass component.",
"status": {
  "componentState": "DEPLOYABLE",
  "message": "NONE",
  "errors": {}
},
"platforms": [
  {
    "attributes": {
      "os": "linux"
    }
  }
]
}
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [DescribeComponent](#) in der AWS CLI Befehlsreferenz.

disassociate-service-role-from-account

Das folgende Codebeispiel zeigt die Verwendung `disassociate-service-role-from-account`.

AWS CLI

Um die Greengrass-Service-Rolle von Ihrem Konto zu trennen AWS

Im folgenden `disassociate-service-role-from-account` Beispiel wird die Greengrass-Service-Rolle für Ihr Konto von AWS IoT Greengrass getrennt. AWS

```
aws greengrassv2 disassociate-service-role-from-account
```

Ausgabe:

```
{
```

```
"disassociatedAt": "2022-01-19T19:26:09Z"
}
```

Weitere Informationen finden Sie unter [Greengrass-Servicerolle](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [DisassociateServiceRoleFromAccount AWS CLI](#) Befehlsreferenz.

get-component-version-artifact

Das folgende Codebeispiel zeigt die Verwendung `get-component-version-artifact`.

AWS CLI

URL Um ein Komponentenartefakt herunterzuladen

Im folgenden `get-component-version-artifact` Beispiel wird die Datei URL der lokalen Debug-Konsolenkomponente heruntergeladen. JAR

```
aws greengrassv2 get-component-version-artifact \
  --arn arn:aws:greengrass:us-west-2:aws:components:aws.greengrass.LocalDebugConsole:versions:2.0.3 \
  --artifact-name "Uvt6ZEzQ9TKiAuLbfXBX_APdY0TWks3uc46tHFHTzBM=/aws.greengrass.LocalDebugConsole.jar"
```

Ausgabe:

```
{
  "preSignedUrl": "https://evergreencomponentmanagem-
  artifactbucket7410c9ef-g18n1iya8kwr.s3.us-west-2.amazonaws.com/public/
  aws.greengrass.LocalDebugConsole/2.0.3/s3/ggv2-component-releases-prod-pdx/
  EvergreenHttpDebugView/2ffc496ba41b39568968b22c582b4714a937193ee7687a45527238e696672521/
  aws.greengrass.LocalDebugConsole/aws.greengrass.LocalDebugConsole.jar?X-Amz-
  Security-Token=KwflKSdEXAMPLE..."
}
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [GetComponentVersionArtifact](#) in der AWS CLI Befehlsreferenz.

get-component

Das folgende Codebeispiel zeigt die Verwendung get-component.

AWS CLI

Beispiel 1: Um das Rezept einer Komponente im YAML Format herunterzuladen (Linux, macOS oder Unix)

Im folgenden get-component Beispiel wird das Rezept einer Hello World-Komponente in eine Datei im YAML Format heruntergeladen. Der Befehl hat folgende Auswirkungen:

Verwendet die --query Parameter --output und, um die Ausgabe des Befehls zu steuern. Diese Parameter extrahieren den Rezept-Blob aus der Ausgabe des Befehls. Weitere Informationen zur Steuerung der Ausgabe finden Sie unter [Steuern der Befehlsausgabe](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle. Verwendet das base64 Hilfsprogramm. Dieses Tool dekodiert den extrahierten Blob in den Originaltext. Bei dem Blob, der bei einem erfolgreichen get-component Befehl zurückgegeben wird, handelt es sich um Base64-codierten Text. Sie müssen dieses Blob dekodieren, um den Originaltext zu erhalten. Speichert den dekodierten Text in einer Datei. Der letzte Abschnitt des Befehls (> com.example>HelloWorld-1.0.0.json) speichert den dekodierten Text in einer Datei.

```
aws greengrassv2 get-component \  
  --arn arn:aws:greengrass:us-west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0 \  
  --recipe-output-format YAML \  
  --query recipe \  
  --output text | base64 --decode > com.example>HelloWorld-1.0.0.json
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

Beispiel 2: So laden Sie das Rezept einer Komponente im YAML Format herunter (WindowsCMD)

Im folgenden get-component Beispiel wird das Rezept einer Hello World-Komponente in eine Datei im YAML Format heruntergeladen. Dieser Befehl verwendet das certutil Hilfsprogramm.

```
aws greengrassv2 get-component ^  
  --arn arn:aws:greengrass:us-west-2:675946970638:components:com.example>HelloWorld:versions:1.0.0 ^  
  --recipe-output-format YAML ^
```

```
--query recipe ^
--output text > com.example.HelloWorld-1.0.0.yaml.b64
```

```
certutil -
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

Beispiel 3: So laden Sie das Rezept einer Komponente im YAML Format herunter (Windows PowerShell)

Im folgenden `get-component` Beispiel wird das Rezept einer Hello World-Komponente in eine Datei im YAML Format heruntergeladen. Dieser Befehl verwendet das `certutil` Hilfsprogramm.

```
aws greengrassv2 get-component `
  --arn arn:aws:greengrass:us-
west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 `
  --recipe-output-format YAML `
  --query recipe `
  --output text > com.example.HelloWorld-1.0.0.yaml.b64
```

```
certutil -
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [GetComponent](#) in der AWS CLI Befehlsreferenz.

get-connectivity-info

Das folgende Codebeispiel zeigt die Verwendung `get-connectivity-info`.

AWS CLI

Um die Konnektivitätsinformationen für ein Greengrass-Core-Gerät abzurufen

Im folgenden `get-connectivity-info` Beispiel werden die Konnektivitätsinformationen für ein Greengrass-Core-Gerät abgerufen. Client-Geräte verwenden diese Informationen, um eine Verbindung zu dem MQTT Broker herzustellen, der auf diesem Kerngerät ausgeführt wird.

```
aws greengrassv2 get-connectivity-info \  
  --thing-name MyGreengrassCore
```

Ausgabe:

```
{  
  "connectivityInfo": [  
    {  
      "id": "localIP_192.0.2.0",  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Manage Core Device Endpoints](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [GetConnectivityInfo AWS CLI](#) Befehlsreferenz.

get-core-device

Das folgende Codebeispiel zeigt die Verwendung `get-core-device`.

AWS CLI

Um ein Kerngerät zu bekommen

Das folgende `get-core-device` Beispiel ruft Informationen über ein AWS IoT-Greengrass-Core-Gerät ab.

```
aws greengrassv2 get-core-device \  
  --core-device-thing-name MyGreengrassCore
```

Ausgabe:

```
{  
  "coreDeviceThingName": "MyGreengrassCore",  
  "coreVersion": "2.0.3",  
  "platform": "linux",  
  "architecture": "amd64",
```

```

    "status": "HEALTHY",
    "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",
    "tags": {}
  }

```

Weitere Informationen finden [Sie unter Überprüfen des Status des Kerngeräts](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [GetCoreDevice AWS CLI Befehlsreferenz](#).

get-deployment

Das folgende Codebeispiel zeigt die Verwendung `get-deployment`.

AWS CLI

Um ein Deployment zu erhalten

Das folgende `get-deployment` Beispiel enthält Informationen zur Bereitstellung der AWS IoT Greengrass Nucleus-Komponente auf einer Gruppe von Kerngeräten.

```

aws greengrassv2 get-deployment \
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/MyGreengrassCoreGroup",
  "revisionId": "14",
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "deploymentName": "Deployment for MyGreengrassCoreGroup",
  "deploymentStatus": "ACTIVE",
  "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "components": {
    "aws.greengrass.Nucleus": {
      "componentVersion": "2.0.3",
      "configurationUpdate": {
        "merge": "{\"jvmOptions\":\"-Xmx64m\", \"logging\":{\"level\":\"WARN\"}}",
        "reset": [

```



```
        "/networkProxy",
        "/mqtt"
    ]
  }
},
"deploymentPolicies": {
  "failureHandlingPolicy": "ROLLBACK",
  "componentUpdatePolicy": {
    "timeoutInSeconds": 60,
    "action": "NOTIFY_COMPONENTS"
  },
  "configurationValidationPolicy": {
    "timeoutInSeconds": 60
  }
},
"iotJobConfiguration": {},
"creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
"isLatestForTarget": false,
"tags": {}
}
```

Weitere Informationen finden Sie unter [Bereitstellen von Komponenten auf Geräten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [GetDeployment AWS CLI Befehlsreferenz](#).

get-service-role-for-account

Das folgende Codebeispiel zeigt die Verwendung `get-service-role-for-account`.

AWS CLI

Um die Greengrass-Service-Rolle für Ihr Konto zu AWS erhalten

Im folgenden `get-service-role-for-account` Beispiel wird die Service-Rolle abgerufen, die AWS IoT Greengrass für Ihr AWS-Konto zugeordnet ist.

```
aws greengrassv2 get-service-role-for-account
```

Ausgabe:

```
{
```

```
"associatedAt": "2022-01-19T19:21:53Z",
"roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Weitere Informationen finden Sie unter [Greengrass-Servicerolle](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [GetServiceRoleForAccount AWS CLI Befehlsreferenz](#).

list-client-devices-associated-with-core-device

Das folgende Codebeispiel zeigt die Verwendung `list-client-devices-associated-with-core-device`.

AWS CLI

Um die Client-Geräte aufzulisten, die einem Core-Gerät zugeordnet sind

Das folgende `list-client-devices-associated-with-core-device` Beispiel listet alle Client-Geräte auf, die einem Kerngerät zugeordnet sind.

```
aws greengrassv2 list-client-devices-associated-with-core-device \
  --core-device-thing-name MyTestGreengrassCore
```

Ausgabe:

```
{
  "associatedClientDevices": [
    {
      "thingName": "MyClientDevice2",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    },
    {
      "thingName": "MyClientDevice1",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Interagieren mit lokalen IoT-Geräten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [ListClientDevicesAssociatedWithCoreDevice AWS CLI-Befehlsreferenz](#).

list-component-versions

Das folgende Codebeispiel zeigt die Verwendung `list-component-versions`.

AWS CLI

Um die Versionen einer Komponente aufzulisten

Das folgende `list-component-versions` Beispiel listet alle Versionen einer Hello World-Komponente auf.

```
aws greengrassv2 list-component-versions \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld
```

Ausgabe:

```
{  
  "componentVersions": [  
    {  
      "componentName": "com.example>HelloWorld",  
      "componentVersion": "1.0.1",  
      "arn": "arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.1"  
    },  
    {  
      "componentName": "com.example>HelloWorld",  
      "componentVersion": "1.0.0",  
      "arn": "arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [ListComponentVersions](#) in der AWS CLI Befehlsreferenz.

list-components

Das folgende Codebeispiel zeigt die Verwendung `list-components`.

AWS CLI

Um Komponenten aufzulisten

Im folgenden `list-components` Beispiel werden alle Komponenten und ihre neueste Version aufgeführt, die in Ihrem AWS Konto in der aktuellen Region definiert ist.

```
aws greengrassv2 list-components
```

Ausgabe:

```
{
  "components": [
    {
      "arn": "arn:aws:greengrass:us-west-2:123456789012:components:com.example.HelloWorld",
      "componentName": "com.example.HelloWorld",
      "latestVersion": {
        "arn": "arn:aws:greengrass:us-west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1",
        "componentVersion": "1.0.1",
        "creationTimestamp": "2021-01-08T16:51:07.352000-08:00",
        "description": "My first AWS IoT Greengrass component.",
        "publisher": "Amazon",
        "platforms": [
          {
            "attributes": {
              "os": "linux"
            }
          }
        ]
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Komponenten verwalten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [ListComponents](#) in der AWS CLI Befehlsreferenz.

list-core-devices

Das folgende Codebeispiel zeigt die Verwendung `list-core-devices`.

AWS CLI

Um die wichtigsten Geräte aufzulisten

Das folgende `list-core-devices` Beispiel listet die AWS IoT-Greengrass-Kerngeräte in Ihrem AWS Konto in der aktuellen Region auf.

```
aws greengrassv2 list-core-devices
```

Ausgabe:

```
{
  "coreDevices": [
    {
      "coreDeviceThingName": "MyGreengrassCore",
      "status": "HEALTHY",
      "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"
    }
  ]
}
```

Weitere Informationen finden [Sie unter Überprüfen des Status des Kerngeräts](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [ListCoreDevices AWS CLI](#) Befehlsreferenz.

list-deployments

Das folgende Codebeispiel zeigt die Verwendung `list-deployments`.

AWS CLI

Um Bereitstellungen aufzulisten

Im folgenden `list-deployments` Beispiel wird die neueste Version jeder Bereitstellung aufgeführt, die in Ihrem AWS Konto in der aktuellen Region definiert ist.

```
aws greengrassv2 list-deployments
```

Ausgabe:

```
{
  "deployments": [
    {
      "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
      "revisionId": "14",
      "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "deploymentName": "Deployment for MyGreengrassCoreGroup",
      "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
      "deploymentStatus": "ACTIVE",
      "isLatestForTarget": false
    },
    {
      "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
      "revisionId": "1",
      "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "deploymentName": "Deployment for MyGreengrassCore",
      "creationTimestamp": "2021-01-06T16:10:42.407000-08:00",
      "deploymentStatus": "COMPLETED",
      "isLatestForTarget": false
    }
  ]
}
```

Weitere Informationen finden Sie unter [Bereitstellen von Komponenten auf Geräten](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [ListDeployments AWS CLI-Befehlsreferenz](#).

list-effective-deployments

Das folgende Codebeispiel zeigt die Verwendung `list-effective-deployments`.

AWS CLI

Um Bereitstellungsaufträge aufzulisten

Das folgende `list-effective-deployments` Beispiel listet die Bereitstellungen auf, die für ein AWS IoT-Greengrass-Core-Gerät gelten.

```
aws greengrassv2 list-effective-deployments \  
  --core-device-thing-name MyGreengrassCore
```

Ausgabe:

```
{  
  "effectiveDeployments": [  
    {  
      "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "deploymentName": "Deployment for MyGreengrassCore",  
      "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
      "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/  
MyGreengrassCore",  
      "coreDeviceExecutionStatus": "COMPLETED",  
      "reason": "SUCCESSFUL",  
      "creationTimestamp": "2021-01-06T16:10:42.442000-08:00",  
      "modifiedTimestamp": "2021-01-08T17:21:27.830000-08:00"  
    },  
    {  
      "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "deploymentName": "Deployment for MyGreengrassCoreGroup",  
      "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",  
      "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE44444",  
      "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/  
MyGreengrassCoreGroup",  
      "coreDeviceExecutionStatus": "SUCCEEDED",  
      "reason": "SUCCESSFUL",  
      "creationTimestamp": "2021-01-07T17:19:20.394000-08:00",  
      "modifiedTimestamp": "2021-01-07T17:21:20.721000-08:00"  
    }  
  ]  
}
```

Weitere Informationen finden [Sie unter Überprüfen des Status des Kerngeräts](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [ListEffectiveDeployments AWS CLI](#) Befehlsreferenz.

list-installed-components

Das folgende Codebeispiel zeigt die Verwendung `list-installed-components`.

AWS CLI

Um die auf einem Core-Gerät installierten Komponenten aufzulisten

Das folgende `list-installed-components` Beispiel listet die Komponenten auf, die auf einem AWS IoT Greengrass-Core-Gerät installiert sind.

```
aws greengrassv2 list-installed-components \  
  --core-device-thing-name MyGreengrassCore
```

Ausgabe:

```
{  
  "installedComponents": [  
    {  
      "componentName": "aws.greengrass.Cli",  
      "componentVersion": "2.0.3",  
      "lifecycleState": "RUNNING",  
      "isRoot": true  
    },  
    {  
      "componentName": "aws.greengrass.Nucleus",  
      "componentVersion": "2.0.3",  
      "lifecycleState": "FINISHED",  
      "isRoot": true  
    }  
  ]  
}
```

Weitere Informationen finden [Sie unter Überprüfen des Status des Kerngeräts](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [ListInstalledComponents AWS CLI Befehlsreferenz](#).

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für ein AWS IoT Greengrass-Core-Gerät auf.

```
aws greengrassv2 list-tags-for-resource \  
  --resource-arn arn:aws:greengrass:us-  
west-2:123456789012:coreDevices:MyGreengrassCore
```

Ausgabe:

```
{  
  "tags": {  
    "Owner": "richard-roe"  
  }  
}
```

Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einer Ressource einen Tag hinzu

Das folgende `tag-resource` Beispiel fügt einem AWS IoT-Greengrass-Core-Gerät ein Eigentümergebiet hinzu. Sie können dieses Tag verwenden, um den Zugriff auf das Kerngerät anhand dessen zu steuern, wem es gehört.

```
aws greengrassv2 tag-resource \  
  --resource-arn arn:aws:greengrass:us-  
west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tags Owner=richard-roe
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird ein Besitzer-Tag von einem AWS IoT-Greengrass-Core-Gerät entfernt.

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tag-keys Owner
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-connectivity-info

Das folgende Codebeispiel zeigt die Verwendung `update-connectivity-info`.

AWS CLI

Um die Verbindungsinformationen für ein Greengrass Core-Gerät zu aktualisieren

Im folgenden `update-connectivity-info` Beispiel werden die Konnektivitätsinformationen für ein Greengrass-Core-Gerät abgerufen. Client-Geräte verwenden diese Informationen, um eine Verbindung zu dem MQTT Broker herzustellen, der auf diesem Kerngerät ausgeführt wird.

```
aws greengrassv2 update-connectivity-info \  
  --thing-name MyGreengrassCore \  
  --cli-input-json file://core-device-connectivity-info.json
```

Inhalt von `core-device-connectivity-info.json`:

```
{  
  "connectivityInfo": [  
    {  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883,  
      "id": "localIP_192.0.2.0"  
    }  
  ]  
}
```

Ausgabe:

```
{  
  "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Manage Core Device Endpoints](#) im AWS IoT Greengrass V2 Developer Guide.

- API-Einheiten finden Sie unter [UpdateConnectivityInfo AWS CLI Befehlsreferenz](#).

AWS IoT Jobs SDK release Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Jobs SDK release.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-job-execution

Das folgende Codebeispiel zeigt die Verwendung `describe-job-execution`.

AWS CLI

Um die Details einer Jobausführung abzurufen

Im folgenden `describe-job-execution` Beispiel werden die Details der letzten Ausführung des angegebenen Jobs und Dings abgerufen.

```
aws iot-jobs-data describe-job-execution \  
  --job-id SampleJob \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "execution": {  
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567701875.743,  
    "queuedAt": 1567701902.444,  
    "status": "QUEUED",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

Weitere Informationen finden Sie unter [Geräte und Jobs](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [DescribeJobExecution](#) in der AWS CLI Befehlsreferenz.

get-pending-job-executions

Das folgende Codebeispiel zeigt die Verwendung `get-pending-job-executions`.

AWS CLI

Um eine Liste aller Jobs abzurufen, die sich für ein Ding nicht im Terminalstatus befinden

Im folgenden `get-pending-job-executions` Beispiel wird eine Liste aller Jobs angezeigt, die sich für das angegebene Ding nicht im Terminalstatus befinden.

```
aws iot-jobs-data get-pending-job-executions \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "InProgressJobs": [  
  ],  
  "queuedJobs": [  
    {  
      "executionNumber": 2939653338,  
      "jobId": "SampleJob",  
      "lastUpdatedAt": 1567701875.743,  
      "queuedAt": 1567701902.444,  
      "versionNumber": 3  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Geräte und Jobs](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [GetPendingJobExecutions](#) in der AWS CLI Befehlsreferenz.

start-next-pending-job-execution

Das folgende Codebeispiel zeigt die Verwendung `start-next-pending-job-execution`.

AWS CLI

Um die nächste ausstehende Jobausführung für eine Sache abzurufen und zu starten

Im folgenden `start-next-pending-job-execution` Beispiel wird die nächste Auftragsausführung abgerufen und gestartet, deren Status `IN_PROGRESS` oder `QUEUED` für das angegebene Ding ist.

```
aws iot-jobs-data start-next-pending-job-execution \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "execution": {  
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567714853.743,  
    "queuedAt": 1567701902.444,  
    "startedAt": 1567714871.690,  
    "status": "IN_PROGRESS",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

Weitere Informationen finden Sie unter [Geräte und Jobs](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [StartNextPendingJobExecution](#) in der AWS CLI Befehlsreferenz.

update-job-execution

Das folgende Codebeispiel zeigt die Verwendung `update-job-execution`.

AWS CLI

Um den Status einer Jobausführung zu aktualisieren

Im folgenden `update-job-execution` Beispiel wird der Status des angegebenen Jobs und Dings aktualisiert.

```
aws iot-jobs-data update-job-execution \  
  --job-id SampleJob \  
  --thing-name MotionSensor1 \  
  --status REMOVED \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Ausgabe:

```
{
  "executionState": {
    "status": "REMOVED",
    "versionNumber": 3
  },
}
```

Weitere Informationen finden Sie unter [Geräte und Jobs](#) im AWS IoT Developer Guide.

- API-Einheiten finden Sie [UpdateJobExecution](#) in der AWS CLI Befehlsreferenz.

AWS IoT SiteWise Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT SiteWise.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-assets

Das folgende Codebeispiel zeigt die Verwendung `associate-assets`.

AWS CLI

Um eine untergeordnete Anlage einer übergeordneten Anlage zuzuordnen

Im folgenden `associate-assets` Beispiel wird eine Windturbinenanlage einer Windparkanlage zugeordnet, wobei das Windturbinenanlagenmodell als Hierarchie im Windpark-Anlagenmodell existiert.

```
aws iotsitewise associate-assets \
```

```
--asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
--hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \  
--child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Assets zuordnen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [AssociateAssets](#) in der AWS CLI Befehlsreferenz.

batch-associate-project-assets

Das folgende Codebeispiel zeigt die Verwendung `batch-associate-project-assets`.

AWS CLI

Um ein Asset einem Projekt zuzuordnen

Im folgenden `batch-associate-project-assets` Beispiel wird ein Windpark-Objekt einem Projekt zugeordnet.

```
aws iotsitewise batch-associate-project-assets \  
--project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
--asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen von Assets zu Projekten](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [BatchAssociateProjectAssets](#) in der AWS CLI Befehlsreferenz.

batch-disassociate-project-assets

Das folgende Codebeispiel zeigt die Verwendung `batch-disassociate-project-assets`.

AWS CLI

Um ein Asset von einem Projekt zu trennen

Im folgenden `batch-disassociate-project-assets` Beispiel wird die Zuordnung eines Windpark-Objekts zu einem Projekt aufgehoben.


```
aws iotsitewise batch-disassociate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen von Assets zu Projekten](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [BatchDisassociateProjectAssets](#) in der AWS CLI Befehlsreferenz.

batch-put-asset-property-value

Das folgende Codebeispiel zeigt die Verwendung `batch-put-asset-property-value`.

AWS CLI

Um Daten an Objekteigenschaften zu senden

Im folgenden `batch-put-asset-property-value` Beispiel werden Strom- und Temperaturdaten an die durch Eigenschaftsalias identifizierten Eigenschaften der Anlage gesendet.

```
aws iotsitewise batch-put-asset-property-value \  
  --cli-input-json file://batch-put-asset-property-value.json
```

Inhalt von `batch-put-asset-property-value.json`:

```
{  
  "entries": [  
    {  
      "entryId": "1575691200-company-windfarm-3-turbine-7-power",  
      "propertyAlias": "company-windfarm-3-turbine-7-power",  
      "propertyValues": [  
        {  
          "value": {  
            "doubleValue": 4.92  
          },  
          "timestamp": {  
            "timeInSeconds": 1575691200  
          },  
        }  
      ]  
    }  
  ]  
}
```

```

        "quality": "GOOD"
      }
    ]
  },
  {
    "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
    "propertyAlias": "company-windfarm-3-turbine-7-temperature",
    "propertyValues": [
      {
        "value": {
          "integerValue": 38
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        }
      }
    ]
  }
]
}

```

Ausgabe:

```

{
  "errorEntries": []
}

```

Weitere Informationen finden Sie unter [Erfassung von Daten mithilfe des AWS IoT SiteWise API](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchPutAssetPropertyValue](#) in der AWS CLI Befehlsreferenz.

create-access-policy

Das folgende Codebeispiel zeigt die Verwendung `create-access-policy`.

AWS CLI

Beispiel 1: Um einem Benutzer Administratorzugriff auf ein Portal zu gewähren

Im folgenden `create-access-policy` Beispiel wird eine Zugriffsrichtlinie erstellt, die einem Benutzer Administratorzugriff auf ein Webportal für ein Windparkunternehmen gewährt.

```
aws iotsitewise create-access-policy \  
  --cli-input-json file://create-portal-administrator-access-policy.json
```

Inhalt von create-portal-administrator-access-policy.json:

```
{  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
    }  
  },  
  "accessPolicyPermission": "ADMINISTRATOR",  
  "accessPolicyResource": {  
    "portal": {  
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"  
    }  
  }  
}
```

Ausgabe:

```
{  
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/  
a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Hinzufügen oder Entfernen von Portaladministratoren](#) im AWS SiteWise IoT-Benutzerhandbuch.

Beispiel 2: Um einem Benutzer nur Lesezugriff auf ein Projekt zu gewähren

Im folgenden create-access-policy Beispiel wird eine Zugriffsrichtlinie erstellt, die einem Benutzer nur Lesezugriff auf ein Windparkprojekt gewährt.

```
aws iotsitewise create-access-policy \  
  --cli-input-json file://create-project-viewer-access-policy.json
```

Inhalt von create-project-viewer-access-policy.json:

```
{
```

```

    "accessPolicyIdentity": {
      "user": {
        "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
      }
    },
    "accessPolicyPermission": "VIEWER",
    "accessPolicyResource": {
      "project": {
        "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"
      }
    }
  }
}

```

Ausgabe:

```

{
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"
}

```

Weitere Informationen finden Sie unter [Zuweisen von Projekt-Viewern](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [CreateAccessPolicy](#) in der AWS CLI Befehlsreferenz.

create-asset-model

Das folgende Codebeispiel zeigt die Verwendung `create-asset-model`.

AWS CLI

Um ein Asset-Modell zu erstellen

Im folgenden `create-asset-model` Beispiel wird ein Anlagenmodell erstellt, das eine Windturbine mit den folgenden Eigenschaften definiert:

Seriennummer — Die Seriennummer einer Windenergie — Der erzeugte turbineGenerated Energiedatenstrom aus einem Wind turbineTemperature C — Der Temperaturdatenstrom einer Windkraftanlage in CelsiusTemperature F — Die abgebildeten Temperaturdatenpunkte von Celsius bis Fahrenheit

```
aws iotsitewise create-asset-model \  
--cli-input-json file://create-wind-turbine-model.json
```

Inhalt von `create-wind-turbine-model.json`:

```
{  
  "assetModelName": "Wind Turbine Model",  
  "assetModelDescription": "Represents a wind turbine",  
  "assetModelProperties": [  
    {  
      "name": "Serial Number",  
      "dataType": "STRING",  
      "type": {  
        "attribute": {}  
      }  
    },  
    {  
      "name": "Generated Power",  
      "dataType": "DOUBLE",  
      "unit": "kW",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature C",  
      "dataType": "DOUBLE",  
      "unit": "Celsius",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature F",  
      "dataType": "DOUBLE",  
      "unit": "Fahrenheit",  
      "type": {  
        "transform": {  
          "expression": "temp_c * 9 / 5 + 32",  
          "variables": [  
            {  
              "name": "temp_c",  
              "value": {
```

```

        "propertyId": "Temperature C"
      }
    ]
  },
  {
    "name": "Total Generated Power",
    "dataType": "DOUBLE",
    "unit": "kW",
    "type": {
      "metric": {
        "expression": "sum(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "Generated Power"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    }
  }
]
}

```

Ausgabe:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Weitere Informationen finden Sie unter [Definieren von Asset-Modellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateAssetModel](#) in der AWS CLI Befehlsreferenz.

create-asset

Das folgende Codebeispiel zeigt die Verwendung `create-asset`.

AWS CLI

Um ein Asset zu erstellen

Im folgenden `create-asset` Beispiel wird aus einem Anlagenmodell eine Windenergieanlage erstellt.

```
aws iotsitewise create-asset \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --asset-name "Wind Turbine 1"
```

Ausgabe:

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
  "assetStatus": {  
    "state": "CREATING"  
  }  
}
```

Weitere Informationen finden Sie im AWS SiteWise IoT-Benutzerhandbuch unter [Assets erstellen](#).

- API-Einheiten finden Sie [CreateAsset](#) unter AWS CLI Befehlsreferenz.

create-dashboard

Das folgende Codebeispiel zeigt die Verwendung `create-dashboard`.

AWS CLI

Um ein Dashboard zu erstellen

Im folgenden `create-dashboard` Beispiel wird ein Dashboard mit einem Liniendiagramm erstellt, das die gesamte erzeugte Leistung für einen Windpark anzeigt.

```
aws iotsitewise create-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \
  --dashboard-name "Wind Farm" \
  --dashboard-definition file://create-wind-farm-dashboard.json
```

Inhalt von `create-wind-farm-dashboard.json`:

```
{
  "widgets": [
    {
      "type": "monitor-line-chart",
      "title": "Generated Power",
      "x": 0,
      "y": 0,
      "height": 3,
      "width": 3,
      "metrics": [
        {
          "label": "Power",
          "type": "iotsitewise",
          "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
          "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"
        }
      ]
    }
  ]
}
```

Ausgabe:

```
{
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-ffffffEXAMPLE",
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/a1b2c3d4-5678-90ab-cdef-ffffffEXAMPLE"
}
```

Weitere Informationen finden Sie unter [Erstellen von Dashboards \(CLI\)](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDashboard](#) in der AWS CLI Befehlsreferenz.

create-gateway

Das folgende Codebeispiel zeigt die Verwendung `create-gateway`.

AWS CLI

Um ein Gateway zu erstellen

Das folgende `create-gateway` Beispiel erstellt ein Gateway, das auf AWS IoT Greengrass läuft.

```
aws iotsitewise create-gateway \  
  --gateway-name ExampleCorpGateway \  
  --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}
```

Ausgabe:

```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Konfiguration eines Gateways](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateGateway](#) in der AWS CLI Befehlsreferenz.

create-portal

Das folgende Codebeispiel zeigt die Verwendung `create-portal`.

AWS CLI

Um ein Portal zu erstellen

Im folgenden `create-portal` Beispiel wird ein Webportal für ein Windparkunternehmen erstellt. Sie können Portale nur in derselben Region erstellen, in der Sie AWS Single Sign-On aktiviert haben.

```
aws iotsitewise create-portal \  
  --portal-name WindFarmPortal \  
  --portal-description "A portal that contains wind farm projects for Example Corp." \  
  --portal-contact-email support@example.com \  
  --role-arn arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole
```

Ausgabe:

```
{  
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",  
  "portalStatus": {  
    "state": "CREATING"  
  },  
  "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit AWS IoT SiteWise Monitor](#) im AWS SiteWise IoT-Benutzerhandbuch und [Aktivieren AWS SSO](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [CreatePortal](#) in der AWS CLI Befehlsreferenz.

create-project

Das folgende Codebeispiel zeigt die Verwendung `create-project`.

AWS CLI

Um ein Projekt zu erstellen

Im folgenden `create-project` Beispiel wird ein Windparkprojekt erstellt.

```
aws iotsitewise create-project \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \  
  --project-name "Wind Farm 1" \  
  --role-arn arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole
```

```
--project-description "Contains asset visualizations for Wind Farm #1 for Example Corp."
```

Ausgabe:

```
{
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE"
}
```

Weitere Informationen finden Sie unter [Projekte erstellen](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [CreateProject](#) in der AWS CLI Befehlsreferenz.

delete-access-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-access-policy`.

AWS CLI

Um einem Benutzer den Zugriff auf ein Projekt oder Portal zu entziehen

Im folgenden `delete-access-policy` Beispiel wird eine Zugriffsrichtlinie gelöscht, die einem Benutzer Administratorzugriff auf ein Portal gewährt.

```
aws iotsitewise delete-access-policy \  
--access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen oder Entfernen von Portaladministratoren](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAccessPolicy](#) in der AWS CLI Befehlsreferenz.

delete-asset-model

Das folgende Codebeispiel zeigt die Verwendung `delete-asset-model`.

AWS CLI

Um ein Asset-Modell zu löschen

Im folgenden `delete-asset-model` Beispiel wird ein Anlagenmodell einer Windenergieanlage gelöscht.

```
aws iotsitewise delete-asset-model \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{  
  "assetModelStatus": {  
    "state": "DELETING"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen von Asset-Modellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAssetModel](#) unter AWS CLI Befehlsreferenz.

`delete-asset`

Das folgende Codebeispiel zeigt die Verwendung `delete-asset`.

AWS CLI

Um ein Asset zu löschen

Im folgenden `delete-asset` Beispiel wird ein Windturbinen-Asset gelöscht.

```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Ausgabe:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Löschen von Assets](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAsset](#) in der AWS CLI Befehlsreferenz.

delete-dashboard

Das folgende Codebeispiel zeigt die Verwendung `delete-dashboard`.

AWS CLI

Um ein Dashboard zu löschen

Im folgenden `delete-dashboard` Beispiel wird ein Windturbinen-Dashboard gelöscht.

```
aws iotsitewise delete-dashboard \  
  --dashboard-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen von Dashboards](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [DeleteDashboard](#) in der AWS CLI Befehlsreferenz.

delete-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-gateway`.

AWS CLI

Um ein Gateway zu löschen

Im folgenden `delete-gateway` Beispiel wird ein Gateway gelöscht.

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Daten mithilfe eines Gateways](#) aufnehmen im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteGateway](#) in der AWS CLI Befehlsreferenz.

delete-portal

Das folgende Codebeispiel zeigt die Verwendung `delete-portal`.

AWS CLI

Um ein Portal zu löschen

Im folgenden `delete-portal` Beispiel wird ein Webportal für ein Windparkunternehmen gelöscht.

```
aws iotsitewise delete-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE
```

Ausgabe:

```
{  
  "portalStatus": {  
    "state": "DELETING"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen eines Portals](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DeletePortal](#) in der AWS CLI Befehlsreferenz.

delete-project

Das folgende Codebeispiel zeigt die Verwendung `delete-project`.

AWS CLI

Um ein Projekt zu löschen

Im folgenden `delete-project` Beispiel wird ein Windparkprojekt gelöscht.

```
aws iotsitewise delete-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen von Projekten](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [DeleteProject](#) in der AWS CLI Befehlsreferenz.

describe-access-policy

Das folgende Codebeispiel zeigt die Verwendung `describe-access-policy`.

AWS CLI

Um eine Zugriffsrichtlinie zu beschreiben

Das folgende `describe-access-policy` Beispiel beschreibt eine Zugriffsrichtlinie, die einem Benutzer Administratorzugriff auf ein Webportal für ein Windparkunternehmen gewährt.

```
aws iotsitewise describe-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

Ausgabe:

```
{  
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/  
a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
    }  
  },  
  "accessPolicyResource": {  
    "portal": {  
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE"  
    }  
  },  
  "accessPolicyPermission": "ADMINISTRATOR",  
  "accessPolicyCreationDate": "2020-02-20T22:35:15.552880124Z",
```

```
"accessPolicyLastUpdateDate": "2020-02-20T22:35:15.552880124Z"
}
```

Weitere Informationen finden Sie unter [Hinzufügen oder Entfernen von Portaladministratoren](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAccessPolicy](#) in der AWS CLI Befehlsreferenz.

describe-asset-model

Das folgende Codebeispiel zeigt die Verwendung `describe-asset-model`.

AWS CLI

Um ein Asset-Modell zu beschreiben

Das folgende `describe-asset-model` Beispiel beschreibt ein Anlagenmodell für Windparks.

```
aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Ausgabe:

```
{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind turbines",
  "assetModelProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
              "name": "power",
              "value": {
```



```

        "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
        "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
    }
  ],
  "window": {
    "tumbling": {
      "interval": "1h"
    }
  }
},
{
  "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
  "name": "Region",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": " "
    }
  }
},
"assetModelHierarchies": [
  {
    "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
    "name": "Wind Turbines",
    "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
],
"assetModelCreationDate": 1575671284.0,
"assetModelLastUpdateDate": 1575671988.0,
"assetModelStatus": {
  "state": "ACTIVE"
}
}

```

Weitere Informationen finden Sie unter [Beschreibung eines bestimmten Asset-Modells](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAssetModel](#) in der AWS CLI Befehlsreferenz.

describe-asset-property

Das folgende Codebeispiel zeigt die Verwendung `describe-asset-property`.

AWS CLI

Um eine Immobilie zu beschreiben

Das folgende `describe-asset-property` Beispiel beschreibt die gesamte Stromerzeugungskapazität einer Windparkanlage.

```
aws iotsitewise describe-asset-property \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE
```

Ausgabe:

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",  
  "assetName": "Wind Farm 1",  
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
  "assetProperty": {  
    "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",  
    "name": "Total Generated Power",  
    "notification": {  
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",  
      "state": "DISABLED"  
    },  
    "dataType": "DOUBLE",  
    "unit": "kW",  
    "type": {  
      "metric": {  
        "expression": "sum(power)",  
        "variables": [  
          {  
            "name": "power",  
            "value": {  
              "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE",  
              "hierarchyId": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE"  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```



```

    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW"
    }
  ],
  "assetHierarchies": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
      "name": "Wind Turbines"
    }
  ],
  "assetCreationDate": 1575672453.0,
  "assetLastUpdateDate": 1575672453.0,
  "assetStatus": {
    "state": "ACTIVE"
  }
}

```

Weitere Informationen finden Sie unter [Beschreibung eines bestimmten Assets](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAsset](#) in der AWS CLI Befehlsreferenz.

describe-dashboard

Das folgende Codebeispiel zeigt die Verwendung `describe-dashboard`.

AWS CLI

Um ein Dashboard zu beschreiben

Das folgende `describe-dashboard` Beispiel beschreibt das angegebene Windpark-Dashboard.

```

aws iotsitewise describe-dashboard \
  --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE

```

Ausgabe:

```
{
```

```

    "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardName": "Wind Farm",
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",
    "dashboardDefinition": "{\"widgets\": [{\"type\": \"monitor-line-chart\", \"title
\": \"Generated Power\", \"x\": 0, \"y\": 0, \"height\": 3, \"width\": 3, \"metrics\":
[ {\"label\": \"Power\", \"type\": \"iotsitewise\", \"assetId\": \"a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE\", \"propertyId\": \"a1b2c3d4-5678-90ab-cdef-99999EXAMPLE\"} ] } ] }",
    "dashboardCreationDate": "2020-05-01T20:32:12.228476348Z",
    "dashboardLastUpdateDate": "2020-05-01T20:32:12.228476348Z"
}

```

Weitere Informationen finden Sie unter [Anzeigen von Dashboards](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [DescribeDashboard](#) in der AWS CLI Befehlsreferenz.

describe-gateway-capability-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-gateway-capability-configuration`.

AWS CLI

Um eine Gateway-Fähigkeit zu beschreiben

Das folgende `describe-gateway-capability-configuration` Beispiel beschreibt eine OPC-UA-Quellfunktion.

```

aws iotsitewise describe-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1"

```

Ausgabe:

```

{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilityConfiguration": "{\"sources\": [ {\"name\": \"Wind Farm #1\",
  \"endpoint\": {\"certificateTrust\": {\"type\": \"TrustAny\"}, \"endpointUri
\": \"opc.tcp://203.0.113.0:49320\", \"securityPolicy\": \"BASIC256\",

```

```

\"messageSecurityMode\": \"SIGN_AND_ENCRYPT\", \"identityProvider\":
{ \"type\": \"Username\", \"usernameSecretArn\": \"arn:aws:secretsmanager:us-
east-1:123456789012:secret:greenrass-factory1-auth-3QNDmM\" }, \"nodeFilterRules\":
[] }, \"measurementDataStreamPrefix\": \"\" } } ],
  \"capabilitySyncStatus\": \"IN_SYNC\"
}

```

Weitere Informationen finden Sie unter [Konfiguration von Datenquellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeGatewayCapabilityConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-gateway

Das folgende Codebeispiel zeigt die Verwendung `describe-gateway`.

AWS CLI

Um ein Gateway zu beschreiben

Das folgende `describe-gateway` Beispiel beschreibt ein Gateway.

```

aws iotsitewise describe-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE

```

Ausgabe:

```

{
  \"gatewayId\": \"a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE\",
  \"gatewayName\": \"ExampleCorpGateway\",
  \"gatewayArn\": \"arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE\",
  \"gatewayPlatform\": {
    \"greengrass\": {
      \"groupArn\": \"arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE\"
    }
  },
  \"gatewayCapabilitySummaries\": [
    {
      \"capabilityNamespace\": \"iotsitewise:opcuacollector:1\",

```

```
        "capabilitySyncStatus": "IN_SYNC"
      }
    ],
    "creationDate": 1588369971.457,
    "lastUpdateDate": 1588369971.457
  }
```

Weitere Informationen finden Sie unter [Daten mithilfe eines Gateways](#) aufnehmen im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeGateway](#) in der AWS CLI Befehlsreferenz.

describe-logging-options

Das folgende Codebeispiel zeigt die Verwendung `describe-logging-options`.

AWS CLI

Um die aktuellen AWS SiteWise IoT-Protokollierungsoptionen abzurufen

Im folgenden `describe-logging-options` Beispiel werden die aktuellen AWS SiteWise IoT-Protokollierungsoptionen für Ihr AWS Konto in der aktuellen Region abgerufen.

```
aws iotsitewise describe-logging-options
```

Ausgabe:

```
{
  "loggingOptions": {
    "level": "INFO"
  }
}
```

Weitere Informationen finden Sie unter [Monitoring AWS IoT SiteWise with Amazon CloudWatch Logs](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeLoggingOptions](#) in der AWS CLI Befehlsreferenz.

describe-portal

Das folgende Codebeispiel zeigt die Verwendung `describe-portal`.

AWS CLI

Um ein Portal zu beschreiben

Das folgende `describe-portal` Beispiel beschreibt ein Webportal für ein Windparkunternehmen.

```
aws iotsitewise describe-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Ausgabe:

```
{  
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/  
a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "portalName": "WindFarmPortal",  
  "portalDescription": "A portal that contains wind farm projects for Example  
Corp.",  
  "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",  
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-  
aaaaaEXAMPLE.app.iotsitewise.aws",  
  "portalContactEmail": "support@example.com",  
  "portalStatus": {  
    "state": "ACTIVE"  
  },  
  "portalCreationDate": "2020-02-04T23:01:52.90248068Z",  
  "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",  
  "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"  
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Portale](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribePortal](#) in der AWS CLI Befehlsreferenz.

describe-project

Das folgende Codebeispiel zeigt die Verwendung `describe-project`.

AWS CLI

Um ein Projekt zu beschreiben

Das folgende `describe-project` Beispiel beschreibt ein Windparkprojekt.

```
aws iotsitewise describe-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE
```

Ausgabe:

```
{  
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/  
a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
  "projectName": "Wind Farm 1",  
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "projectDescription": "Contains asset visualizations for Wind Farm #1 for  
Example Corp.",  
  "projectCreationDate": "2020-02-20T21:58:43.362246001Z",  
  "projectLastUpdateDate": "2020-02-20T21:58:43.362246095Z"  
}
```

Weitere Informationen finden Sie unter [Projektdetails anzeigen](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [DescribeProject](#) in der AWS CLI Befehlsreferenz.

disassociate-assets

Das folgende Codebeispiel zeigt die Verwendung `disassociate-assets`.

AWS CLI

Um eine untergeordnete Anlage von einer übergeordneten Anlage zu trennen

Im folgenden `disassociate-assets` Beispiel wird die Zuordnung einer Windturbinenanlage von einer Windparkanlage getrennt.

```
aws iotsitewise disassociate-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \  
  --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Assets zuordnen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateAssets](#) in der AWS CLI Befehlsreferenz.

get-asset-property-aggregates

Das folgende Codebeispiel zeigt die Verwendung `get-asset-property-aggregates`.

AWS CLI

Um die aggregierten Durchschnitts- und Zählwerte einer Anlageeigenschaft abzurufen

Im folgenden `get-asset-property-aggregates` Beispiel werden die durchschnittliche Gesamtleistung und die Anzahl der Gesamtleistungsdatenpunkte einer Windenergieanlage für einen Zeitraum von 1 Stunde abgerufen.

```
aws iotsitewise get-asset-property-aggregates \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE \  
  --start-date 1580849400 \  
  --end-date 1580853000 \  
  --aggregate-types AVERAGE COUNT \  
  --resolution 1h
```

Ausgabe:

```
{  
  "aggregatedValues": [  
    {  
      "timestamp": 1580850000.0,  
      "quality": "GOOD",  
      "value": {  
        "average": 8723.46538886233,  
        "count": 12.0  
      }  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Abfragen von Aggregaten für Asset-Eigenschaften](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetAssetPropertyAggregates AWS CLI Befehlsreferenz](#).

get-asset-property-value-history

Das folgende Codebeispiel zeigt die Verwendung `get-asset-property-value-history`.

AWS CLI

Um die historischen Werte einer Anlageeigenschaft abzurufen

Im folgenden `get-asset-property-value-history` Beispiel werden die Gesamtleistungswerte einer Windenergieanlage für einen Zeitraum von 20 Minuten abgerufen.

```
aws iotsitewise get-asset-property-value-history \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE \  
  --start-date 1580851800 \  
  --end-date 1580853000
```

Ausgabe:

```
{  
  "assetPropertyValueHistory": [  
    {  
      "value": {  
        "doubleValue": 7217.787046814844  
      },  
      "timestamp": {  
        "timeInSeconds": 1580852100,  
        "offsetInNanos": 0  
      },  
      "quality": "GOOD"  
    },  
    {  
      "value": {  
        "doubleValue": 6941.242811875451  
      },  
      "timestamp": {  
        "timeInSeconds": 1580852400,  
        "offsetInNanos": 0  
      },  
      "quality": "GOOD"  
    }  
  ]  
}
```

```
    },
    {
      "value": {
        "doubleValue": 6976.797662266717
      },
      "timestamp": {
        "timeInSeconds": 1580852700,
        "offsetInNanos": 0
      },
      "quality": "GOOD"
    },
    {
      "value": {
        "doubleValue": 6890.8677520453875
      },
      "timestamp": {
        "timeInSeconds": 1580853000,
        "offsetInNanos": 0
      },
      "quality": "GOOD"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Abfragen historischer Immobilienwerte](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetAssetPropertyValueHistory AWS CLI Befehlsreferenz](#).

get-asset-property-value

Das folgende Codebeispiel zeigt die Verwendung von `get-asset-property-value`.

AWS CLI

Um den aktuellen Wert einer Anlageeigenschaft abzurufen

Im folgenden `get-asset-property-value` Beispiel wird die aktuelle Gesamtleistung einer Windenergieanlage abgerufen.

```
aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE
```

Ausgabe:

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

Weitere Informationen finden Sie unter [Abfragen aktueller Objekteigenschaftswerte](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetAssetPropertyValue AWS CLI Befehlsreferenz](#).

list-access-policies

Das folgende Codebeispiel zeigt die Verwendung `list-access-policies`.

AWS CLI

Um alle Zugriffsrichtlinien aufzulisten

Das folgende `list-access-policies` Beispiel listet alle Zugriffsrichtlinien für einen Benutzer auf, der Portaladministrator ist.

```
aws iotsitewise list-access-policies \
  --identity-type USER \
  --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE
```

Ausgabe:

```
{
  "accessPolicySummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",

```

```

    "identity": {
      "user": {
        "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE"
      }
    },
    "resource": {
      "portal": {
        "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
      }
    },
    "permission": "ADMINISTRATOR"
  }
]
}

```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Portale](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAccessPolicies](#) in der AWS CLI Befehlsreferenz.

list-asset-models

Das folgende Codebeispiel zeigt die Verwendung `list-asset-models`.

AWS CLI

Um alle Asset-Modelle aufzulisten

Das folgende `list-asset-models` Beispiel listet alle Vermögensmodelle auf, die in Ihrem AWS Konto in der aktuellen Region definiert sind.

```
aws iotsitewise list-asset-models
```

Ausgabe:

```

{
  "assetModelSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "Wind Farm Model",
    }
  ]
}

```

```

        "description": "Represents a wind farm that comprises many wind
turbines",
        "creationDate": 1575671284.0,
        "lastUpdateDate": 1575671988.0,
        "status": {
            "state": "ACTIVE"
        }
    },
    {
        "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "name": "Wind Turbine Model",
        "description": "Represents a wind turbine manufactured by Example Corp",
        "creationDate": 1575671207.0,
        "lastUpdateDate": 1575686273.0,
        "status": {
            "state": "ACTIVE"
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Auflisten aller Asset-Modelle](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAssetModels](#) in der AWS CLI Befehlsreferenz.

list-assets

Das folgende Codebeispiel zeigt die Verwendung `list-assets`.

AWS CLI

Beispiel 1: Um alle Vermögenswerte der obersten Ebene aufzulisten

Das folgende `list-assets` Beispiel listet alle Vermögenswerte auf, die sich in der Asset-Hierarchiestruktur auf oberster Ebene befinden und in Ihrem AWS Konto in der aktuellen Region definiert sind.

```

aws iotsitewise list-assets \
  --filter TOP_LEVEL

```

Ausgabe:

```
{
  "assetSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "name": "Wind Farm 1",
      "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "creationDate": 1575672453.0,
      "lastUpdateDate": 1575672453.0,
      "status": {
        "state": "ACTIVE"
      },
      "hierarchies": [
        {
          "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
          "name": "Wind Turbines"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Auflisten von Assets](#) im AWS SiteWise IoT-Benutzerhandbuch.

Beispiel 2: Um alle Anlagen aufzulisten, die auf einem Asset-Modell basieren

Im folgenden `list-assets` Beispiel werden alle Vermögenswerte aufgeführt, die auf einem Vermögensmodell basieren und in Ihrem AWS Konto in der aktuellen Region definiert sind.

```
aws iotsitewise list-assets \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{
  "assetSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
```



```

        "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
        "name": "Wind Turbine 1",
        "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "creationDate": 1575671550.0,
        "lastUpdateDate": 1575686308.0,
        "status": {
            "state": "ACTIVE"
        },
        "hierarchies": []
    }
]
}

```

Weitere Informationen finden Sie unter [Auflisten von Assets](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAssets](#) in der AWS CLI Befehlsreferenz.

list-associated-assets

Das folgende Codebeispiel zeigt die Verwendung `list-associated-assets`.

AWS CLI

Um alle mit einem Asset verknüpften Assets in einer bestimmten Hierarchie aufzulisten

Das folgende `list-associated-assets` Beispiel listet alle Windturbinenanlagen auf, die dem angegebenen Windpark-Asset zugeordnet sind.

```

aws iotsitewise list-associated-assets \
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE

```

Ausgabe:

```

{
  "assetSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",

```

```

        "name": "Wind Turbine 1",
        "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "creationDate": 1575671550.0,
        "lastUpdateDate": 1575686308.0,
        "status": {
            "state": "ACTIVE"
        },
        "hierarchies": []
    }
]
}

```

Weitere Informationen finden Sie im AWS SiteWise IoT-Benutzerhandbuch unter [Auflisten von Assets, die einem bestimmten Asset zugeordnet](#) sind.

- API-Einheiten finden Sie [ListAssociatedAssets](#) in der AWS CLI Befehlsreferenz.

list-dashboards

Das folgende Codebeispiel zeigt die Verwendung `list-dashboards`.

AWS CLI

Um alle Dashboards in einem Projekt aufzulisten

Das folgende `list-dashboards` Beispiel listet alle Dashboards auf, die in einem Projekt definiert sind.

```

aws iotsitewise list-dashboards \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE

```

Ausgabe:

```

{
  "dashboardSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE",
      "name": "Wind Farm",
      "creationDate": "2020-05-01T20:32:12.228476348Z",
      "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"
    }
  ]
}

```

```
}
```

Weitere Informationen finden Sie unter [Anzeigen von Dashboards](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [ListDashboards](#) in der AWS CLI Befehlsreferenz.

list-gateways

Das folgende Codebeispiel zeigt die Verwendung `list-gateways`.

AWS CLI

Um alle Gateways aufzulisten

Das folgende `list-gateways` Beispiel listet alle Gateways auf, die in Ihrem AWS Konto in der aktuellen Region definiert sind.

```
aws iotsitewise list-gateways
```

Ausgabe:

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

Weitere Informationen finden Sie unter [Daten mithilfe eines Gateways](#) aufnehmen im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [ListGateways](#) in der AWS CLI Befehlsreferenz.

list-portals

Das folgende Codebeispiel zeigt die Verwendung `list-portals`.

AWS CLI

Um alle Portale aufzulisten

Das folgende `list-portals` Beispiel listet alle Portale auf, die in Ihrem AWS Konto in der aktuellen Region definiert sind.

```
aws iotsitewise list-portals
```

Ausgabe:

```
{
  "portalSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "name": "WindFarmPortal",
      "description": "A portal that contains wind farm projects for Example Corp.",
      "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
      "creationDate": "2020-02-04T23:01:52.90248068Z",
      "lastUpdateDate": "2020-02-04T23:01:52.90248078Z",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Portale](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPortals](#) in der AWS CLI Befehlsreferenz.

list-project-assets

Das folgende Codebeispiel zeigt die Verwendung `list-project-assets`.

AWS CLI

Um alle mit einem Projekt verknüpften Assets aufzulisten

Im folgenden `list-project-assets` Beispiel werden alle Anlagen aufgeführt, die einem Windparkprojekt zugeordnet sind.

```
aws iotsitewise list-projects \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Ausgabe:

```
{  
  "assetIds": [  
    "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"  
  ]  
}
```

Weitere Informationen finden Sie unter [Hinzufügen von Assets zu Projekten](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [ListProjectAssets](#) in der AWS CLI Befehlsreferenz.

list-projects

Das folgende Codebeispiel zeigt die Verwendung `list-projects`.

AWS CLI

Um alle Projekte in einem Portal aufzulisten

Das folgende `list-projects` Beispiel listet alle Projekte auf, die in einem Portal definiert sind.

```
aws iotsitewise list-projects \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE
```

Ausgabe:

```
{  
  "projectSummaries": [  
    {  
      "name": "Project 1"  
    },  
    {  
      "name": "Project 2"  
    }  
  ]  
}
```

```
{
  "id": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",
  "name": "Wind Farm 1",
  "description": "Contains asset visualizations for Wind Farm #1 for
Example Corp.",
  "creationDate": "2020-02-20T21:58:43.362246001Z",
  "lastUpdateDate": "2020-02-20T21:58:43.362246095Z"
}
]
```

Weitere Informationen finden Sie unter [Projektdetails anzeigen](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [ListProjects](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um alle Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für eine Windturbinenanlage auf.

```
aws iotsitewise list-tags-for-resource \
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Ausgabe:

```
{
  "tags": {
    "Owner": "richard-roe"
  }
}
```

Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-logging-options

Das folgende Codebeispiel zeigt die Verwendung `put-logging-options`.

AWS CLI

Um die Protokollierungsebene anzugeben

Das folgende `put-logging-options` Beispiel aktiviert die INFO Level-Protokollierung in AWS IoT SiteWise. Zu den anderen Ebenen gehören DEBUG und OFF.

```
aws iotsitewise put-logging-options \  
  --logging-options Level=INFO
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Monitoring AWS IoT SiteWise with Amazon CloudWatch Logs](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [PutLoggingOptions](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einer Ressource einen Tag hinzu

Im folgenden `tag-resource` Beispiel wird einer Windturbinenanlage ein Besitzer-Tag hinzugefügt. Auf diese Weise können Sie den Zugriff auf das Asset anhand dessen steuern, wem es gehört.

```
aws iotsitewise tag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --tags Owner=richard-roe
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird ein Besitzer-Tag aus einer Windturbinenanlage entfernt.

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --tag-keys Owner
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Taggen Ihrer Ressourcen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-access-policy

Das folgende Codebeispiel zeigt die Verwendung `update-access-policy`.

AWS CLI

Um einem Projektbetrachter das Eigentum an einem Projekt zu gewähren

Im folgenden `update-access-policy` Beispiel wird eine Zugriffsrichtlinie aktualisiert, die einem Projektbetrachter die Inhaberschaft an einem Projekt gewährt.

```
aws iotsitewise update-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \  
  --policy Owner
```



```
--cli-input-json file://update-project-viewer-access-policy.json
```

Inhalt von `update-project-viewer-access-policy.json`:

```
{
  "accessPolicyIdentity": {
    "user": {
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
    }
  },
  "accessPolicyPermission": "ADMINISTRATOR",
  "accessPolicyResource": {
    "project": {
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE"
    }
  }
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Zuweisen von Projekteigentümern](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [UpdateAccessPolicy](#) in der AWS CLI Befehlsreferenz.

update-asset-model

Das folgende Codebeispiel zeigt die Verwendung `update-asset-model`.

AWS CLI

Um ein Asset-Modell zu aktualisieren

Im folgenden `update-asset-model` Beispiel wird die Beschreibung eines Windpark-Assetmodells aktualisiert. In diesem Beispiel werden die vorhandenen IDs und die Definitionen des Modells berücksichtigt, da das bestehende Modell durch das neue Modell `update-asset-model` überschrieben wird.

```
aws iotsitewise update-asset-model \  
--cli-input-json file://update-wind-farm-model.json
```

Inhalt von update-wind-farm-model.json:

```

{
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind
turbines",
  "assetModelProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING",
      "type": {
        "attribute": {}
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
              "name": "power",
              "value": {
                "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE",
                "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE"
              }
            }
          ]
        },
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    }
  ],
  "assetModelHierarchies": [
    {

```

```

        "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
        "name": "Wind Turbines",
        "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
]
}

```

Ausgabe:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren von Asset-Modellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAssetModel](#) in der AWS CLI Befehlsreferenz.

update-asset-property

Das folgende Codebeispiel zeigt die Verwendung `update-asset-property`.

AWS CLI

Beispiel 1: Um den Alias einer Asset-Eigenschaft zu aktualisieren

Im folgenden `update-asset-property` Beispiel wird der Alias für die Energieeigenschaft einer Windenergieanlage aktualisiert.

```

aws iotsitewise update-asset-property \
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-55555EXAMPLE \
  --property-alias "/examplecorp/windfarm/1/turbine/1/power" \
  --property-notification-state DISABLED

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS SiteWise IoT-Benutzerhandbuch unter [Zuordnung von industriellen Datenströmen zu Anlageneigenschaften](#).

Beispiel 2: Um Benachrichtigungen über Anlageneigenschaften zu aktivieren

Im folgenden `update-asset-property` Beispiel werden Benachrichtigungen zur Aktualisierung von Anlageneigenschaften für die Energieeigenschaften einer Windenergieanlage aktiviert. Aktualisierungen von Eigenschaftswerten werden unter dem MQTT Thema veröffentlicht `$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>`, wobei jede ID durch die Eigenschafts-, Objekt- und Modell-ID der Anlageneigenschaft ersetzt wird.

```
aws iotsitewise update-asset-property \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE \  
  --property-notification-state ENABLED \  
  --property-alias "/examplecorp/windfarm/1/turbine/1/power"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Interaktion mit anderen Diensten](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAssetProperty](#) in der AWS CLI Befehlsreferenz.

update-asset

Das folgende Codebeispiel zeigt die Verwendung `update-asset`.

AWS CLI

Um den Namen eines Assets zu aktualisieren

Im folgenden `update-asset` Beispiel wird der Name einer Windenergieanlage aktualisiert.

```
aws iotsitewise update-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --asset-name "Wind Turbine 2"
```

Ausgabe:

```
{
  "assetStatus": {
    "state": "UPDATING"
  }
}
```

Weitere Informationen finden Sie unter [Aktualisieren von Ressourcen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAsset](#) in der AWS CLI Befehlsreferenz.

update-dashboard

Das folgende Codebeispiel zeigt die Verwendung `update-dashboard`.

AWS CLI

Um ein Dashboard zu aktualisieren

Im folgenden `update-dashboard` Beispiel wird der Titel des Liniendiagramms eines Dashboards geändert, in dem die gesamte erzeugte Leistung für einen Windpark angezeigt wird.

```
aws iotsitewise update-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE \
  --dashboard-name "Wind Farm" \
  --dashboard-definition file://update-wind-farm-dashboard.json
```

Inhalt von `update-wind-farm-dashboard.json`:

```
{
  "widgets": [
    {
      "type": "monitor-line-chart",
      "title": "Total Generated Power",
      "x": 0,
      "y": 0,
      "height": 3,
      "width": 3,
      "metrics": [
        {
          "label": "Power",
          "type": "iotsitewise",
```

```

        "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"
      }
    ]
  }
}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen von Dashboards \(CLI\)](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDashboard](#) in der AWS CLI Befehlsreferenz.

update-gateway-capability-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-gateway-capability-configuration`.

AWS CLI

Um eine Gateway-Funktion zu aktualisieren

Im folgenden `update-gateway-capability-configuration` Beispiel wird eine OPC-UA-Quelle mit den folgenden Eigenschaften konfiguriert:

Vertraut jedem Zertifikat. Verwendet den Basic256-Algorithmus, um Nachrichten zu sichern. Verwendet den SignAndEncrypt Modus, um Verbindungen zu sichern. Verwendet Authentifizierungsdaten, die in einem Secrets Manager-Geheimnis gespeichert sind. AWS

```

aws iotsitewise update-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1" \
  --capability-configuration file://opc-ua-capability-configuration.json

```

Inhalt von `opc-ua-capability-configuration.json`:

```

{
  "sources": [
    {

```

```

    "name": "Wind Farm #1",
    "endpoint": {
      "certificateTrust": {
        "type": "TrustAny"
      },
      "endpointUri": "opc.tcp://203.0.113.0:49320",
      "securityPolicy": "BASIC256",
      "messageSecurityMode": "SIGN_AND_ENCRYPT",
      "identityProvider": {
        "type": "Username",
        "usernameSecretArn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:green-grass-windfarm1-auth-1ABCDE"
      },
      "nodeFilterRules": []
    },
    "measurementDataStreamPrefix": ""
  }
]
}

```

Ausgabe:

```

{
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilitySyncStatus": "OUT_OF_SYNC"
}

```

Weitere Informationen finden Sie unter [Konfiguration von Datenquellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateGatewayCapabilityConfiguration](#) in der AWS CLI Befehlsreferenz.

update-gateway

Das folgende Codebeispiel zeigt die Verwendung `update-gateway`.

AWS CLI

Um den Namen eines Gateways zu aktualisieren

Im folgenden `update-gateway` Beispiel wird der Name eines Gateways aktualisiert.

```
aws iotsitewise update-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --gateway-name ExampleCorpGateway1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Daten mithilfe eines Gateways](#) aufnehmen im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateGateway](#) in der AWS CLI Befehlsreferenz.

update-portal

Das folgende Codebeispiel zeigt die Verwendung `update-portal`.

AWS CLI

Um die Details eines Portals zu aktualisieren

Im folgenden `update-portal` Beispiel wird ein Webportal für ein Windparkunternehmen aktualisiert.

```
aws iotsitewise update-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \  
  --portal-name WindFarmPortal \  
  --portal-description "A portal that contains wind farm projects for Example Corp." \  
  --portal-contact-email support@example.com \  
  --role-arn arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole
```

Ausgabe:

```
{  
  "portalStatus": {  
    "state": "UPDATING"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer Portale](#) im AWS SiteWise IoT-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdatePortal](#) in der AWS CLI Befehlsreferenz.

update-project

Das folgende Codebeispiel zeigt die Verwendung `update-project`.

AWS CLI

Um die Details eines Projekts zu aktualisieren

Im folgenden `update-project` Beispiel wird ein Windparkprojekt aktualisiert.

```
aws iotsitewise update-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --project-name "Wind Farm 1" \  
  --project-description "Contains asset visualizations for Wind Farm #1 for Example Corp."
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Projektdetails ändern](#) im AWS IoT SiteWise Monitor-Anwendungshandbuch.

- API-Einheiten finden Sie [UpdateProject](#) in der AWS CLI Befehlsreferenz.

AWS IoT Things Graph Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Things Graph.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-entity-to-thing

Das folgende Codebeispiel zeigt die Verwendung `associate-entity-to-thing`.

AWS CLI

Um eine Sache mit einem Gerät zu verknüpfen

Im folgenden `associate-entity-to-thing` Beispiel wird ein Ding einem Gerät zugeordnet. Das Beispiel verwendet ein Bewegungssensorgerät, das sich im öffentlichen Namespace befindet.

```
aws iotthingsgraph associate-entity-to-thing \  
  --thing-name "MotionSensorName" \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Modelle erstellen und hochladen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [AssociateEntityToThing](#) in der AWS CLI Befehlsreferenz.

create-flow-template

Das folgende Codebeispiel zeigt die Verwendung `create-flow-template`.

AWS CLI

Um einen Flow zu erstellen

Im folgenden `create-flow-template` Beispiel wird ein Flow (Workflow) erstellt. Der Wert von `MyFlowDefinition` ist das GraphQL, das den Fluss modelliert.

```
aws iotthingsgraph create-flow-template \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

Ausgabe:

```
{  
  "summary": {  
    "createdAt": 1559248067.545,  
  }  
}
```

```

    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
    "revisionNumber": 1
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateFlowTemplate](#) in der AWS CLI Befehlsreferenz.

create-system-instance

Das folgende Codebeispiel zeigt die Verwendung `create-system-instance`.

AWS CLI

Um eine Systeminstanz zu erstellen

Das folgende `create-system-instance` Beispiel erstellt eine Systeminstanz. Der Wert von `MySystemInstanceDefinition` ist GraphQL, das die Systeminstanz modelliert.

```

aws iotthingsgraph create-system-instance -\
  -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
  --target CLOUD \
  --flow-actions-role-arn myRoleARN

```

Ausgabe:

```

{
  "summary": {
    "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218",
    "status": "NOT_DEPLOYED",
    "target": "CLOUD",
    "createdAt": 1559249315.208,
    "updatedAt": 1559249315.208
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSystemInstance](#) in der AWS CLI Befehlsreferenz.

create-system-template

Das folgende Codebeispiel zeigt die Verwendung `create-system-template`.

AWS CLI

Um ein System zu erstellen

Das folgende `create-system-template` Beispiel erstellt ein System. Der Wert von `MySystemDefinition` ist das GraphQL, das das System modelliert.

```
aws iotthingsgraph create-system-template \  
  --definition language=GRAPHQL,text="MySystemDefinition"
```

Ausgabe:

```
{  
  "summary": {  
    "createdAt": 1559249776.254,  
    "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",  
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/  
MySystem",  
    "revisionNumber": 1  
  }  
}
```

Weitere Informationen finden Sie unter [Creating Systems](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSystemTemplate](#) in der AWS CLI Befehlsreferenz.

delete-flow-template

Das folgende Codebeispiel zeigt die Verwendung `delete-flow-template`.

AWS CLI

Um einen Flow zu löschen

Im folgenden `delete-flow-template` Beispiel wird ein Flow (Workflow) gelöscht.

```
aws iotthingsgraph delete-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteFlowTemplate](#) in der AWS CLI Befehlsreferenz.

delete-namespace

Das folgende Codebeispiel zeigt die Verwendung `delete-namespace`.

AWS CLI

Um einen Namespace zu löschen

Im folgenden `delete-namespace` Beispiel wird ein Namespace gelöscht.

```
aws iotthingsgraph delete-namespace
```

Ausgabe:

```
{  
  "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",  
  "namespaceName": "us-west-2/123456789012/default"  
}
```

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteNamespace](#) in der AWS CLI Befehlsreferenz.

delete-system-instance

Das folgende Codebeispiel zeigt die Verwendung `delete-system-instance`.

AWS CLI

Um eine Systeminstanz zu löschen

Das folgende `delete-system-instance` Beispiel löscht eine Systeminstanz.

```
aws iotthingsgraph delete-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSystemInstance](#) in der AWS CLI Befehlsreferenz.

`delete-system-template`

Das folgende Codebeispiel zeigt die Verwendung `delete-system-template`.

AWS CLI

Um ein System zu löschen

Das folgende `delete-system-template` Beispiel löscht ein System.

```
aws iotthingsgraph delete-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSystemTemplate](#) in der AWS CLI Befehlsreferenz.

`deploy-system-instance`

Das folgende Codebeispiel zeigt die Verwendung `deploy-system-instance`.

AWS CLI

Um eine Systeminstanz bereitzustellen

Im folgenden `delete-system-template` Beispiel wird eine Systeminstanz bereitgestellt.

```
aws iotthingsgraph deploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Ausgabe:

```
{  
  "summary": {  
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment:Room218",  
    "createdAt": 1559249776.254,  
    "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",  
    "status": "DEPLOYED_IN_TARGET",  
    "target": "CLOUD",  
    "updatedAt": 1559249776.254  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DeploySystemInstance](#) in der AWS CLI Befehlsreferenz.

deprecate-flow-template

Das folgende Codebeispiel zeigt die Verwendung `deprecate-flow-template`.

AWS CLI

Um einen Flow als veraltet zu kennzeichnen

Im folgenden `deprecate-flow-template` Beispiel wird ein Flow (Workflow) als veraltet eingestuft.

```
aws iotthingsgraph deprecate-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DeprecateFlowTemplate](#) in der AWS CLI Befehlsreferenz.

deprecate-system-template

Das folgende Codebeispiel zeigt die Verwendung `deprecate-system-template`.

AWS CLI

Um ein System als veraltet zu kennzeichnen

Im folgenden `deprecate-system-template` Beispiel wird ein System als veraltet eingestuft.

```
aws iotthingsgraph deprecate-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Benutzerhandbuch.

- API-Einheiten finden Sie [DeprecateSystemTemplate](#) in der AWS CLI Befehlsreferenz.

describe-namespace

Das folgende Codebeispiel zeigt die Verwendung `describe-namespace`.

AWS CLI

Um eine Beschreibung Ihres Namespaces zu erhalten

Im folgenden `describe-namespace` Beispiel wird eine Beschreibung Ihres Namespaces abgerufen.

```
aws iotthingsgraph describe-namespace
```

Ausgabe:

```
{  
  "namespaceName": "us-west-2/123456789012/default",  
  "trackingNamespaceName": "aws",  
  "trackingNamespaceVersion": 1,  
  "namespaceVersion": 5  
}
```


Weitere Informationen finden Sie unter [Namespaces](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeNamespace](#) in AWS CLI der Befehlsreferenz.

dissociate-entity-from-thing

Das folgende Codebeispiel zeigt die Verwendung `dissociate-entity-from-thing`.

AWS CLI

Um eine Sache von einem Gerät zu trennen

Im folgenden `dissociate-entity-from-thing` Beispiel wird die Verbindung zwischen einem Objekt und einem Gerät getrennt.

```
aws iotthingsgraph dissociate-entity-from-thing \  
  --thing-name "MotionSensorName" \  
  --entity-type "DEVICE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Modelle erstellen und hochladen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [DissociateEntityFromThing](#) in der AWS CLI Befehlsreferenz.

get-entities

Das folgende Codebeispiel zeigt die Verwendung `get-entities`.

AWS CLI

Um Definitionen für Entitäten zu erhalten

Im folgenden `get-entities` Beispiel wird eine Definition für ein Gerätemodell abgerufen.

```
aws iotthingsgraph get-entities \  
  --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

Ausgabe:

```
{
```

```

    "descriptions": [
      {
        "id": "urn:tdm:aws/examples:DeviceModel:MotionSensor",
        "type": "DEVICE_MODEL",
        "createdAt": 1559256190.599,
        "definition": {
          "language": "GRAPHQL",
          "text": "##\n# Specification of motion sensor devices interface.\n##
\n#type MotionSensor @deviceModel(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\n\", \n          capability: \"urn:tdm:aws/examples:capability:MotionSensorCapability\")
          {ignore:void}"
        }
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Modelle erstellen und hochladen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetEntities](#) in der AWS CLI Befehlsreferenz.

get-flow-template-revisions

Das folgende Codebeispiel zeigt die Verwendung von `get-flow-template-revisions`.

AWS CLI

Um Revisionsinformationen zu einem Flow abzurufen

Im folgenden `get-flow-template-revisions` Beispiel werden Revisionsinformationen zu einem Flow (Workflow) abgerufen.

```

aws iotthingsgraph get-flow-template-revisions \
  --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow

```

Ausgabe:

```

{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
      "revisionNumber": 1,
    }
  ]
}

```

```

    "createdAt": 1559247540.292
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetFlowTemplateRevisions](#) in der AWS CLI Befehlsreferenz.

get-flow-template

Das folgende Codebeispiel zeigt die Verwendung von `get-flow-template`.

AWS CLI

Um eine Flow-Definition zu erhalten

Im folgenden `get-flow-template` Beispiel wird eine Definition für einen Flow (Workflow) abgerufen.

```

aws iotthingsgraph get-flow-template \
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"

```

Ausgabe:

```

{
  "description": {
    "summary": {
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
      "revisionNumber": 1,
      "createdAt": 1559247540.292
    },
    "definition": {
      "language": "GRAPHQL",
      "text": "{\nquery MyFlow($camera: string!, $screen: string!)
@workflowType(id: \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\")
@annotation(type: \"tgc:FlowEvent\", id: \"sledged790c1b2bcd949e09da0c9bfc077f79d
\", x: 1586, y: 653) @triggers(definition: \"{MotionSensor(description:
\\\\\"\\\\\") @position(x: 1045, y: 635.6666564941406) {\n  condition(expr:
\\\\\"devices[name == \\\\\\\\\\\\\\\\\\"motionSensor\\\\\\\\\\\\\\\\\"].events[name == \\\\\
\\\\\\\\\"StateChanged\\\\\\\\\\\\\\\\\"].lastEvent\\\\\\\\\")\n  action(expr: \\\\\\"\\\\\")\n
\n}}\") {\n  variables {\n    cameraResult @property(id: \"urn:tdm:aws/

```

```
examples:property:CameraStateProperty\")\n }\n steps {\n   step(name: \"Camera
\", outEvent: [\"sledged790c1b2bcd949e09da0c9bfc077f79d\"]) @position(x: 1377,
y: 638.6666564941406) {\n     DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Camera\", out: \"cameraResult\", deviceId: \"${camera}\")
{\n       capture\n     }\n   }\n   step(name: \"Screen\", inEvent:
[\"sledged790c1b2bcd949e09da0c9bfc077f79d\"]) @position(x: 1675.6666870117188,
y: 637.9999847412109) {\n     DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Screen\", deviceId: \"${screen}\") {\n       display(imageUrl:
\"${cameraResult.lastClickedImage}\")\n     }\n   }\n }\n }\n }\n }
},
  \"validatedNamespaceVersion\": 5
}
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetFlowTemplate](#) in der AWS CLI Befehlsreferenz.

get-namespace-deletion-status

Das folgende Codebeispiel zeigt die Verwendung `get-namespace-deletion-status`.

AWS CLI

Um den Status der Aufgabe zum Löschen von Namespaces abzurufen

Im folgenden `get-namespace-deletion-status` Beispiel wird der Status der Aufgabe zum Löschen von Namespaces abgerufen.

```
aws iotthingsgraph get-namespace-deletion-status
```

Ausgabe:

```
{
  \"namespaceArn\": \"arn:aws:iotthingsgraph:us-west-2:123456789012\",
  \"namespaceName\": \"us-west-2/123456789012/default\"
  \"status\": \"SUCCEEDED \"
}
```

Weitere Informationen finden Sie unter [Namespaces](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetNamespaceDeletionStatus](#) in AWS CLI der Befehlsreferenz.

get-system-instance

Das folgende Codebeispiel zeigt die Verwendung `get-system-instance`.

AWS CLI

Um eine Systeminstanz zu erhalten

Im folgenden `get-system-instance` Beispiel wird eine Definition für eine Systeminstanz abgerufen.

```
aws iotthingsgraph get-system-instance \
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Ausgabe:

```
{
  "description": {
    "summary": {
      "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218",
      "status": "NOT_DEPLOYED",
      "target": "CLOUD",
      "createdAt": 1559249315.208,
      "updatedAt": 1559249315.208
    },
    "definition": {
      "language": "GRAPHQL",
      "text": "{\r\nquery Room218 @deployment(id: \"urn:tdm:us-west-2/123456789012/default:Deployment:Room218\", systemId: \"urn:tdm:us-west-2/123456789012/default:System:SecurityFlow\") {\r\n  motionSensor(deviceId: \"MotionSensorName\")\r\n  screen(deviceId: \"ScreenName\")\r\n  camera(deviceId: \"CameraName\")\r\n  triggers {MotionEventTrigger(description: \"a trigger\") { \r\n    condition(expr: \"devices[name == 'motionSensor'].events[name == 'StateChanged'].lastEvent\") \r\n    action(expr: \"ThingsGraph.startFlow('SecurityFlow', bindings[name == 'camera'].deviceId, bindings[name == 'screen'].deviceId))\r\n  }\r\n  }\r\n  }\r\n  }\r\n  }"}"
    },
    "metricsConfiguration": {
```

```
        "cloudMetricEnabled": false
    },
    "validatedNamespaceVersion": 5,
    "flowActionsRoleArn": "arn:aws:iam::123456789012:role/ThingsGraphRole"
}
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSystemInstance](#) in der AWS CLI Befehlsreferenz.

get-system-template-revisions

Das folgende Codebeispiel zeigt die Verwendung `get-system-template-revisions`.

AWS CLI

Um Revisionsinformationen zu einem System abzurufen

Im folgenden `get-system-template-revisions` Beispiel werden Revisionsinformationen zu einem System abgerufen.

```
aws iotthingsgraph get-system-template-revisions \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Ausgabe:

```
{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/MySystem",
      "revisionNumber": 1,
      "createdAt": 1559247540.656
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSystemTemplateRevisions](#) in der AWS CLI Befehlsreferenz.

get-system-template

Das folgende Codebeispiel zeigt die Verwendung `get-system-template`.

AWS CLI

Um ein System zu bekommen

Das folgende `get-system-template` Beispiel ruft eine Definition für ein System ab.

```
aws iotthingsgraph get-system-template \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Ausgabe:

```
{
  "description": {
    "summary": {
      "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/MyFlow",
      "revisionNumber": 1,
      "createdAt": 1559247540.656
    },
    "definition": {
      "language": "GRAPHQL",
      "text": "{\n  type MySystem @systemType(id: \"urn:tdm:us-west-2/123456789012/default:System:MySystem\", description: \"\") {\n    camera: Camera @thing(id: \"urn:tdm:aws/examples:deviceModel:Camera\")\n    screen: Screen @thing(id: \"urn:tdm:aws/examples:deviceModel:Screen\")\n    motionSensor: MotionSensor @thing(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor\")\n    MyFlow: MyFlow @workflow(id: \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\")\n  }\n}"
    },
    "validatedNamespaceVersion": 5
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSystemTemplate](#) in der AWS CLI Befehlsreferenz.

get-upload-status

Das folgende Codebeispiel zeigt die Verwendung `get-upload-status`.

AWS CLI

Um den Status Ihrer Entität abzurufen, laden Sie sie hoch

Im folgenden `get-upload-status` Beispiel wird der Status Ihres Entitäts-Upload-Vorgangs abgerufen. Der Wert von `MyUploadId` ist der ID-Wert, der von der `upload-entity-definitions` Operation zurückgegeben wurde.

```
aws iotthingsgraph get-upload-status \  
  --upload-id "MyUploadId"
```

Ausgabe:

```
{  
  "namespaceName": "us-west-2/123456789012/default",  
  "namespaceVersion": 5,  
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",  
  "uploadStatus": "SUCCEEDED"  
}
```

Weitere Informationen finden Sie unter [Modeling Entities](#) im AWS IoT Things Graph User Guide.

- API-Einheiten finden Sie [GetUploadStatus](#) in der AWS CLI Befehlsreferenz.

list-flow-execution-messages

Das folgende Codebeispiel zeigt die Verwendung `list-flow-execution-messages`.

AWS CLI

Um Informationen über Ereignisse in einer Flow-Ausführung abzurufen

Im folgenden `list-flow-execution-messages` Beispiel werden Informationen zu Ereignissen in einer Flow-Ausführung abgerufen.

```
aws iotthingsgraph list-flow-execution-messages \  
  --namespace-name "us-west-2/123456789012/default" \  
  --namespace-version 5 \  
  --flow-id "f6294f1e-b109-4bbe-9073-f451a2dda2da" \  
  --start-time "2019-01-01T00:00:00Z" \  
  --end-time "2019-01-01T00:00:00Z" \  
  --limit 10
```



```
--flow-execution-id "urn:tdm:us-west-2/123456789012/  
default:Workflow:SecurityFlow_2019-05-11T19:39:55.317Z_MotionSensor_69b151ad-  
a611-42f5-ac21-fe537f9868ad"
```

Ausgabe:

```
{  
  "messages": [  
    {  
      "eventType": "EXECUTION_STARTED",  
      "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",  
      "payload": "Flow execution started",  
      "timestamp": 1559247540.656  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [ListFlowExecutionMessages](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um alle Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für eine AWS IoT Things Graph Graph-Ressource auf.

```
aws iotthingsgraph list-tags-for-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218"
```

Ausgabe:

```
{  
  "tags": [  
    {
```

```

        "key": "Type",
        "value": "Residential"
    }
]
}

```

Weitere Informationen finden Sie unter [Tagging Your AWS IoT Things Graph Resources](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

search-entities

Das folgende Codebeispiel zeigt die Verwendung `search-entities`.

AWS CLI

Um nach Entitäten zu suchen

Im folgenden `search-entities` Beispiel wird nach allen Entitäten des Typs `EVENT` gesucht.

```

aws iotthingsgraph search-entities \
  --entity-types "EVENT"

```

Ausgabe:

```

{
  "descriptions": [
    {
      "id": "urn:tdm:aws/examples:Event:MotionSensorEvent",
      "type": "EVENT",
      "definition": {
        "language": "GRAPHQL",
        "text": "##\n# Description of events emitted by motion
sensor.\n##\n# type MotionSensorEvent @eventType(id: \"urn:tdm:aws/
examples:event:MotionSensorEvent\", \n          payload: \"urn:tdm:aws/
examples:property:MotionSensorStateProperty\") {ignore:void}"
      }
    },
    {
      "id": "urn:tdm:us-west-2/123456789012/
default:Event:CameraClickedEventV2",
      "type": "EVENT",

```

```

      "definition": {
        "language": "GraphQL",
        "text": "type CameraClickedEventV2 @eventType(id: \"urn:tdm:us-
west-2/123456789012/default:event:CameraClickedEventV2\" ,\r\npayload:
 \"urn:tdm:aws:Property:Boolean\") {ignore:void}"
      }
    },
    {
      "id": "urn:tdm:us-west-2/123456789012/
default:Event:MotionSensorEventV2",
      "type": "EVENT",
      "definition": {
        "language": "GraphQL",
        "text": "# Event emitted by the motion sensor.\r\nntype
MotionSensorEventV2 @eventType(id: \"urn:tdm:us-west-2/123456789012/
default:event:MotionSensorEventV2\" ,\r\npayload: \"urn:tdm:us-west-2/123456789012/
default:property:MotionSensorStateProperty2\") {ignore:void}"
      }
    }
  ],
  "nextToken": "urn:tdm:us-west-2/123456789012/default:Event:MotionSensorEventV2"
}

```

Weitere Informationen finden Sie unter [AWS IoT Things Graph Data Model Reference](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [SearchEntities](#) unter AWS CLI Befehlsreferenz.

search-flow-executions

Das folgende Codebeispiel zeigt die Verwendung `search-flow-executions`.

AWS CLI

Um nach Flow-Ausführungen zu suchen

Im folgenden `search-flow-executions` Beispiel wird nach allen Ausführungen eines Flows in einer angegebenen Systeminstanz gesucht.

```

aws iotthingsgraph search-flow-executions \
  --system-instance-id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"

```

Ausgabe:

```
{
  "summaries": [
    {
      "createdAt": 1559247540.656,
      "flowExecutionId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
      "flowTemplateId": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
      "status": "RUNNING ",
      "systemInstanceId": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
      "updatedAt": 1559247540.656
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [SearchFlowExecutions](#) in der AWS CLI Befehlsreferenz.

search-flow-templates

Das folgende Codebeispiel zeigt die Verwendung `search-flow-templates`.

AWS CLI

Um nach Flows (oder Workflows) zu suchen

Im folgenden `search-flow-templates` Beispiel wird nach allen Flows (Workflows) gesucht, die das Kamera-Gerätemodell enthalten.

```
aws iotthingsgraph search-flow-templates \
  --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

Ausgabe:

```
{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
      "revisionNumber": 1,
      "createdAt": 1559247540.292
    },
  ],
}
```

```
{
  "id": "urn:tdm:us-west-2/123456789012/default:Workflow:SecurityFlow",
  "revisionNumber": 3,
  "createdAt": 1548283099.27
}
]
```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [SearchFlowTemplates](#) in der AWS CLI Befehlsreferenz.

search-system-instances

Das folgende Codebeispiel zeigt die Verwendung `search-system-instances`.

AWS CLI

Um nach Systeminstanzen zu suchen

Im folgenden `search-system-instances` Beispiel wird nach allen Systeminstanzen gesucht, die das angegebene System enthalten.

```
aws iotthingsgraph search-system-instances \
  --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
  default:System:SecurityFlow"
```

Ausgabe:

```
{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/
      default:Deployment:DeploymentForSample",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
      default/DeploymentForSample",
      "status": "NOT_DEPLOYED",
      "target": "GREENGRASS",
      "greengrassGroupName": "ThingsGraphGrnGr",
      "createdAt": 1555716314.707,
      "updatedAt": 1555716314.707
    },
  ],
}
```

```
{
  "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment",
  "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment",
  "status": "DELETED_IN_TARGET",
  "target": "GREENGRASS",
  "greengrassGroupName": "ThingsGraphGrnGr",
  "createdAt": 1549416462.049,
  "updatedAt": 1549416722.361,
  "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
  "greengrassGroupVersionId": "7365aed7-2d3e-4d13-aad8-75443d45eb05"
},
{
  "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment2",
  "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment2",
  "status": "DEPLOYED_IN_TARGET",
  "target": "GREENGRASS",
  "greengrassGroupName": "ThingsGraphGrnGr",
  "createdAt": 1549572385.774,
  "updatedAt": 1549572418.408,
  "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
  "greengrassGroupVersionId": "bfa70ab3-2bf7-409c-a4d4-bc8328ae5b86"
},
{
  "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",
  "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room215",
  "status": "NOT_DEPLOYED",
  "target": "GREENGRASS",
  "greengrassGroupName": "ThingsGraphGG",
  "createdAt": 1547056918.413,
  "updatedAt": 1547056918.413
},
{
  "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
  "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218",
  "status": "NOT_DEPLOYED",
  "target": "CLOUD",
  "createdAt": 1559249315.208,
  "updatedAt": 1559249315.208
}
```

```
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Systemen und Ablaufkonfigurationen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [SearchSystemInstances](#) in der AWS CLI Befehlsreferenz.

search-system-templates

Das folgende Codebeispiel zeigt die Verwendung `search-system-templates`.

AWS CLI

Um nach einem System zu suchen

Im folgenden `search-system-templates` Beispiel wird nach allen Systemen gesucht, die den angegebenen Flow enthalten.

```
aws iotthingsgraph search-system-templates \  
  --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/  
  default:Workflow:SecurityFlow"
```

Ausgabe:

```
{  
  "summaries": [  
    {  
      "id": "urn:tdm:us-west-2/123456789012/default:System:SecurityFlow",  
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/  
SecurityFlow",  
      "revisionNumber": 1,  
      "createdAt": 1548283099.433  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [SearchSystemTemplates](#) in der AWS CLI Befehlsreferenz.

search-things

Das folgende Codebeispiel zeigt die Verwendung `search-things`.

AWS CLI

Um nach Dingen zu suchen, die mit Geräten und Gerätemodellen verknüpft sind

Im folgenden `search-things` Beispiel wird nach allen Dingen gesucht, die mit dem MotionSensor Gerät HCSR5 01 verknüpft sind.

```
aws iotthingsgraph search-things \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Ausgabe:

```
{  
  "things": [  
    {  
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MotionSensor1",  
      "thingName": "MotionSensor1"  
    },  
    {  
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/TG_MS",  
      "thingName": "TG_MS"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Modelle erstellen und hochladen](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- APIEinzelheiten finden Sie [SearchThings](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Tag für eine Ressource zu erstellen

Im folgenden `tag-resource` Beispiel wird ein Tag für die angegebene Ressource erstellt.


```
aws iotthingsgraph tag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218" \  
  --tags key="Type",value="Residential"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your AWS IoT Things Graph Resources](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

undeploy-system-instance

Das folgende Codebeispiel zeigt die Verwendung `undeploy-system-instance`.

AWS CLI

Um die Bereitstellung einer Systeminstanz von ihrem Ziel aufzuheben

Im folgenden `undeploy-system-instance` Beispiel wird eine Systeminstanz von ihrem Ziel entfernt.

```
aws iotthingsgraph undeploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

Ausgabe:

```
{  
  "summary": {  
    "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",  
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/  
Room215",  
    "status": "PENDING_DELETE",  
    "target": "GREENGRASS",  
    "greengrassGroupName": "ThingsGraphGrnGr",  
    "createdAt": 1553189694.255,  
    "updatedAt": 1559344549.601,  
    "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",  
    "greengrassGroupVersionId": "731b371d-d644-4b67-ac64-3934e99b75d7"  
  }  
}
```

Weitere Informationen finden Sie unter [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [UndeploySystemInstance](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag für eine Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird ein Tag für die angegebene Ressource entfernt.

```
aws iotthingsgraph untag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218" \  
  --tag-keys "Type"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your AWS IoT Things Graph Resources](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-flow-template

Das folgende Codebeispiel zeigt die Verwendung `update-flow-template`.

AWS CLI

Um einen Flow zu aktualisieren

Das folgende `update-flow-template` Beispiel aktualisiert einen Flow (Workflow). Der Wert von `MyFlowDefinition` ist das GraphQL, das den Fluss modelliert.

```
aws iotthingsgraph update-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

Ausgabe:

```
{
  "summary": {
    "createdAt": 1559248067.545,
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
    "revisionNumber": 2
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Flows](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateFlowTemplate](#) in der AWS CLI Befehlsreferenz.

update-system-template

Das folgende Codebeispiel zeigt die Verwendung `update-system-template`.

AWS CLI

Um ein System zu aktualisieren

Das folgende `update-system-template` Beispiel aktualisiert ein System. Der Wert von `MySystemDefinition` ist das GraphQL, das das System modelliert.

```
aws iotthingsgraph update-system-template \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \
  --definition language=GRAPHQL,text="MySystemDefinition"
```

Ausgabe:

```
{
  "summary": {
    "createdAt": 1559249776.254,
    "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/MySystem",
    "revisionNumber": 2
  }
}
```

Weitere Informationen finden Sie unter [Creating Systems](#) im AWS IoT Things Graph Graph-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSystemTemplate](#) in der AWS CLI Befehlsreferenz.

upload-entity-definitions

Das folgende Codebeispiel zeigt die Verwendung `upload-entity-definitions`.

AWS CLI

Um Entitätsdefinitionen hochzuladen

Im folgenden `upload-entity-definitions` Beispiel werden Entitätsdefinitionen in Ihren Namespace hochgeladen. Der Wert von `MyEntityDefinitions` ist das GraphQL, das die Entitäten modelliert.

```
aws iotthingsgraph upload-entity-definitions \  
  --document language=GRAPHQL,text="MyEntityDefinitions"
```

Ausgabe:

```
{  
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"  
}
```

Weitere Informationen finden Sie unter [Modeling Entities](#) im AWS IoT Things Graph User Guide.

- API-Einheiten finden Sie [UploadEntityDefinitions](#) in der AWS CLI Befehlsreferenz.

AWS IoT Wireless Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS IoT Wireless.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-aws-account-with-partner-account

Das folgende Codebeispiel zeigt die Verwendung `associate-aws-account-with-partner-account`.

AWS CLI

Um ein Partnerkonto mit Ihrem AWS Konto zu verknüpfen

Im folgenden `associate-aws-account-with-partner-account` Beispiel werden die folgenden Anmeldeinformationen für das Sidewalk-Konto Ihrem AWS Konto zugeordnet.

```
aws iotwireless associate-aws-account-with-partner-account \  
  --sidewalk  
  AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78
```

Ausgabe:

```
{  
  "Sidewalk": {  
    "AmazonId": "12345678901234",  
    "AppServerPrivateKey":  
    "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"  
  }  
}
```

Weitere Informationen finden Sie unter [Amazon Sidewalk Integration for AWS IoT Core](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AssociateAwsAccountWithPartnerAccount](#) in der AWS CLI Befehlsreferenz.

associate-wireless-device-with-thing

Das folgende Codebeispiel zeigt die Verwendung `associate-wireless-device-with-thing`.

AWS CLI

Um ein Objekt einem drahtlosen Gerät zuzuordnen

Im folgenden `associate-wireless-device-with-thing` Beispiel wird Ihrem drahtlosen Gerät, das die angegebene ID hat, ein Objekt zugeordnet.

```
aws iotwireless associate-wireless-device-with-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AssociateWirelessDeviceWithThing](#) in der AWS CLI Befehlsreferenz.

`associate-wireless-gateway-with-certificate`

Das folgende Codebeispiel zeigt die Verwendung `associate-wireless-gateway-with-certificate`.

AWS CLI

Um das Zertifikat dem Wireless-Gateway zuzuordnen

Im Folgenden `associate-wireless-gateway-with-certificate` wird ein drahtloses Gateway einem Zertifikat zugeordnet.

```
aws iotwireless associate-wireless-gateway-with-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --iot-certificate-  
  id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

Ausgabe:

```
{  
  "IotCertificateId":  
  "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"  
}
```

Weitere Informationen finden [Sie unter Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AssociateWirelessGatewayWithCertificate](#) in der AWS CLI Befehlsreferenz.

associate-wireless-gateway-with-thing

Das folgende Codebeispiel zeigt die Verwendung `associate-wireless-gateway-with-thing`.

AWS CLI

Um einem drahtlosen Gateway eine Sache zuzuordnen

Das folgende `associate-wireless-gateway-with-thing` Beispiel ordnet einem drahtlosen Gateway ein Ding zu.

```
aws iotwireless associate-wireless-gateway-with-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [AssociateWirelessGatewayWithThing](#) in der AWS CLI Befehlsreferenz.

create-destination

Das folgende Codebeispiel zeigt die Verwendung `create-destination`.

AWS CLI

Um ein drahtloses IoT-Ziel zu erstellen

Im folgenden `create-destination` Beispiel wird ein Ziel für die Zuordnung einer Gerätenachricht zu einer AWS IoT-Regel erstellt. Bevor Sie diesen Befehl ausführen, müssen Sie eine IAM Rolle erstellt haben, die AWS IoT Core LoRa WAN die zum Senden von Daten an die AWS IoT-Regel erforderlichen Berechtigungen erteilt.

```
aws iotwireless create-destination \  
  --name IoTWirelessDestination \  
  --
```

```
--expression-type RuleName \  
--expression IoTWirelessRule \  
--role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

Ausgabe:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination",  
  "Name": "IoTWirelessDestination"  
}
```

Weitere Informationen finden [Sie unter Hinzufügen von Zielen zu AWS IoT Core LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateDestination](#) in der AWS CLI Befehlsreferenz.

create-device-profile

Das folgende Codebeispiel zeigt die Verwendung `create-device-profile`.

AWS CLI

Um ein neues Geräteprofil zu erstellen

Im folgenden `create-device-profile` Beispiel wird ein neues drahtloses IoT-Geräteprofil erstellt.

```
aws iotwireless create-device-profile
```

Ausgabe:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-  
a1b2-3c45-67d8-e90fa1b2c34d",  
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateDeviceProfile](#) in der AWS CLI Befehlsreferenz.

create-service-profile

Das folgende Codebeispiel zeigt die Verwendung `create-service-profile`.

AWS CLI

Um ein neues Dienstprofil zu erstellen

Im folgenden `create-service-profile` Beispiel wird ein neues drahtloses IoT-Dienstprofil erstellt.

```
aws iotwireless create-service-profile
```

Ausgabe:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateServiceProfile](#) in der AWS CLI Befehlsreferenz.

create-wireless-device

Das folgende Codebeispiel zeigt die Verwendung `create-wireless-device`.

AWS CLI

Um ein drahtloses IoT-Gerät zu erstellen

Im folgenden `create-wireless-device` Beispiel wird eine WLAN-Geräteressource des Typs `LoRaWAN` erstellt.

```
aws iotwireless create-wireless-device \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "Description": "My LoRaWAN wireless device"
  "DestinationName": "IoTWirelessDestination"
  "LoRaWAN": {
    "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
    "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
    "OtaaV1_1": {
      "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
      "JoinEui": "b4c231a359bc2e3d",
      "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
    },
    "DevEui": "ac12efc654d23fc2"
  },
  "Name": "SampleIoTWirelessThing"
  "Type": LoRaWAN
}
```

Ausgabe:

```
{
  "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
  "Id": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateWirelessDevice](#) in der AWS CLI Befehlsreferenz.

create-wireless-gateway-task-definition

Das folgende Codebeispiel zeigt die Verwendung `create-wireless-gateway-task-definition`.

AWS CLI

Um eine Aufgabendefinition für ein drahtloses Gateway zu erstellen

Im Folgenden `create-wireless-gateway-task-definition` werden automatisch Aufgaben erstellt, die diese Aufgabendefinition für alle Gateways mit der angegebenen aktuellen Version verwenden.

```
aws iotwireless create-wireless-gateway-task-definition \  
--cli-input-json file://input.json
```

Inhalt von input.json:

```
{  
  "AutoCreateTasks": true,  
  "Name": "TestAutoUpdate",  
  "Update": {  
    "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",  
    "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",  
    "LoRaWAN" : {  
      "CurrentVersion" : {  
        "PackageVersion" : "1.0.0",  
        "Station" : "2.0.5",  
        "Model" : "linux"  
      },  
      "UpdateVersion" : {  
        "PackageVersion" : "1.0.1",  
        "Station" : "2.0.5",  
        "Model" : "minihub"  
      }  
    }  
  }  
}
```

Ausgabe:

```
{  
  "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"  
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateWirelessGatewayTaskDefinition](#) in der AWS CLI Befehlsreferenz.

create-wireless-gateway-task

Das folgende Codebeispiel zeigt die Verwendung `create-wireless-gateway-task`.

AWS CLI

Um die Aufgabe für ein drahtloses Gateway zu erstellen

Im folgenden `create-wireless-gateway-task` Beispiel wird eine Aufgabe für ein drahtloses Gateway erstellt.

```
aws iotwireless create-wireless-gateway-task \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"
```

Ausgabe:

```
{  
  "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",  
  "Status": "Success"  
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateWirelessGatewayTask](#) in der AWS CLI Befehlsreferenz.

`create-wireless-gateway`

Das folgende Codebeispiel zeigt die Verwendung `create-wireless-gateway`.

AWS CLI

Um ein drahtloses Gateway zu erstellen

Im folgenden `create-wireless-gateway` Beispiel wird ein Gateway für drahtlose LoRa WAN Geräte erstellt.

```
aws iotwireless create-wireless-gateway \  
  --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \  
  --name "myFirstLoRaWANGateway" \  
  --description "Using my first LoRaWAN gateway"
```

Ausgabe:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [CreateWirelessGateway](#) in der AWS CLI Befehlsreferenz.

delete-destination

Das folgende Codebeispiel zeigt die Verwendung `delete-destination`.

AWS CLI

Um ein drahtloses IoT-Ziel zu löschen

Im folgenden `delete-destination` Beispiel wird die WLAN-Zielressource mit dem von Ihnen `IoWirelessDestination` erstellten Namen gelöscht.

```
aws iotwireless delete-destination \
  --name "IoWirelessDestination"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen von Zielen zu AWS IoT Core LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteDestination](#) in der AWS CLI Befehlsreferenz.

delete-device-profile

Das folgende Codebeispiel zeigt die Verwendung `delete-device-profile`.

AWS CLI

Um ein Geräteprofil zu löschen

Im folgenden `delete-device-profile` Beispiel wird ein Geräteprofil mit der angegebenen ID gelöscht, das Sie erstellt haben.

```
aws iotwireless delete-device-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- APIEinzelheiten finden Sie [DeleteDeviceProfile](#) in der AWS CLI Befehlsreferenz.

delete-service-profile

Das folgende Codebeispiel zeigt die Verwendung `delete-service-profile`.

AWS CLI

Um ein Dienstprofil zu löschen

Im folgenden `delete-service-profile` Beispiel wird ein Dienstprofil mit der angegebenen ID gelöscht, das Sie erstellt haben.

```
aws iotwireless delete-service-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- APIEinzelheiten finden Sie [DeleteServiceProfile](#) in der AWS CLI Befehlsreferenz.

delete-wireless-device

Das folgende Codebeispiel zeigt die Verwendung `delete-wireless-device`.

AWS CLI

Um ein drahtloses Gerät zu löschen

Im folgenden `delete-wireless-device` Beispiel wird ein drahtloses Gerät mit der angegebenen ID gelöscht.

```
aws iotwireless delete-wireless-device \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteWirelessDevice](#) in der AWS CLI Befehlsreferenz.

delete-wireless-gateway-task-definition

Das folgende Codebeispiel zeigt die Verwendung `delete-wireless-gateway-task-definition`.

AWS CLI

Um eine Aufgabendefinition für ein drahtloses Gateway zu löschen

Im folgenden `delete-wireless-gateway-task-definition` Beispiel wird die Aufgabendefinition für das drahtlose Gateway gelöscht, die Sie mit der folgenden ID erstellt haben.

```
aws iotwireless delete-wireless-gateway-task-definition \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteWirelessGatewayTaskDefinition](#) in der AWS CLI Befehlsreferenz.

delete-wireless-gateway-task

Das folgende Codebeispiel zeigt die Verwendung `delete-wireless-gateway-task`.

AWS CLI

Um eine Aufgabe für ein drahtloses Gateway zu löschen

Im folgenden `delete-wireless-gateway-task` Beispiel wird die Aufgabe für ein drahtloses Gateway gelöscht, die die angegebene ID hat.

```
aws iotwireless delete-wireless-gateway-task \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteWirelessGatewayTask](#) in der AWS CLI Befehlsreferenz.

delete-wireless-gateway

Das folgende Codebeispiel zeigt die Verwendung `delete-wireless-gateway`.

AWS CLI

Um ein drahtloses Gateway zu löschen

Im folgenden `delete-wireless-gateway` Beispiel wird ein drahtloses Gateway mit der angegebenen ID gelöscht.

```
aws iotwireless delete-wireless-gateway \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DeleteWirelessGateway](#) in der AWS CLI Befehlsreferenz.

disassociate-aws-account-from-partner-account

Das folgende Codebeispiel zeigt die Verwendung `disassociate-aws-account-from-partner-account`.

AWS CLI

Um das Partnerkonto vom Konto zu trennen AWS

Im folgenden `disassociate-aws-account-from-partner-account` Beispiel wird ein Partnerkonto von Ihrem aktuell verknüpften AWS Konto getrennt.

```
aws iotwireless disassociate-aws-account-from-partner-account \  
  --partner-account-id "12345678901234" \  
  --partner-type "Sidewalk"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DisassociateAwsAccountFromPartnerAccount](#) in der AWS CLI Befehlsreferenz.

disassociate-wireless-device-from-thing

Das folgende Codebeispiel zeigt die Verwendung `disassociate-wireless-device-from-thing`.

AWS CLI

Um die Verbindung zwischen dem Ding und dem drahtlosen Gerät zu trennen

Im folgenden `disassociate-wireless-device-from-thing` Beispiel wird die Verbindung zwischen einem drahtlosen Gerät und dem aktuell verknüpften Gerät getrennt.

```
aws iotwireless disassociate-wireless-device-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DisassociateWirelessDeviceFromThing](#) in der AWS CLI Befehlsreferenz.

disassociate-wireless-gateway-from-certificate

Das folgende Codebeispiel zeigt die Verwendung `disassociate-wireless-gateway-from-certificate`.

AWS CLI

Um die Zuordnung des Zertifikats zum Wireless-Gateway zu trennen

Im Folgenden wird die Zuordnung eines `disassociate-wireless-gateway-from-certificate` drahtlosen Gateways zu seinem aktuell verknüpften Zertifikat getrennt.

```
aws iotwireless disassociate-wireless-gateway-from-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DisassociateWirelessGatewayFromCertificate](#) in der AWS CLI Befehlsreferenz.

`disassociate-wireless-gateway-from-thing`

Das folgende Codebeispiel zeigt die Verwendung `disassociate-wireless-gateway-from-thing`.

AWS CLI

Um die Verbindung zwischen dem Ding und dem drahtlosen Gateway zu trennen

Im folgenden `disassociate-wireless-gateway-from-thing` Beispiel wird die Zuordnung eines drahtlosen Gateways zu seinem aktuell verknüpften Objekt getrennt.

```
aws iotwireless disassociate-wireless-gateway-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Hinzufügen Ihrer Gateways und drahtlosen Geräte zu AWS IoT Core für LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [DisassociateWirelessGatewayFromThing](#) in der AWS CLI Befehlsreferenz.

get-destination

Das folgende Codebeispiel zeigt die Verwendung `get-destination`.

AWS CLI

Um Informationen über ein drahtloses IoT-Ziel zu erhalten

Im folgenden `get-destination` Beispiel werden Informationen zur Zielressource mit dem von Ihnen `IoTWirelessDestination` erstellten Namen abgerufen.

```
aws iotwireless get-destination \  
  --name "IoTWirelessDestination"
```

Ausgabe:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination",  
  "Name": "IoTWirelessDestination",  
  "Expression": "IoTWirelessRule",  
  "ExpressionType": "RuleName",  
  "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"  
}
```

Weitere Informationen finden [Sie unter Hinzufügen von Zielen zu AWS IoT Core LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetDestination](#) in der AWS CLI Befehlsreferenz.

get-device-profile

Das folgende Codebeispiel zeigt die Verwendung `get-device-profile`.

AWS CLI

Um Informationen über ein Geräteprofil abzurufen

Im folgenden `get-device-profile` Beispiel werden Informationen über das Geräteprofil mit der angegebenen ID abgerufen, das Sie erstellt haben.

```
aws iotwireless get-device-profile \  
  --device-profile-id "IoTWirelessDeviceProfile"
```

```
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Ausgabe:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "LoRaWAN": {
    "MacVersion": "1.0.3",
    "MaxDutyCycle": 10,
    "Supports32BitFCnt": false,
    "RegParamsRevision": "RP002-1.0.1",
    "SupportsJoin": true,
    "RfRegion": "US915",
    "MaxEirp": 13,
    "SupportsClassB": false,
    "SupportsClassC": false
  }
}
```

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetDeviceProfile](#) in der AWS CLI Befehlsreferenz.

get-partner-account

Das folgende Codebeispiel zeigt die Verwendung von `get-partner-account`.

AWS CLI

Um die Informationen zum Partnerkonto abzurufen

Im folgenden `get-partner-account` Beispiel werden Informationen zu Ihrem Sidewalk-Konto abgerufen, das die folgende ID hat.

```
aws iotwireless get-partner-account \
  --partner-account-id "12345678901234" \
  --partner-type "Sidewalk"
```

Ausgabe:

```
{
  "Sidewalk": {
    "AmazonId": "12345678901234",
    "Fingerprint":
"a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
  },
  "AccountLinked": false
}
```

Weitere Informationen finden Sie unter [Amazon Sidewalk Integration for AWS IoT Core](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetPartnerAccount](#) in der AWS CLI Befehlsreferenz.

get-service-endpoint

Das folgende Codebeispiel zeigt die Verwendung `get-service-endpoint`.

AWS CLI

Um den Service-Endpoint zu erhalten

Im folgenden `get-service-endpoint` Beispiel wird der kontospezifische Endpunkt für CUPS das Protokoll abgerufen.

```
aws iotwireless get-service-endpoint
```

Ausgabe:

```
{
  "ServiceType": "CUPS",
  "ServiceEndpoint": "https://A1RMKZ37ACAGOT.cups.lorawan.us-east-1.amazonaws.com:443",
  "ServerTrust": "-----BEGIN CERTIFICATE-----\n
MIIESTCCAzGgAwIBAgITBn+UV4WH6Kx33rJTMlu8mYtWDTANBgkqhkiG9w0BAQsF\n
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDEExBBbWF6\n
b24gUm9vdCBDQSAxMB4XDTE1MTAyMjAwMDAwMFoXDTE1MTAxOTAwMDAwMFowRjEL\n
MAKGA1UEBhMCMVVMxZDzANBgNVBAoTBkFtYXpvcjEVMGMGA1UECjEUMTUyVjdmVjIENB\n
IDFCMQ8wDQYDVQQDEwZBbWF6b24wgwEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n
AoIBAQCThZn3c68asg3Wuw6MLAd5tES6BIOsMzoKcG5b1PVo+sD0RrMd4f2AbnZ\n
cMzPa43j4wNxphty6aUKk4T1qe9B0wKFjwK6zmxXLVYo7bHViXsP1J6q0MpFge5\n
b1DP+18x+B26A0piiQ0uPkfyDyeR4xQghfj66Yo19V+emU3nazfvpFA+R0z6WoVm\n
-----"
```

```

B5x+F2pV8xeKNR7u6azDdU5YVX1TawpımxRC1+WsAYmz6qP+z8ArDITC2FMVy2fw\n
0IjK0tEXc/VfmtTFch5+AfGYMGmqqvJ6LcXiAhqG5TI+Dr0RtM88k+8XUBCeQ8IG\n
KuANaL7TiItKZYxK1MMuTJtV9Ib1AgMBAAGjggE7MIIBNzASBgNVHRMBAf8ECDAG\n
AQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUWaRmBlKge5WSPK0UByeW\n
dFv5PdAwHwYDVR0jBBgwFoAUhBjMhTTsvAyU1C4IWZzHshB0CggwewYIKwYBBQUH\n
AQEEbzBtMC8GCCsGAQUFBzABhiNodHRwOi8vb2NzcC5yb290Y2ExLmFtYXpvbnRy\n
dXN0LmNvbTA6BgggrBgEFBQcwAoYuaHR0cDovL2NydC5yb290Y2ExLmFtYXpvbnRy\n
dXN0LmNvbS9yb290Y2ExLmNlclcA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3Js\n
LnJvb3RjYTEuYW1hem9udHJ1c3QuY29tL3Jvb3RjYTEuY3JsMBMGA1UdIAQMMMAow\n
CAYGZ4EMAQIBMA0GCSqGSIb3DQEBCwUAA4IBAQCfkr41u3nPo4FCH0TjY3NT0VI1\n
59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss21JwJFqMp2PphKg5625kXg8kP2CN5t\n
6G7bMQcT8C8xDZntYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n
8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB480z+ctBWuZSkbvkNodPLamkB2g1\n
upRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtkPISLMVCx7i4hncxHZS\n
yLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n
-----END CERTIFICATE-----\n"
}

```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetServiceEndpoint](#) in der AWS CLI Befehlsreferenz.

get-service-profile

Das folgende Codebeispiel zeigt die Verwendung `get-service-profile`.

AWS CLI

Um Informationen über ein Dienstprofil abzurufen

Im folgenden `get-service-profile` Beispiel werden Informationen über das Dienstprofil mit der angegebenen ID abgerufen, das Sie erstellt haben.

```

aws iotwireless get-service-profile \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"

```

Ausgabe:

```

{
  "Arn": "arn:aws:iotwireless:us-east-1:651419225604:ServiceProfile/538185bb-
d7e7-4b95-96a0-c51aa4a5b9a0",

```

```
"Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
"LoRaWAN": {
  "HrAllowed": false,
  "NwkGeoLoc": false,
  "DrMax": 15,
  "UlBucketSize": 4096,
  "PrAllowed": false,
  "ReportDevStatusBattery": false,
  "DrMin": 0,
  "DlRate": 60,
  "AddGwMetadata": false,
  "ReportDevStatusMargin": false,
  "MinGwDiversity": 1,
  "RaAllowed": false,
  "DlBucketSize": 4096,
  "DevStatusReqFreq": 24,
  "TargetPer": 5,
  "UlRate": 60
}
}
```

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetServiceProfile](#) in der AWS CLI Befehlsreferenz.

get-wireless-device-statistics

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-device-statistics`.

AWS CLI

Um Betriebsinformationen zu einem drahtlosen Gerät abzurufen

Im folgenden `get-wireless-device-statistics` Beispiel werden Betriebsinformationen zu einem drahtlosen Gerät abgerufen.

```
aws iotwireless get-wireless-device-statistics \
  --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

Ausgabe:

```
{
```

```
"WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessDeviceStatistics](#) in der AWS CLI Befehlsreferenz.

get-wireless-device

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-device`.

AWS CLI

Um Informationen über das drahtlose Gerät zu erhalten

Das folgende `get-wireless-device` Beispiel listet die verfügbaren Widgets in Ihrem AWS Konto auf.

```
aws iotwireless get-wireless-device \
  --identifizier "1ffd32c8-8130-4194-96df-622f072a315f" \
  --identifizier-type WirelessDeviceID
```

Ausgabe:

```
{
  "Name": "myLoRaWANDevice",
  "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/44b87eb4-9bce-423d-
b5fc-973f5ecc358b",
  "DestinationName": "IoTWirelessDestination",
  "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
  "ThingName": "44b87eb4-9bce-423d-b5fc-973f5ecc358b",
  "Type": "LoRaWAN",
  "LoRaWAN": {
    "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
    "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
    "OtaaV1_1": {
      "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
      "JoinEui": "b4c231a359bc2e3d",
      "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
    },
    "DevEui": "ac12efc654d23fc2"
  }
}
```



```
  },
  "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1fffd32c8-8130-4194-96df-622f072a315f",
  "Description": "My LoRaWAN wireless device"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessDevice](#) in der AWS CLI Befehlsreferenz.

get-wireless-gateway-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-gateway-certificate`.

AWS CLI

Um die ID eines Zertifikats abzurufen, das einem drahtlosen Gateway zugeordnet ist

Im folgenden `get-wireless-gateway-certificate` Beispiel wird die Zertifikat-ID abgerufen, die einem drahtlosen Gateway mit der angegebenen ID zugeordnet ist.

```
aws iotwireless get-wireless-gateway-certificate \
  --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

Ausgabe:

```
{
  "IotCertificateId":
  "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessGatewayCertificate](#) in der AWS CLI Befehlsreferenz.

get-wireless-gateway-firmware-information

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-gateway-firmware-information`.

AWS CLI

Um Firmware-Informationen über ein drahtloses Gateway abzurufen

Im folgenden `get-wireless-gateway-firmware-information` Beispiel werden die Firmware-Version und andere Informationen zu einem drahtlosen Gateway abgerufen.

```
aws iotwireless get-wireless-gateway-firmware-information \
  --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Ausgabe:

```
{
  "LoRaWAN" :{
    "CurrentVersion" :{
      "PackageVersion" : "1.0.0",
      "Station" : "2.0.5",
      "Model" : "linux"
    }
  }
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessGatewayFirmwareInformation](#) in der AWS CLI Befehlsreferenz.

`get-wireless-gateway-statistics`

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-gateway-statistics`.

AWS CLI

Um Betriebsinformationen zu einem drahtlosen Gateway abzurufen

Im folgenden `get-wireless-gateway-statistics` Beispiel werden Betriebsinformationen zu einem drahtlosen Gateway abgerufen.

```
aws iotwireless get-wireless-gateway-statistics \
  --wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Ausgabe:

```
{
  "WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessGatewayStatistics](#) in der AWS CLI Befehlsreferenz.

get-wireless-gateway-task-definition

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-gateway-task-definition`.

AWS CLI

Um Informationen über eine Aufgabendefinition für ein drahtloses Gateway abzurufen

Im folgenden `get-wireless-gateway-task-definition` Beispiel werden Informationen zur Aufgabendefinition für ein drahtloses Netzwerk mit der angegebenen ID abgerufen.

```
aws iotwireless get-wireless-gateway-task-definition \
  --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

Ausgabe:

```
{
  "AutoCreateTasks": true,
  "Name": "TestAutoUpdate",
  "Update": {
    "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",
    "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",
    "LoRaWAN" : {
      "CurrentVersion" : {
        "PackageVersion" : "1.0.0",
        "Station" : "2.0.5",
        "Model" : "linux"
      },
      "UpdateVersion" : {
        "PackageVersion" : "1.0.1",
        "Station" : "2.0.5",

```

```
    "Model" : "minihub"  
  }  
}   
}   
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessGatewayTaskDefinition](#) in der AWS CLI Befehlsreferenz.

get-wireless-gateway-task

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-gateway-task`.

AWS CLI

Um Informationen über die Wireless Gateway-Aufgabe abzurufen

Im folgenden `get-wireless-gateway-task` Beispiel werden Informationen zur Aufgabe für das drahtlose Gateway mit der angegebenen ID abgerufen.

```
aws iotwireless get-wireless-gateway-task \   
  --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

Ausgabe:

```
{   
  "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",   
  "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",   
  "Status": "Success"   
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessGatewayTask](#) in der AWS CLI Befehlsreferenz.

get-wireless-gateway

Das folgende Codebeispiel zeigt die Verwendung `get-wireless-gateway`.

AWS CLI

Um Informationen über ein drahtloses Gateway zu erhalten

Im folgenden `get-wireless-gateway` Beispiel werden Informationen über das Wireless-Gateway abgerufen `myFirstLoRaWANGateway`.

```
aws iotwireless get-wireless-gateway \
  --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --identifier-type WirelessGatewayId
```

Ausgabe:

```
{
  "Description": "My first LoRaWAN gateway",
  "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/a1b2c3d4-5678-90ab-cdef-12ab345c67de",
  "LoRaWAN": {
    "RfRegion": "US915",
    "GatewayEui": "a1b2c3d4567890ab"
  },
  "ThingName": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/6c44ab31-8b4d-407a-bed3-19b6c7cda551",
  "Name": "myFirstLoRaWANGateway"
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [GetWirelessGateway](#) in der AWS CLI Befehlsreferenz.

list-destinations

Das folgende Codebeispiel zeigt die Verwendung `list-destinations`.

AWS CLI

Um die drahtlosen Ziele aufzulisten

Das folgende `list-destinations` Beispiel listet die verfügbaren Ziele auf, die für Ihr AWS Konto registriert sind.

```
aws iotwireless list-destinations
```

Ausgabe:

```
{
  "DestinationList": [
    {
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
      "Name": "IoTWirelessDestination",
      "Expression": "IoTWirelessRule",
      "Description": "Destination for messages processed using
IoTWirelessRule",
      "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
    },
    {
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination2",
      "Name": "IoTWirelessDestination2",
      "Expression": "IoTWirelessRule2",
      "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
    }
  ]
}
```

Weitere Informationen finden [Sie unter Hinzufügen von Zielen zu AWS IoT Core LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListDestinations](#) in der AWS CLI Befehlsreferenz.

list-device-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-device-profiles`.

AWS CLI

Um die Geräteprofile aufzulisten

Das folgende `list-device-profiles` Beispiel listet die verfügbaren Geräteprofile auf, die für Ihr AWS Konto registriert sind.

```
aws iotwireless list-device-profiles
```

Ausgabe:

```
{
  "DeviceProfileList": [
    {
      "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d"
    },
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/a1b2c3d4-5678-90ab-cdef-12ab345c67de"
    }
  ]
}
```

Weitere Informationen finden [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListDeviceProfiles](#) in der AWS CLI Befehlsreferenz.

list-partner-accounts

Das folgende Codebeispiel zeigt die Verwendung `list-partner-accounts`.

AWS CLI

Um die Partnerkonten aufzulisten

Das folgende `list-partner-accounts` Beispiel listet die verfügbaren Partnerkonten auf, die mit Ihrem AWS Konto verknüpft sind.

```
aws iotwireless list-partner-accounts
```

Ausgabe:

```
{
  "Sidewalk": [
    {
      "AmazonId": "78965678771228",

```

```
    "Fingerprint":
      "bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
    },
    {
      "AmazonId": "89656787651228",
      "Fingerprint":
        "bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon Sidewalk Integration for AWS IoT Core](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListPartnerAccounts](#) in der AWS CLI Befehlsreferenz.

list-service-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-service-profiles`.

AWS CLI

Um die Dienstprofile aufzulisten

Im folgenden `list-service-profiles` Beispiel werden die verfügbaren Dienstprofile aufgeführt, die für Ihr AWS Konto registriert sind.

```
aws iotwireless list-service-profiles
```

Ausgabe:

```
{
  "ServiceProfileList": [
    {
      "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/538185bb-d7e7-4b95-96a0-c51aa4a5b9a0"
    },
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/ea8bc823-5d13-472e-8d26-9550737d8100"
    }
  ]
}
```



```
]
}
```

Weitere Informationen finden Sie unter [Sie unter Profile zu AWS IoT Core hinzufügen LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListServiceProfiles](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die der Ressource zugewiesenen Tags aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags auf, die einer drahtlosen Zielressource zugewiesen sind.

```
aws iotwireless list-tags-for-resource \
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
  IoWirelessDestination"
```

Ausgabe:

```
{
  "Tags": [
    {
      "Value": "MyValue",
      "Key": "MyTag"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Beschreiben Sie Ihren AWS IoT Core für LoRa WAN Ressourcen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-wireless-devices

Das folgende Codebeispiel zeigt die Verwendung `list-wireless-devices`.

AWS CLI

Um die verfügbaren drahtlosen Geräte aufzulisten

Das folgende `list-wireless-devices` Beispiel listet die verfügbaren drahtlosen Geräte auf, die für Ihr AWS Konto registriert sind.

```
aws iotwireless list-wireless-devices
```

Ausgabe:

```
{
  "WirelessDeviceList": [
    {
      "Name": "myLoRaWANDevice",
      "DestinationName": "IoTWirelessDestination",
      "Id": "1fffd32c8-8130-4194-96df-622f072a315f",
      "Type": "LoRaWAN",
      "LoRaWAN": {
        "DevEui": "ac12efc654d23fc2"
      },
      "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1fffd32c8-8130-4194-96df-622f072a315f"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListWirelessDevices](#) in der AWS CLI Befehlsreferenz.

`list-wireless-gateway-task-definitions`

Das folgende Codebeispiel zeigt die Verwendung `list-wireless-gateway-task-definitions`.

AWS CLI

Um die Aufgabendefinitionen für das Wireless-Gateway aufzulisten

Im folgenden `list-wireless-gateway-task-definitions` Beispiel werden die verfügbaren Aufgabendefinitionen für Drahtlos-Gateways aufgeführt, die für Ihr AWS Konto registriert sind.

```
aws iotwireless list-wireless-gateway-task-definitions
```

Ausgabe:

```
{
  "TaskDefinitions": [
    {
      "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
      "LoRaWAN" :
        {
          "CurrentVersion" :{
            "PackageVersion" : "1.0.0",
            "Station" : "2.0.5",
            "Model" : "linux"
          },
          "UpdateVersion" :{
            "PackageVersion" : "1.0.1",
            "Station" : "2.0.5",
            "Model" : "minihub"
          }
        }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListWirelessGatewayTaskDefinitions](#) in der AWS CLI Befehlsreferenz.

list-wireless-gateways

Das folgende Codebeispiel zeigt die Verwendung `list-wireless-gateways`.

AWS CLI

Um die drahtlosen Gateways aufzulisten

Im folgenden `list-wireless-gateways` Beispiel werden die verfügbaren drahtlosen Gateways in Ihrem AWS Konto aufgeführt.

```
aws iotwireless list-wireless-gateways
```

Ausgabe:

```
{
  "WirelessGatewayList": [
    {
      "Description": "My first LoRaWAN gateway",
      "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "dac632ebc01d23e4"
      },
      "Id": "3039b406-5cc9-4307-925b-9948c63da25b",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3039b406-5cc9-4307-925b-9948c63da25b",
      "Name": "myFirstLoRaWANGateway"
    },
    {
      "Description": "My second LoRaWAN gateway",
      "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "cda123ffffe92ecd2"
      },
      "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Name": "mySecondLoRaWANGateway"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [ListWirelessGateways](#) in der AWS CLI Befehlsreferenz.

send-data-to-wireless-device

Das folgende Codebeispiel zeigt die Verwendung `send-data-to-wireless-device`.

AWS CLI

Um Daten an das drahtlose Gerät zu senden

Im folgenden `send-data-to-wireless-device` Beispiel wird ein entschlüsselter Anwendungsdatenframe an das drahtlose Gerät gesendet.

```
aws iotwireless send-data-to-wireless-device \  
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \  
  --transmit-mode "1" \  
  --payload-data "SGVsbG8gVG8gRGV2c2lt" \  
  --wireless-metadata LoRaWAN={FPort=1}
```

Ausgabe:

```
{  
  MessageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [SendDataToWirelessDevice](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einen Tag-Schlüssel und einen Wert für eine Ressource anzugeben

Im folgenden `tag-resource` Beispiel wird das WLAN-Ziel `IoTWirelessDestination` mit dem Schlüssel `MyTag` und dem Wert `gekennzeichnetMyValue` gekennzeichnet.

```
aws iotwireless tag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/  
IoTWirelessDestination" \  
  --tags Key="MyTag",Value="MyValue"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Beschreiben Sie Ihren AWS IoT Core für LoRa WAN Ressourcen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

test-wireless-device

Das folgende Codebeispiel zeigt die Verwendung `test-wireless-device`.

AWS CLI

Um das drahtlose Gerät zu testen

Im folgenden `test-wireless-device` Beispiel werden Uplink-Daten von Hello an ein Gerät mit der angegebenen ID gesendet.

```
aws iotwireless test-wireless-device \  
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

Ausgabe:

```
{  
  Result: "Test succeeded. one message is sent with payload: hello"  
}
```

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [TestWirelessDevice](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein oder mehrere Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden das Tag `MyTag` und sein Wert aus dem drahtlosen Ziel entfernt `IoTWirelessDestination`.

```
aws iotwireless untag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoWirelessDestination" \  
  --tag-keys "MyTag"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Beschreiben Sie Ihren AWS IoT Core für LoRa WAN Ressourcen](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-destination

Das folgende Codebeispiel zeigt die Verwendung `update-destination`.

AWS CLI

Um die Eigenschaften eines Ziels zu aktualisieren

Im folgenden `update-destination` Beispiel wird die Eigenschaft `description` eines drahtlosen Ziels aktualisiert.

```
aws iotwireless update-destination \  
  --name "IoWirelessDestination" \  
  --description "Destination for messages processed using IoWirelessRule"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen von Zielen zu AWS IoT Core LoRa WAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateDestination](#) in der AWS CLI Befehlsreferenz.

update-partner-account

Das folgende Codebeispiel zeigt die Verwendung `update-partner-account`.

AWS CLI

Um die Eigenschaften eines Partnerkontos zu aktualisieren

Im Folgenden wird das `AppServerPrivateKey` für das Konto `update-partner-account` aktualisiert, das die angegebene ID hat.

```
aws iotwireless update-partner-account \  
  --partner-account-id "78965678771228" \  
  --partner-type "Sidewalk" \  
  --sidewalk  
  AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Amazon Sidewalk Integration for AWS IoT Core](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdatePartnerAccount](#) in der AWS CLI Befehlsreferenz.

update-wireless-device

Das folgende Codebeispiel zeigt die Verwendung `update-wireless-device`.

AWS CLI

Um die Eigenschaften eines drahtlosen Geräts zu aktualisieren

Im folgenden `update-wireless-device` Beispiel werden die Eigenschaften eines drahtlosen Geräts aktualisiert, das in Ihrem AWS Konto registriert ist.

```
aws iotwireless update-wireless-device \  
  --id "1ffd32c8-8130-4194-96df-622f072a315f" \  
  --destination-name IoTWirelessDestination2 \  
  --description "Using my first LoRaWAN device"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateWirelessDevice](#) in der AWS CLI Befehlsreferenz.

update-wireless-gateway

Das folgende Codebeispiel zeigt die Verwendung `update-wireless-gateway`.

AWS CLI

Um das Wireless-Gateway zu aktualisieren

Im folgenden `update-wireless-gateway` Beispiel wird die Beschreibung Ihres drahtlosen Gateways aktualisiert.

```
aws iotwireless update-wireless-gateway \  
  --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \  
  --description "Using my LoRaWAN gateway"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Geräte und Gateways mit AWS IoT Core verbinden LoRaWAN](#) im AWS IoT Developers Guide.

- API-Einheiten finden Sie [UpdateWirelessGateway](#) in der AWS CLI Befehlsreferenz.

IVS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden IVS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-get-channel

Das folgende Codebeispiel zeigt die Verwendung `batch-get-channel`.

AWS CLI

Um Informationen zur Kanalkonfiguration über mehrere Kanäle abzurufen

Das folgende `batch-get-channel` Beispiel listet Informationen zu den angegebenen Kanälen auf.

```
aws ivs batch-get-channel \
  --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
  arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl
```

Ausgabe:

```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "authorized": false,
      "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
      "insecureIngest": false,
      "latencyMode": "LOW",
      "name": "channel-1",
      "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel-1.abcdEFGH.m3u8",
      "preset": "",
      "playbackRestrictionPolicyArn": "",
      "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
      "srt": {
        "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
        "passphrase":
"AB1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
      },
      "tags": {},
      "type": "STANDARD"
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
      "authorized": false,
      "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
      "insecureIngest": true,
      "latencyMode": "LOW",
      "name": "channel-2",
```

```

        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/
api/video/v1/us-west-2.123456789012.channel-2.abcdEFGH.m3u8",
        "preset": "",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABCdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
"BA1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWXYZAbCDEfghIJ"
        },
        "tags": {},
        "type": "STANDARD"
    }
]
}

```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetChannel](#) in der AWS CLI Befehlsreferenz.

batch-get-stream-key

Das folgende Codebeispiel zeigt die Verwendung `batch-get-stream-key`.

AWS CLI

Um Informationen über mehrere Stream-Schlüssel zu erhalten

Im folgenden `batch-get-stream-key` Beispiel werden Informationen zu den angegebenen Stream-Schlüsseln abgerufen.

```

aws ivs batch-get-stream-key \
  --arns arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh \
  arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop

```

Ausgabe:

```

{
  "streamKeys": [
    {

```

```

    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  },
  {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop",
    "value": "sk_us-west-2_abcdABCDefgh_567890ghijkl",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  }
]
}

```

Weitere Informationen finden Sie unter [Create a Channel](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetStreamKey](#) in der AWS CLI Befehlsreferenz.

batch-start-viewer-session-revocation

Das folgende Codebeispiel zeigt die Verwendung `batch-start-viewer-session-revocation`.

AWS CLI

Um Zuschauersitzungen für mehrere Kanal- ARN und Viewer-ID-Paare zu widerrufen

Im folgenden `batch-start-viewer-session-revocation` Beispiel wird der Sitzungswiderruf für mehrere Kanal- ARN und Viewer-ID-Paare gleichzeitig ausgeführt. Die Anfrage kann normal abgeschlossen werden, gibt aber Werte im Fehlerfeld zurück, wenn der Aufrufer nicht berechtigt ist, die angegebene Sitzung zu widerrufen.

```

aws ivs batch-start-viewer-session-revocation \
  --viewer-sessions '[{"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1","viewerId":"abcdefg1","viewerSessionVersionsLessThanOrEqualTo":1234567890},
\
  {"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2","viewerId":"abcdefg2","viewerSessionVersionsLessThanOrEqualTo":1234567890}]'

```

Ausgabe:

```
{
```

```

    "errors": [
      {
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1",
        "viewerId": "abcdefg1",
        "code": "403",
        "message": "not authorized",
      },
      {
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2",
        "viewerId": "abcdefg2",
        "code": "403",
        "message": "not authorized",
      }
    ]
  }

```

Weitere Informationen finden Sie unter [Einrichten privater Kanäle](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchStartViewerSessionRevocation](#) in der AWS CLI Befehlsreferenz.

create-channel

Das folgende Codebeispiel zeigt die Verwendung `create-channel`.

AWS CLI

Beispiel 1: Um einen Kanal ohne Aufnahme zu erstellen

Das folgende `create-channel` Beispiel erstellt einen neuen Kanal und einen zugehörigen Stream-Key, um das Streaming zu starten.

```

aws ivs create-channel \
  --name "test-channel" \
  --no-insecure-ingest

```

Ausgabe:

```
{
```

```

"channel": {
  "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
  "authorized": false,
  "name": "test-channel",
  "latencyMode": "LOW",
  "playbackRestrictionPolicyArn": "",
  "recordingConfigurationArn": "",
  "srt": {
    "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
    "passphrase":
"AB1C2defGHijkLMNo3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
  },
  "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
  "insecureIngest": false,
  "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
  "preset": "",
  "tags": {},
  "type": "STANDARD"
},
"streamKey": {
  "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6",
  "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
  "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
  "tags": {}
}
}

```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 2: Um einen Kanal mit aktivierter Aufnahme zu erstellen, verwenden Sie dabei RecordingConfiguration die in seiner ARN

Das folgende create-channel Beispiel erstellt einen neuen Kanal und einen zugehörigen Stream-Key, um das Streaming zu starten, und richtet die Aufnahme für den Kanal ein.

```

aws ivs create-channel \
  --name test-channel-with-recording \
  --insecure-ingest \
  --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh"

```

Ausgabe:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "test-channel-with-recording",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"BA1C2defGHijklMN03PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
    "tags": {},
    "type": "STANDARD"
  },
  "streamKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Record to Amazon S3](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 3: Um einen Kanal mit einer Wiedergabebeschränkungsrichtlinie zu erstellen, die in seiner ARN

Das folgende `create-channel` Beispiel erstellt einen neuen Kanal und einen zugehörigen Stream-Key, um das Streaming zu starten, und richtet eine Wiedergabebeschränkungsrichtlinie für den Kanal ein.

```
aws ivs create-channel \
  --name test-channel-with-playback-restriction-policy \
  --insecure-ingest \
  --playback-restriction-policy-arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

Ausgabe:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "test-channel-with-playback-restriction-policy",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
    "recordingConfigurationArn": "",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2edfGHijkLMNo3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
    "tags": {},
    "type": "STANDARD"
  },
  "streamKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateChannel](#) in der AWS CLI Befehlsreferenz.

create-playback-restriction-policy

Das folgende Codebeispiel zeigt die Verwendung `create-playback-restriction-policy`.

AWS CLI

Um eine Richtlinie für Wiedergabebeschränkungen zu erstellen

Im folgenden `create-playback-restriction-policy` Beispiel wird eine neue Richtlinie zur Wiedergabebeschränkung erstellt.

```
aws ivs create-playback-restriction-policy \  
  --name "test-playback-restriction-policy" \  
  --enable-strict-origin-enforcement \  
  --tags "key1=value1, key2=value2" \  
  --allowed-countries US MX \  
  --allowed-origins https://www.website1.com https://www.website2.com
```

Ausgabe:

```
{  
  "playbackRestrictionPolicy": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
ABcdef34ghIJ",  
    "allowedCountries": [  
      "US",  
      "MX"  
    ],  
    "allowedOrigins": [  
      "https://www.website1.com",  
      "https://www.website2.com"  
    ],  
    "enableStrictOriginEnforcement": true,  
    "name": "test-playback-restriction-policy",  
    "tags": {  
      "key1": "value1",  
      "key2": "value2"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [CreatePlaybackRestrictionPolicy](#) in der AWS CLI Befehlsreferenz.

create-recording-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-recording-configuration`.

AWS CLI

Um eine `RecordingConfiguration` Ressource zu erstellen

Das folgende `create-recording-configuration` Beispiel erstellt eine `RecordingConfiguration` Ressource, um die Aufzeichnung auf Amazon S3 zu ermöglichen.

```
aws ivs create-recording-configuration \
  --name "test-recording-config" \
  --recording-reconnect-window-seconds 60 \
  --tags "key1=value1, key2=value2" \
  --rendition-configuration renditionSelection="CUSTOM",renditions="HD" \
  --thumbnail-configuration
recordingMode="INTERVAL",targetIntervalSeconds=1,storage="LATEST",resolution="LOWEST_RESOLUTION" \
  --destination-configuration s3={bucketName=demo-recording-bucket}
```

Ausgabe:

```
{
  "recordingConfiguration": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
    "name": "test-recording-config",
    "destinationConfiguration": {
      "s3": {
        "bucketName": "demo-recording-bucket"
      }
    },
    "state": "CREATING",
    "tags": {
      "key1": "value1",
      "key2": "value2"
    },
    "thumbnailConfiguration": {
      "recordingMode": "INTERVAL",
      "targetIntervalSeconds": 1,

```

```

        "resolution": "LOWEST_RESOLUTION",
        "storage": [
            "LATEST"
        ]
    },
    "recordingReconnectWindowSeconds": 60,
    "renditionConfiguration": {
        "renditionSelection": "CUSTOM",
        "renditions": [
            "HD"
        ]
    }
}
}

```

Weitere Informationen finden Sie unter [Auf Amazon S3 aufzeichnen](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateRecordingConfiguration](#) in der AWS CLI Befehlsreferenz.

create-stream-key

Das folgende Codebeispiel zeigt die Verwendung `create-stream-key`.

AWS CLI

Um einen Stream-Schlüssel zu erstellen

Das folgende `create-stream-key` Beispiel erstellt einen Stream-Schlüssel für eine angegebene ARN (Amazon-Ressourcenname).

```

aws ivs create-stream-key \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh

```

Ausgabe:

```

{
  "streamKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  }
}

```

```
}
```

Weitere Informationen finden Sie unter [Create a Channel](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateStreamKey](#) in der AWS CLI Befehlsreferenz.

delete-channel

Das folgende Codebeispiel zeigt die Verwendung `delete-channel`.

AWS CLI

Um einen Kanal und die zugehörigen Stream-Keys zu löschen

Das folgende `delete-channel` Beispiel löscht den Kanal mit dem angegebenen ARN (Amazon-Ressourcennamen).

```
aws ivs delete-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteChannel](#) in der AWS CLI Befehlsreferenz.

delete-playback-key-pair

Das folgende Codebeispiel zeigt die Verwendung `delete-playback-key-pair`.

AWS CLI

Um ein bestimmtes Playback-Schlüsselpaar zu löschen

Das folgende `delete-playback-key-pair` Beispiel gibt den Fingerabdruck des angegebenen key pair zurück.

```
aws ivs delete-playback-key-pair \  
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten privater Kanäle](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeletePlaybackKeyPair](#) unter AWS CLI Befehlsreferenz.

delete-playback-restriction-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-playback-restriction-policy`.

AWS CLI

Um eine Richtlinie zur Wiedergabeeinschränkung zu löschen

Im folgenden `delete-playback-restriction-policy` Beispiel wird die Wiedergabebeschränkungsrichtlinie mit der angegebenen Richtlinie ARN (Amazon Resource Name) gelöscht.

```
aws ivs delete-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABcdef34ghIJ"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im Low-Latency-Benutzerhandbuch. IVS

- API-Einheiten finden Sie [DeletePlaybackRestrictionPolicy](#) in der AWS CLI Befehlsreferenz.

delete-recording-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-recording-configuration`.

AWS CLI

Um die durch ihre angegebene RecordingConfiguration Ressource zu löschen ARN

Im folgenden `delete-recording-configuration` Beispiel wird die RecordingConfiguration Ressource mit dem angegebenen ARN Wert gelöscht.

```
aws ivs delete-recording-configuration \  
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Auf Amazon S3 aufzeichnen](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteRecordingConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-stream-key

Das folgende Codebeispiel zeigt die Verwendung `delete-stream-key`.

AWS CLI

Um einen Stream-Schlüssel zu löschen

Das folgende `delete-stream-key` Beispiel löscht den Stream-Schlüssel für eine angegebene ARN (Amazon-Ressourcenname), sodass er nicht mehr zum Streamen verwendet werden kann.

```
aws ivs delete-stream-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Create a Channel](#) im IVS Low-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteStreamKey](#) in der AWS CLI Befehlsreferenz.

get-channel

Das folgende Codebeispiel zeigt die Verwendung `get-channel`.

AWS CLI

Um die Konfigurationsinformationen eines Kanals abzurufen

Im folgenden `get-channel` Beispiel wird die Kanalkonfiguration für einen angegebenen Kanal ARN (Amazon Resource Name) abgerufen.

```
aws ivs get-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Ausgabe:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "channel-1",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "preset": "",
    "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": false,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [GetChannel](#) in der AWS CLI Befehlsreferenz.

get-playback-key-pair

Das folgende Codebeispiel zeigt die Verwendung `get-playback-key-pair`.

AWS CLI

Um ein bestimmtes Playback-Schlüsselpaar abzurufen

Das folgende `get-playback-key-pair` Beispiel gibt den Fingerabdruck des angegebenen key pair zurück.

```
aws ivs get-playback-key-pair \
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

Ausgabe:

```
{
  "keyPair": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
    "name": "my-playback-key",
    "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Einrichten privater Kanäle](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetPlaybackKeyPair](#) unter AWS CLI Befehlsreferenz.

get-playback-restriction-policy

Das folgende Codebeispiel zeigt die Verwendung `get-playback-restriction-policy`.

AWS CLI

Um die Konfigurationsinformationen einer Richtlinie zur Wiedergabeeinschränkung abzurufen

Im folgenden `get-playback-restriction-policy` Beispiel wird die Konfiguration der Wiedergabebeschränkungsrichtlinie mit der angegebenen Richtlinie ARN (Amazon Resource Name) abgerufen.

```
aws ivs get-playback-restriction-policy \
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
  ABcdef34ghIJ"
```

Ausgabe:

```
{
  "playbackRestrictionPolicy": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
  ABcdef34ghIJ",
    "allowedCountries": [
      "US",
      "MX"
    ],
  },
}
```



```

    "allowedOrigins": [
      "https://www.website1.com",
      "https://www.website2.com"
    ],
    "enableStrictOriginEnforcement": true,
    "name": "test-playback-restriction-policy",
    "tags": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}

```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [GetPlaybackRestrictionPolicy](#) in der AWS CLI Befehlsreferenz.

get-recording-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-recording-configuration`.

AWS CLI

Um Informationen über eine `RecordingConfiguration` Ressource zu erhalten

Im folgenden `get-recording-configuration` Beispiel werden Informationen über die `RecordingConfiguration` Ressource für die angegebene Ressource abgerufen ARN.

```

aws ivs get-recording-configuration \
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCdef34ghIJ"

```

Ausgabe:

```

{
  "recordingConfiguration": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCdef34ghIJ",
    "destinationConfiguration": {
      "s3": {
        "bucketName": "demo-recording-bucket"
      }
    }
  },
}

```

```

    "name": "test-recording-config",
    "recordingReconnectWindowSeconds": 60,
    "state": "ACTIVE",
    "tags": {
      "key1" : "value1",
      "key2" : "value2"
    },
    "thumbnailConfiguration": {
      "recordingMode": "INTERVAL",
      "targetIntervalSeconds": 1,
      "resolution": "LOWEST_RESOLUTION",
      "storage": [
        "LATEST"
      ]
    },
    "renditionConfiguration": {
      "renditionSelection": "CUSTOM",
      "renditions": [
        "HD"
      ]
    }
  }
}

```

Weitere Informationen finden Sie unter [Auf Amazon S3 aufzeichnen](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetRecordingConfiguration](#) in der AWS CLI Befehlsreferenz.

get-stream-key

Das folgende Codebeispiel zeigt die Verwendung `get-stream-key`.

AWS CLI

Um Informationen über einen Stream zu erhalten

Im folgenden `get-stream-key` Beispiel werden Informationen über den angegebenen Stream-Schlüssel abgerufen.

```

aws ivs get-stream-key \
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-
west-2

```

Ausgabe:

```
{
  "streamKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Create a Channel](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStreamKey](#) in der AWS CLI Befehlsreferenz.

get-stream-session

Das folgende Codebeispiel zeigt die Verwendung `get-stream-session`.

AWS CLI

Um Metadaten für einen bestimmten Stream abzurufen

Im folgenden `get-stream-session` Beispiel wird die Metadatenkonfiguration für den angegebenen Kanal ARN (Amazon-Ressourcenname) und den angegebenen Stream abgerufen. Wenn nicht angegeben, `streamId` wird der neueste Stream für den Kanal ausgewählt.

```
aws ivs get-stream-session \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
  --stream-id "mystream"
```

Ausgabe:

```
{
  "streamSession": {
    "streamId": "mystream1",
    "startTime": "2023-06-26T19:09:28+00:00",
    "channel": {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "name": "mychannel",
      "latencyMode": "LOW",
      "type": "STANDARD",
    }
  }
}
```

```
    "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABcdef34ghIJ",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "playbackUrl": "url-string",
    "authorized": false,
    "insecureIngest": false,
    "preset": ""
  },
  "ingestConfiguration": {
    "video": {
      "avcProfile": "Baseline",
      "avcLevel": "4.2",
      "codec": "avc1.42C02A",
      "encoder": "Lavf58.45.100",
      "targetBitrate": 8789062,
      "targetFramerate": 60,
      "videoHeight": 1080,
      "videoWidth": 1920
    },
    "audio": {
      "codec": "mp4a.40.2",
      "targetBitrate": 46875,
      "sampleRate": 8000,
      "channels": 2
    }
  },
  "recordingConfiguration": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
    "name": "test-recording-config",
    "destinationConfiguration": {
      "s3": {
        "bucketName": "demo-recording-bucket"
      }
    },
    "state": "ACTIVE",
    "tags": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```

```
    },
    "thumbnailConfiguration": {
      "recordingMode": "INTERVAL",
      "targetIntervalSeconds": 1,
      "resolution": "LOWEST_RESOLUTION",
      "storage": [
        "LATEST"
      ]
    },
    "recordingReconnectWindowSeconds": 60,
    "renditionConfiguration": {
      "renditionSelection": "CUSTOM",
      "renditions": [
        "HD"
      ]
    }
  },
  "truncatedEvents": [
    {
      "name": "Recording Start",
      "type": "IVS Recording State Change",
      "eventTime": "2023-06-26T19:09:35+00:00"
    },
    {
      "name": "Stream Start",
      "type": "IVS Stream State Change",
      "eventTime": "2023-06-26T19:09:34+00:00"
    },
    {
      "name": "Session Created",
      "type": "IVS Stream State Change",
      "eventTime": "2023-06-26T19:09:28+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStreamSession](#) in der AWS CLI Befehlsreferenz.

get-stream

Das folgende Codebeispiel zeigt die Verwendung `get-stream`.

AWS CLI

Um Informationen über einen Stream zu erhalten

Im folgenden `get-stream` Beispiel werden Informationen über den Stream für den angegebenen Kanal abgerufen.

```
aws ivs get-stream \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Ausgabe:

```
{  
  "stream": {  
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "startTime": "2020-05-05T21:55:38Z",  
    "state": "LIVE",  
    "health": "HEALTHY",  
    "streamId": "st-ABCDefghij01234KLMN5678",  
    "viewerCount": 1  
  }  
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStream](#) in der AWS CLI Befehlsreferenz.

import-playback-key-pair

Das folgende Codebeispiel zeigt die Verwendung `import-playback-key-pair`.

AWS CLI

Um den öffentlichen Teil eines neuen key pair zu importieren

Das folgende `import-playback-key-pair` Beispiel importiert den angegebenen öffentlichen Schlüssel (angegeben als Zeichenfolge im PEM Format) und gibt den ARN und den Fingerprint des neuen key pair zurück.

```
aws ivs import-playback-key-pair \  
  --name "my-playback-key" \  
  --public-key-material "G1lbnQx0TA3BgNVBAMMFdoeSBhcmUgew91IGR1..."
```

Ausgabe:

```
{  
  "keyPair": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",  
    "name": "my-playback-key",  
    "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Einrichten privater Kanäle](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ImportPlaybackKeyPair](#) unter AWS CLI Befehlsreferenz.

list-channels

Das folgende Codebeispiel zeigt die Verwendung `list-channels`.

AWS CLI

Beispiel 1: Um zusammenfassende Informationen über alle Kanäle zu erhalten

Das folgende `list-channels` Beispiel listet alle Kanäle für Ihr AWS Konto auf.

```
aws ivs list-channels
```

Ausgabe:

```
{  
  "channels": [  

```

```

    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "name": "channel-1",
      "latencyMode": "LOW",
      "authorized": false,
      "insecureIngest": false,
      "preset": "",
      "playbackRestrictionPolicyArn": "",
      "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
      "tags": {},
      "type": "STANDARD"
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
      "name": "channel-2",
      "latencyMode": "LOW",
      "authorized": false,
      "preset": "",
      "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABCdef34ghIJ",
      "recordingConfigurationArn": "",
      "tags": {},
      "type": "STANDARD"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 2: Um zusammenfassende Informationen über alle Kanäle abzurufen, gefiltert nach den angegebenen RecordingConfiguration ARN

Das folgende `list-channels` Beispiel listet alle Kanäle für Ihr AWS Konto auf, die mit den angegebenen verknüpft sind RecordingConfiguration ARN.

```

aws ivs list-channels \
  --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"

```

Ausgabe:


```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "name": "channel-1",
      "latencyMode": "LOW",
      "authorized": false,
      "insecureIngest": false,
      "preset": "",
      "playbackRestrictionPolicyArn": "",
      "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
      "tags": {},
      "type": "STANDARD"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Record to Amazon S3](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 3: Um zusammenfassende Informationen über alle Kanäle abzurufen, gefiltert nach den angegebenen PlaybackRestrictionPolicy ARN

Das folgende `list-channels` Beispiel listet alle Kanäle für Ihr AWS Konto auf, die mit den angegebenen verknüpft sind PlaybackRestrictionPolicy ARN.

```
aws ivs list-channels \
  --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

Ausgabe:

```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
      "name": "channel-2",
      "latencyMode": "LOW",
      "authorized": false,
      "preset": "",

```

```
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "tags": {},
        "type": "STANDARD"
    }
]
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [ListChannels](#) in der AWS CLI Befehlsreferenz.

list-playback-key-pairs

Das folgende Codebeispiel zeigt die Verwendung `list-playback-key-pairs`.

AWS CLI

Um zusammenfassende Informationen zu allen Playback-Tastenpaaren zu erhalten

Das folgende `list-playback-key-pairs` Beispiel gibt Informationen über alle Schlüsselpaare zurück.

```
aws ivs list-playback-key-pairs
```

Ausgabe:

```
{
  "keyPairs": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
      "name": "test-key-0",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/ijkl15678mnop",
      "name": "test-key-1",
      "tags": {}
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Einrichten privater Kanäle](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPlaybackKeyPairs](#) unter AWS CLI Befehlsreferenz.

list-playback-restriction-policies

Das folgende Codebeispiel zeigt die Verwendung `list-playback-restriction-policies`.

AWS CLI

Um zusammenfassende Informationen zu allen Richtlinien für Wiedergabebeschränkungen zu erhalten

Das folgende `list-playback-restriction-policies` Beispiel listet alle Richtlinien für Wiedergabebeschränkungen für Ihr AWS Konto auf.

```
aws ivs list-playback-restriction-policies
```

Ausgabe:

```
{
  "playbackRestrictionPolicies": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ",
      "allowedCountries": [
        "US",
        "MX"
      ],
      "allowedOrigins": [
        "https://www.website1.com",
        "https://www.website2.com"
      ],
      "enableStrictOriginEnforcement": true,
      "name": "test-playback-restriction-policy",
      "tags": {
        "key1": "value1",
        "key2": "value2"
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPlaybackRestrictionPolicies](#) in der AWS CLI Befehlsreferenz.

list-recording-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-recording-configurations`.

AWS CLI

Um alle in diesem Konto erstellten RecordingConfiguration-Ressourcen aufzulisten

Im folgenden `list-recording-configurations`-Beispiel werden Informationen zu allen RecordingConfiguration-Ressourcen in Ihrem Konto abgerufen.

```
aws ivs list-recording-configurations
```

Ausgabe:

```
{  
  "recordingConfigurations": [  
    {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/  
ABCdef34ghIJ",  
      "name": "test-recording-config-1",  
      "destinationConfiguration": {  
        "s3": {  
          "bucketName": "demo-recording-bucket-1"  
        }  
      },  
      "state": "ACTIVE",  
      "tags": {}  
    },  
    {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/  
CD12abcdGHIJ",  
      "name": "test-recording-config-2",
```

```

        "destinationConfiguration": {
            "s3": {
                "bucketName": "demo-recording-bucket-2"
            }
        },
        "state": "ACTIVE",
        "tags": {}
    }
]
}

```

Weitere Informationen finden Sie unter [Auf Amazon S3 aufzeichnen](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRecordingConfigurations](#) in der AWS CLI Befehlsreferenz.

list-stream-keys

Das folgende Codebeispiel zeigt die Verwendung `list-stream-keys`.

AWS CLI

Um eine Liste von Stream-Schlüsseln zu erhalten

Das folgende `list-stream-keys` Beispiel listet alle Stream-Schlüssel für eine angegebene ARN (Amazon-Ressourcenname) auf.

```

aws ivs list-stream-keys \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh

```

Ausgabe:

```

{
  "streamKeys": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "tags": {}
    }
  ]
}

```

Für weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVS Low-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStreamKeys](#) in der AWS CLI Befehlsreferenz.

list-stream-sessions

Das folgende Codebeispiel zeigt die Verwendung `list-stream-sessions`.

AWS CLI

Um eine Zusammenfassung der aktuellen und vorherigen Streams für einen bestimmten Kanal in der aktuellen AWS Region zu erhalten

Das folgende `list-stream-sessions` Beispiel meldet zusammenfassende Informationen für Streams für einen bestimmten Kanal ARN (Amazon-Ressourcenname).

```
aws ivs list-stream-sessions \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
  --max-results 25 \
  --next-token ""
```

Ausgabe:

```
{
  "nextToken": "set-2",
  "streamSessions": [
    {
      "startTime": 1641578182,
      "endTime": 1641579982,
      "hasErrorEvent": false,
      "streamId": "mystream"
    }
    ...
  ]
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVS Low-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStreamSessions](#) in der AWS CLI Befehlsreferenz.

list-streams

Das folgende Codebeispiel zeigt die Verwendung `list-streams`.

AWS CLI

Um eine Liste der Live-Streams und deren Status zu erhalten

Das folgende `list-streams` Beispiel listet alle Live-Streams für Ihr AWS Konto auf.

```
aws ivs list-streams
```

Ausgabe:

```
{
  "streams": [
    {
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "state": "LIVE",
      "health": "HEALTHY",
      "streamId": "st-ABCDEFghij01234KLMN5678",
      "viewerCount": 1
    }
  ]
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStreams](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um alle Tags für eine AWS Ressource aufzulisten (zum Beispiel: Channel, Stream Key)

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für eine angegebene Ressource ARN (Amazon-Ressourcenname) auf.

```
aws ivs list-tags-for-resource \
```

```
--resource-arn arn:aws:ivs:us-west-2:12345689012:channel/abcdABCDefgh
```

Ausgabe:

```
{
  "tags":
  {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Weitere Informationen finden Sie unter [Tagging](#) in der Amazon Interactive Video Service API Reference.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-metadata

Das folgende Codebeispiel zeigt die Verwendung `put-metadata`.

AWS CLI

Um Metadaten für einen bestimmten Kanal in den aktiven Stream einzufügen

Das folgende `put-metadata` Beispiel fügt die angegebenen Metadaten in den Stream für den angegebenen Kanal ein.

```
aws ivs put-metadata \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
  --metadata '{"my": "metadata"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVS Low-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [PutMetadata](#) in der AWS CLI Befehlsreferenz.

start-viewer-session-revocation

Das folgende Codebeispiel zeigt die Verwendung `start-viewer-session-revocation`.

AWS CLI

Um eine Zuschauersitzung für ein bestimmtes Paar aus mehreren Kanälen ARN und Zuschauer-IDs zu widerrufen

Das folgende `start-viewer-session-revocation` Beispiel startet den Vorgang des Widerrufs der Zuschauersitzung, die einem bestimmten Kanal ARN und einer bestimmten Zuschauer-ID zugeordnet ist, bis einschließlich der angegebenen Sitzungsversionsnummer. Wenn die Version nicht angegeben wird, ist sie standardmäßig 0.

```
aws ivs batch-start-viewer-session-revocation \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --viewer-id abcdefg \  
  --viewer-session-versions-less-than-or-equal-to 1234567890
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten privater Kanäle](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [StartViewerSessionRevocation](#) in der AWS CLI Befehlsreferenz.

stop-stream

Das folgende Codebeispiel zeigt die Verwendung `stop-stream`.

AWS CLI

Um einen bestimmten Stream zu stoppen

Im folgenden `stop-stream` Beispiel wird der Stream auf dem angegebenen Kanal gestoppt.

```
aws ivs stop-stream \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVS Low-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [StopStream](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags für eine AWS Ressource hinzuzufügen oder zu aktualisieren (zum Beispiel: Channel, Stream Key)

Im folgenden `tag-resource` Beispiel werden Tags für eine angegebene Ressource ARN (Amazon Resource Name) hinzugefügt oder aktualisiert.

```
aws ivs tag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) in der Amazon Interactive Video Service API Reference.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags für eine AWS Ressource zu entfernen (zum Beispiel: Channel, Stream Key)

Das folgende `untag-resource` Beispiel entfernt die angegebenen Tags für eine angegebene Ressource ARN (Amazon Resource Name).

```
aws ivs untag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tag-keys "tagkey1, tagkey2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) in der Amazon Interactive Video Service API Reference.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-channel

Das folgende Codebeispiel zeigt die Verwendung `update-channel`.

AWS CLI

Beispiel 1: Um die Konfigurationsinformationen eines Kanals zu aktualisieren

Im folgenden `update-channel` Beispiel wird die Kanalkonfiguration für einen angegebenen Kanal aktualisiert, um den Kanalnamen zu ändern. Dies wirkt sich nicht auf einen laufenden Stream dieses Kanals aus. Sie müssen den Stream beenden und neu starten, damit die Änderungen wirksam werden.

```
aws ivs update-channel \
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
  --name "channel-1" \
  --insecure-ingest
```

Ausgabe:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "channel-1",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "recordingConfigurationArn": "",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijkLMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
```

```
"tags": {}
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 2: Um die Konfiguration eines Kanals zu aktualisieren, um die Aufnahme zu ermöglichen

Im folgenden `update-channel` Beispiel wird die Kanalkonfiguration für einen bestimmten Kanal aktualisiert, um die Aufnahme zu ermöglichen. Dies wirkt sich nicht auf einen laufenden Stream dieses Kanals aus. Sie müssen den Stream beenden und neu starten, damit die Änderungen wirksam werden.

```
aws ivs update-channel \
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
  --no-insecure-ingest \
  --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh"
```

Ausgabe:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "test-channel-with-recording",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"BA1C2defGHijklMN03PqQRstUvwxyzABCDEFGHIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": false,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
    "tags": {}
  }
}
```

```
}
}
```

Weitere Informationen finden Sie unter [Record to Amazon S3](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 3: Um die Konfiguration eines Kanals zu aktualisieren, um die Aufnahme zu deaktivieren

Im folgenden `update-channel` Beispiel wird die Kanalkonfiguration für einen bestimmten Kanal aktualisiert, um die Aufnahme zu deaktivieren. Dies wirkt sich nicht auf einen laufenden Stream dieses Kanals aus. Sie müssen den Stream beenden und neu starten, damit die Änderungen wirksam werden.

```
aws ivs update-channel \
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
  --recording-configuration-arn ""
```

Ausgabe:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "test-channel-with-recording",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "recordingConfigurationArn": "",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2edfGHijkLMNo3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDefghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": false,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Record to Amazon S3](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 4: Um die Konfiguration eines Kanals zu aktualisieren, um die Wiedergabebeschränkung zu aktivieren

Im folgenden `update-channel` Beispiel wird die Kanalkonfiguration für einen bestimmten Kanal aktualisiert, um eine Wiedergabebeschränkungsrichtlinie anzuwenden. Dies wirkt sich nicht auf einen laufenden Stream dieses Kanals aus. Sie müssen den Stream beenden und neu starten, damit die Änderungen wirksam werden.

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --no-insecure-ingest \  
  --playback-restriction-policy-arn "arn:aws:ivs:us-west-2:123456789012:playback-  
restriction-policy/ABcdef34ghIJ"
```

Ausgabe:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "test-channel-with-playback-restriction-policy",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "arn:aws:ivs:us-  
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijk1MN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

Beispiel 5: So aktualisieren Sie die Konfiguration eines Kanals, um die Wiedergabebeschränkung zu deaktivieren

Im folgenden `update-channel` Beispiel wird die Kanalkonfiguration für einen bestimmten Kanal aktualisiert, ARN um die Wiedergabebeschränkung zu deaktivieren. Dies wirkt sich nicht auf einen laufenden Stream dieses Kanals aus. Sie müssen den Stream beenden und neu starten, damit die Änderungen wirksam werden.

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --playback-restriction-policy-arn ""
```

Ausgabe:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "test-channel-with-playback-restriction-policy",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDeFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateChannel](#) in der AWS CLI Befehlsreferenz.

update-playback-restriction-policy

Das folgende Codebeispiel zeigt die Verwendung `update-playback-restriction-policy`.

AWS CLI

Um eine Richtlinie zur Wiedergabebeschränkung zu aktualisieren

Im folgenden `update-playback-restriction-policy` Beispiel wird die Richtlinie für Wiedergabebeschränkungen mit der angegebenen Richtlinie aktualisiert, um die strikte Durchsetzung der Herkunft zu deaktivieren. Dies wirkt sich nicht auf einen laufenden Stream des zugehörigen Kanals aus. Sie müssen den Stream beenden und neu starten, damit die Änderungen wirksam werden.

```
aws ivs update-playback-restriction-policy \
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
  ABcdef34ghIJ" \
  --no-enable-strict-origin-enforcement
```

Ausgabe:

```
{
  "playbackRestrictionPolicy": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
  ABcdef34ghIJ",
    "allowedCountries": [
      "US",
      "MX"
    ],
    "allowedOrigins": [
      "https://www.website1.com",
      "https://www.website2.com"
    ],
    "enableStrictOriginEnforcement": false,
    "name": "test-playback-restriction-policy",
    "tags": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```



```
}
```

Weitere Informationen finden Sie unter [Unerwünschte Inhalte und Zuschauer](#) im IVSLow-Latency-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdatePlaybackRestrictionPolicy](#) in der AWS CLI Befehlsreferenz.

Amazon IVS Chat-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon IVS Chat Aktionen ausführen und gängige Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-chat-token

Das folgende Codebeispiel zeigt die Verwendung `create-chat-token`.

AWS CLI

Um ein Chat-Token zu erstellen

Im folgenden `create-chat-token` Beispiel wird ein verschlüsseltes Chat-Token erstellt, das verwendet wird, um eine individuelle WebSocket-Verbindung zu einem Raum herzustellen. Das Token ist für eine Minute gültig, und eine mit dem Token hergestellte Verbindung (Sitzung) ist für die angegebene Dauer gültig.

```
aws ivschat create-chat-token \
  --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \
```

```
--userId "11231234" \  
--capabilities "SEND_MESSAGE", \  
--sessionDurationInMinutes 30
```

Ausgabe:

```
{  
  "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",  
  "sessionExpirationTime": "2022-03-16T04:44:09+00:00"  
  "state": "CREATING",  
  "tokenExpirationTime": "2022-03-16T03:45:09+00:00"  
}
```

Weitere Informationen finden Sie unter [Schritt 3: Chat-Clients authentifizieren und autorisieren](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateChatToken](#) in der AWS CLI Befehlsreferenz.

create-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-logging-configuration`.

AWS CLI

Um eine `LoggingConfiguration` Chat-Ressource zu erstellen

Im folgenden `create-logging-configuration` Beispiel wird eine `LoggingConfiguration` Ressource erstellt, die es Clients ermöglicht, gesendete Nachrichten zu speichern und aufzuzeichnen.

```
aws ivschat create-logging-configuration \  
  --destination-configuration s3={bucketName=demo-logging-bucket} \  
  --name "test-logging-config" \  
  --tags "key1=value1, key2=value2"
```

Ausgabe:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABCdef34ghIJ",  
  "createTime": "2022-09-14T17:48:00.653000+00:00",
```

```
"destinationConfiguration": {
  "s3": {
    "bucketName": "demo-logging-bucket"
  }
},
"id": "ABcdef34ghIJ",
"name": "test-logging-config",
"state": "ACTIVE",
"tags": { "key1" : "value1", "key2" : "value2" },
"updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

create-room

Das folgende Codebeispiel zeigt die Verwendung `create-room`.

AWS CLI

Um einen Raum zu erstellen

Im folgenden `create-room` Beispiel wird ein neuer Raum erstellt.

```
aws ivschat create-room \
  --name "test-room-1" \
  --logging-configuration-identifiers "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
  --maximum-message-length 256 \
  --maximum-message-rate-per-second 5
```

Ausgabe:

```
{
  "arn": "arn:aws:ivschat:us-west-2:123456789012:room/g1H2I3j4k5L6",
  "id": "g1H2I3j4k5L6",
  "createTime": "2022-03-16T04:44:09+00:00",
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
```

```
"maximumMessageLength": 256,  
"maximumMessageRatePerSecond": 5,  
"name": "test-room-1",  
"tags": {}  
"updateTime": "2022-03-16T07:22:09+00:00"  
}
```

Weitere Informationen finden Sie unter [Schritt 2: Einen Chatroom erstellen](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateRoom](#) in der AWS CLI Befehlsreferenz.

delete-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-logging-configuration`.

AWS CLI

Um eine `LoggingConfiguration` Chat-Ressource zu löschen

Im folgenden `delete-logging-configuration` Beispiel wird die `LoggingConfiguration` Ressource für die angegebene ARN-Ressource gelöscht.

```
aws ivschat delete-logging-configuration \  
  --identifizier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABcdef34ghIJ"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

delete-message

Das folgende Codebeispiel zeigt die Verwendung `delete-message`.

AWS CLI

Um Nachrichten aus einem bestimmten Raum zu löschen

Im folgenden `delete-message` Beispiel wird ein Ereignis an den angegebenen Raum gesendet, wodurch die Clients angewiesen werden, die angegebene Nachricht zu löschen, d. h. sie aus der Ansicht rückgängig zu machen und sie aus dem Chat-Verlauf des Clients zu löschen.

```
aws ivschat delete-message \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --id "ABC123def456" \  
  --reason "Message contains profanity"
```

Ausgabe:

```
{  
  "id": "12345689012"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteMessage](#) unter AWS CLI Befehlsreferenz.

delete-room

Das folgende Codebeispiel zeigt die Verwendung `delete-room`.

AWS CLI

Um einen Raum zu löschen

Im folgenden `delete-room` Beispiel wird der angegebene Raum gelöscht. Verbundene Clients sind getrennt. Bei Erfolg gibt es HTTP 204 mit einem leeren Antworttext zurück.

```
aws ivschat delete-room \  
  --identifizier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteRoom](#) unter AWS CLI Befehlsreferenz.

disconnect-user

Das folgende Codebeispiel zeigt die Verwendung `disconnect-user`.

AWS CLI

Um einen Benutzer von einem Raum zu trennen

Im folgenden `disconnect-user` Beispiel werden alle Verbindungen für den angegebenen Benutzer mit dem angegebenen Raum getrennt. Bei Erfolg wird HTTP 200 mit einem leeren Antworttext zurückgegeben.

```
aws ivschat disconnect-user \  
  --roomId "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --userId "ABC123def456" \  
  --reason "Violated terms of service"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DisconnectUser](#) unter AWS CLI Befehlsreferenz.

get-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-logging-configuration`.

AWS CLI

Um Informationen über eine `LoggingConfiguration` Ressource zu erhalten

Im folgenden `get-logging-configuration` Beispiel werden Informationen über die `LoggingConfiguration` Ressource für die angegebene Ressource abgerufen ARN.

```
aws ivschat get-logging-configuration \  
  --id "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABCdef34ghIJ"
```

Ausgabe:

```
{
```

```

    "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
    ABcdef34ghIJ",
    "createTime": "2022-09-14T17:48:00.653000+00:00",
    "destinationConfiguration": {
      "s3": {
        "bucketName": "demo-logging-bucket"
      }
    },
    "id": "ABcdef34ghIJ",
    "name": "test-logging-config",
    "state": "ACTIVE",
    "tags": { "key1" : "value1", "key2" : "value2" },
    "updateTime": "2022-09-14T17:48:01.104000+00:00"
  }

```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

get-room

Das folgende Codebeispiel zeigt die Verwendung get-room.

AWS CLI

Um das angegebene Zimmer zu erhalten

Im folgenden get-room Beispiel werden Informationen über den angegebenen Raum abgerufen.

```

aws ivschat get-room \
  --identifizier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"

```

Ausgabe:

```

{
  "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
  "createTime": "2022-03-16T04:44:09+00:00",
  "id": "g1H2I3j4k5L6",
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
  west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
  "maximumMessageLength": 256,

```

```
"maximumMessageRatePerSecond": 5,  
"name": "test-room-1",  
"tags": {},  
"updateTime": "2022-03-16T07:22:09+00:00"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetRoom](#) unter AWS CLI Befehlsreferenz.

list-logging-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-logging-configurations`.

AWS CLI

Um zusammenfassende Informationen über alle Protokollierungskonfigurationen für den Benutzer in der AWS Region zu erhalten, in der die API-Anfrage verarbeitet wird

Im folgenden `list-logging-configurations` Beispiel werden Informationen zu allen `LoggingConfiguration` Ressourcen für den Benutzer in der AWS Region aufgeführt, in der die API-Anfrage verarbeitet wird.

```
aws ivschat list-logging-configurations \  
--max-results 2 \  
--next-token ""
```

Ausgabe:

```
{  
  "nextToken": "set-2",  
  "loggingConfigurations": [  
    {  
      "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABcdef34ghIJ",  
      "createTime": "2022-09-14T17:48:00.653000+00:00",  
      "destinationConfiguration": {  
        "s3": {  
          "bucketName": "demo-logging-bucket"  
        }  
      },  
    },  
  ],  
}
```



```

        "id": "ABCdef34ghIJ",
        "name": "test-logging-config",
        "state": "ACTIVE",
        "tags": { "key1" : "value1", "key2" : "value2" },
        "updateTime": "2022-09-14T17:48:01.104000+00:00"
    }
    ...
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListLoggingConfigurations](#) unter AWS CLI Befehlsreferenz.

list-rooms

Das folgende Codebeispiel zeigt die Verwendung `list-rooms`.

AWS CLI

Um zusammenfassende Informationen zu all Ihren Zimmern in der aktuellen Region zu erhalten

Im folgenden `list-rooms` Beispiel werden zusammenfassende Informationen zu allen Räumen in der AWS Region abgerufen, in der die Anfrage bearbeitet wird. Die Ergebnisse sind in absteigender Reihenfolge von `updateTime` sortiert.

```

aws ivschat list-rooms \
  --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABCdef34ghIJ" \
  --max-results 10 \
  --next-token ""

```

Ausgabe:

```

{
  "nextToken": "page3",
  "rooms": [
    {
      "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
      "createTime": "2022-03-16T04:44:09+00:00",
      "id": "g1H2I3j4k5L6",

```

```
        "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],  
        "name": "test-room-1",  
        "tags": {},  
        "updateTime": "2022-03-16T07:22:09+00:00"  
    }  
]  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRooms](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um alle Tags für eine AWS-Ressource aufzulisten (zum Beispiel: Raum)

Das folgende `list-tags-for-resource` Beispiel listet alle Tags für eine angegebene Ressource-ARN (Amazon-Ressourcenname) auf.

```
aws ivschat list-tags-for-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:123456789012:room/g1H2I3j4k5L6
```

Ausgabe:

```
{  
  "tags":  
  {  
    "key1": "value1",  
    "key2": "value2"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging](#) in der Amazon Interactive Video Service API Reference.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

send-event

Das folgende Codebeispiel zeigt die Verwendung send-event.

AWS CLI

Um ein Ereignis in einen Raum zu senden

Im folgenden send-event Beispiel wird das angegebene Ereignis in den angegebenen Raum gesendet.

```
aws ivschat send-event \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --eventName "SystemMessage" \  
  --attributes \  
    "msgType"="user-notification", \  
    "msgText"="This chat room will close in 15 minutes."
```

Ausgabe:

```
{  
  "id": "12345689012"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [SendEvent](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung tag-resource.

AWS CLI

Um Tags für eine AWS Ressource hinzuzufügen oder zu aktualisieren (zum Beispiel: Raum)

Im folgenden tag-resource Beispiel werden Tags für eine angegebene Ressource ARN (Amazon Resource Name) hinzugefügt oder aktualisiert. Bei Erfolg gibt es HTTP 200 mit einem leeren Antworttext zurück.

```
aws ivschat tag-resource \  
  --resourceArn "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --tags "key=value"
```

```
--resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
--tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) in der Amazon Interactive Video Service API Reference.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags für eine AWS Ressource zu entfernen (zum Beispiel: Raum)

Das folgende `untag-resource` Beispiel entfernt die angegebenen Tags für eine angegebene Ressource ARN (Amazon Resource Name). Bei Erfolg gibt es HTTP 200 mit einem leeren Antworttext zurück.

```
aws ivschat untag-resource \  
--resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
--tag-keys "tagkey1, tagkey2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging](#) in der Amazon Interactive Video Service API Reference.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-logging-configuration`.

AWS CLI

Um die Logging-Konfiguration eines Raums zu aktualisieren

Das folgende `update-logging-configuration` Beispiel aktualisiert eine `LoggingConfiguration` Ressource mit den angegebenen Daten.

```
aws ivschat update-logging-configuration \
  --destination-configuration s3={bucketName=demo-logging-bucket} \
  --identifizier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
  ABcdef34ghIJ" \
  --name "test-logging-config"
```

Ausgabe:

```
{
  "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
  ABcdef34ghIJ",
  "createTime": "2022-09-14T17:48:00.653000+00:00",
  "destinationConfiguration": {
    "s3": {
      "bucketName": "demo-logging-bucket"
    }
  },
  "id": "ABcdef34ghIJ",
  "name": "test-logging-config",
  "state": "ACTIVE",
  "tags": { "key1" : "value1", "key2" : "value2" },
  "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

update-room

Das folgende Codebeispiel zeigt die Verwendung `update-room`.

AWS CLI

Um die Konfiguration eines Raums zu aktualisieren

Das folgende `update-room` Beispiel aktualisiert die Konfiguration des angegebenen Raums mit den angegebenen Daten.

```
aws ivschat update-room \
```

```
--identifizier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
--logging-configuration-identifiers "arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABCdef34ghIJ" \  
--name "chat-room-a" \  
--maximum-message-length 256 \  
--maximum-message-rate-per-second 5
```

Ausgabe:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",  
  "createTime": "2022-03-16T04:44:09+00:00",  
  "id": "g1H2I3j4k5L6",  
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABCdef34ghIJ"],  
  "maximumMessageLength": 256,  
  "maximumMessageRatePerSecond": 5,  
  "name": "chat-room-a",  
  "tags": {},  
  "updateTime": "2022-03-16T07:22:09+00:00"  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon IVS Chat](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRoom](#) unter AWS CLI Befehlsreferenz.

Beispiele für Amazon IVS Real-Time Streaming mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon IVS Real-Time Streaming Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-encoder-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-encoder-configuration`.

AWS CLI

Um eine Composition Encoder-Konfiguration zu erstellen

Im folgenden `create-encoder-configuration` Beispiel wird eine Composition Encoder-Konfiguration mit den angegebenen Eigenschaften erstellt.

```
aws ivs-realtime create-encoder-configuration \  
  --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

Ausgabe:

```
{  
  "encoderConfiguration": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/  
ABabCDcdEFef",  
    "name": "test-ec",  
    "tags": {},  
    "video": {  
      "bitrate": 3500000,  
      "framerate": 30,  
      "height": 1080,  
      "width": 1920  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateEncoderConfiguration](#) in der AWS CLI Befehlsreferenz.

create-participant-token

Das folgende Codebeispiel zeigt die Verwendung `create-participant-token`.

AWS CLI

Um ein Etappenteilnehmer-Token zu erstellen

Im folgenden `create-participant-token` Beispiel wird ein Teilnehmer-Token für die angegebene Phase erstellt.

```
aws ivs-realtime create-participant-token \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --user-id bob
```

Ausgabe:

```
{  
  "participantToken": {  
    "expirationTime": "2023-03-07T09:47:43+00:00",  
    "participantId": "ABCDEFghij01234KLMN6789",  
    "token": "abcd1234defg5678"  
  }  
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateParticipantToken](#) in der AWS CLI Befehlsreferenz.

create-stage

Das folgende Codebeispiel zeigt die Verwendung `create-stage`.

AWS CLI

Beispiel 1: Um eine Phase zu erstellen

Im folgenden `create-stage` Beispiel wird ein Token für die Phase und den Teilnehmer der Phase für einen angegebenen Benutzer erstellt.

```
aws ivs-realtime create-stage \  
  --name stage1 \  
  --participant-token-configurations userId=alice
```

Ausgabe:


```
{
  "participantTokens": [
    {
      "participantId": "ABCDEFghij01234KLMN5678",
      "token": "a1b2c3d4567890ab",
      "userId": "alice"
    }
  ],
  "stage": {
    "activeSessionId": "st-a1b2c3d4e5f6g",
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
    "endpoints": {
      "events": "wss://global.events.live-video.net",
      "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
    },
    "name": "stage1",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

Beispiel 2: Um eine Phase zu erstellen und die Aufzeichnung einzelner Teilnehmer zu konfigurieren

Im folgenden `create-stage` Beispiel wird eine Phase erstellt und die Aufzeichnung einzelner Teilnehmer konfiguriert.

```
aws ivs-realtime create-stage \
  --name stage1 \
  --auto-participant-recording-configuration '{"mediaTypes":  
["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-  
west-2:123456789012:storage-configuration/abcdABCDefgh"}'
```

Ausgabe:

```
{
  "stage": {
    "activeSessionId": "st-a1b2c3d4e5f6g",
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
    "autoParticipantRecordingConfiguration": {
```

```

        "mediaTypes": [
            "AUDIO_VIDEO"
        ],
        "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-configuration/abcdABCDefgh",
    },
    "endpoints": {
        "events": "wss://global.events.live-video.net",
        "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
    },
    "name": "stage1",
    "tags": {}
}
}

```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateStage](#) in der AWS CLI Befehlsreferenz.

create-storage-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-storage-configuration`.

AWS CLI

Um eine Speicherkonfiguration für Kompositionen zu erstellen

Im folgenden `create-storage-configuration` Beispiel wird eine Kompositionsspeicherkonfiguration mit den angegebenen Eigenschaften erstellt.

```

aws ivs-realtime create-storage-configuration \
  --name "test-sc" --s3 "bucketName=test-bucket-name"

```

Ausgabe:

```

{
  "storageConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/ABabCDcdEFef",
    "name": "test-sc",
    "s3": {
      "bucketName": "test-bucket-name"
    }
  }
}

```

```
    },  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateStorageConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-encoder-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-encoder-configuration`.

AWS CLI

Um eine Composition Encoder-Konfiguration zu löschen

Im Folgenden wird die durch den angegebenen ARN (Amazon-Ressourcennamen) angegebene Konfiguration des Kompositions-Encoders `delete-encoder-configuration` gelöscht.

```
aws ivs-realtime delete-encoder-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/  
  ABabCDcdEFef"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteEncoderConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-public-key

Das folgende Codebeispiel zeigt die Verwendung `delete-public-key`.

AWS CLI

Um einen öffentlichen Schlüssel zu löschen

Im Folgenden `delete-public-key` wird der angegebene öffentliche Schlüssel gelöscht.

```
aws ivs-realtime delete-public-key \  
  --key-id "key-id"
```

```
--arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verteilung von Teilnehmer-Token](#) im Amazon IVS Real-Time Streaming-Benutzerhandbuch.

- API-Einheiten finden Sie [DeletePublicKey](#) in der AWS CLI Befehlsreferenz.

delete-stage

Das folgende Codebeispiel zeigt die Verwendung `delete-stage`.

AWS CLI

Um eine Phase zu löschen

Im folgenden `delete-stage` Beispiel wird die angegebene Phase gelöscht.

```
aws ivs-realtime delete-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteStage](#) in der AWS CLI Befehlsreferenz.

delete-storage-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-storage-configuration`.

AWS CLI

Um eine Speicherkonfiguration für Kompositionen zu löschen

Im Folgenden wird die durch den angegebenen ARN (Amazon-Ressourcennamen) angegebene Kompositionsspeicherkonfiguration `delete-storage-configuration` gelöscht.

```
aws ivs-realtime delete-storage-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
  ABabCDcdEFef"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteStorageConfiguration](#) in der AWS CLI Befehlsreferenz.

disconnect-participant

Das folgende Codebeispiel zeigt die Verwendung `disconnect-participant`.

AWS CLI

Um die Verbindung mit einem Teilnehmer der Phase zu trennen

Im folgenden `disconnect-participant` Beispiel wird die Verbindung zwischen dem angegebenen Teilnehmer und der angegebenen Phase getrennt.

```
aws ivs-realtime disconnect-participant \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --participant-id ABCDEFGHIJ01234KLMN5678
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [DisconnectParticipant](#) in der AWS CLI Befehlsreferenz.

get-composition

Das folgende Codebeispiel zeigt die Verwendung `get-composition`.

AWS CLI

Beispiel 1: Um eine Komposition mit Standard-Layouteinstellungen zu erhalten

Im folgenden `get-composition` Beispiel wird die Zusammensetzung für den angegebenen ARN (Amazon-Ressourcennamen) abgerufen.

```
aws ivs-realtime get-composition \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Ausgabe:

```

{
  "composition": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
    "destinations": [
      {
        "configuration": {
          "channel": {
            "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
            "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
          },
          "name": ""
        },
        "id": "AabBCcdDEefF",
        "startTime": "2023-10-16T23:26:00+00:00",
        "state": "ACTIVE"
      },
      {
        "configuration": {
          "name": "",
          "s3": {
            "encoderConfigurationArns": [
              "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
            ],
            "recordingConfiguration": {
              "format": "HLS"
            },
            "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
          }
        },
        "detail": {
          "s3": {
            "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
          }
        },
        "id": "GHFabcgefABC",
        "startTime": "2023-10-16T23:26:00+00:00",
        "state": "STARTING"
      }
    ]
  }
}

```

```

    }
  ],
  "layout": {
    "grid": {
      "featuredParticipantAttribute": ""
      "gridGap": 2,
      "omitStoppedVideo": false,
      "videoAspectRatio": "VIDEO",
      "videoFillMode": ""
    }
  },
  "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCdabcd",
  "startTime": "2023-10-16T23:24:00+00:00",
  "state": "ACTIVE",
  "tags": {}
}
}

```

Weitere Informationen finden Sie unter [Composite Recording \(Echtzeit-Streaming\)](#) im Amazon Interactive Video Service-Benutzerhandbuch.

Beispiel 2: Um eine Komposition mit PiP-Layout zu erhalten

Im folgenden `get-composition` Beispiel wird die Komposition für die angegebene ARN (Amazon-Ressourcenname) abgerufen, die das PiP-Layout verwendet.

```

aws ivs-realtime get-composition \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"

```

Ausgabe:

```

{
  "composition": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
    "destinations": [
      {
        "configuration": {
          "channel": {
            "channelArn": "arn:aws:ivs:ap-northeast-1:123456789012:channel/abcABCdefDEg",
            "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
          },
          "name": ""
        }
      }
    ]
  }
}

```

```

    },
    "id": "AabBCcdDEefF",
    "startTime": "2023-10-16T23:26:00+00:00",
    "state": "ACTIVE"
  },
  {
    "configuration": {
      "name": "",
      "s3": {
        "encoderConfigurationArns": [
          "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
        ],
        "recordingConfiguration": {
          "format": "HLS"
        },
        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
      }
    },
    "detail": {
      "s3": {
        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
      }
    },
    "id": "GHFabcgefABC",
    "startTime": "2023-10-16T23:26:00+00:00",
    "state": "STARTING"
  }
],
"layout": {
  "pip": {
    "featuredParticipantAttribute": "abcdefg",
    "gridGap": 0,
    "omitStoppedVideo": false,
    "pipBehavior": "STATIC",
    "pipOffset": 0,
    "pipParticipantAttribute": "",
    "pipPosition": "BOTTOM_RIGHT",
    "videoFillMode": "COVER"
  }
},
"stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",

```



```
    "startTime": "2023-10-16T23:24:00+00:00",
    "state": "ACTIVE",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Composite Recording \(Echtzeit-Streaming\)](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetComposition](#) in der AWS CLI Befehlsreferenz.

get-encoder-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-encoder-configuration`.

AWS CLI

Um eine Composition Encoder-Konfiguration zu erhalten

Im folgenden `get-encoder-configuration` Beispiel wird die durch den angegebenen Wert ARN (Amazon-Ressourcenname) angegebene Konfiguration des Kompositions-Encoders abgerufen.

```
aws ivs-realtime get-encoder-configuration \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

Ausgabe:

```
{
  "encoderConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
    "name": "test-ec",
    "tags": {},
    "video": {
      "bitrate": 3500000,
      "framerate": 30,
      "height": 1080,
      "width": 1920
    }
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetEncoderConfiguration](#) in der AWS CLI Befehlsreferenz.

get-participant

Das folgende Codebeispiel zeigt die Verwendung `get-participant`.

AWS CLI

Um einen Bühnenteilnehmer zu bekommen

Im folgenden `get-participant` Beispiel wird der Phasenteilnehmer für eine angegebene Teilnehmer-ID und Sitzungs-ID in der angegebenen Phase ARN (Amazon-Ressourcenname) abgerufen.

```
aws ivs-realtime get-participant \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g \
  --participant-id abCDEf12GHIj
```

Ausgabe:

```
{
  "participant": {
    "browserName", "Google Chrome",
    "browserVersion", "116",
    "firstJoinTime": "2023-04-26T20:30:34+00:00",
    "ispName", "Comcast",
    "osName", "Microsoft Windows 10 Pro",
    "osVersion", "10.0.19044"
    "participantId": "abCDEf12GHIj",
    "published": true,
    "recordingS3BucketName": "bucket-name",
    "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
    "recordingState": "ACTIVE",
    "sdkVersion", "",
    "state": "CONNECTED",
```

```

    "userId": "",
  }
}

```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetParticipant](#) in der AWS CLI Befehlsreferenz.

get-public-key

Das folgende Codebeispiel zeigt die Verwendung `get-public-key`.

AWS CLI

Um einen vorhandenen öffentlichen Schlüssel abzurufen, der zum Signieren von Etappenteilnehmer-Token verwendet wird

Im folgenden `get-public-key` Beispiel wird ein öffentlicher Schlüssel abgerufen, der durch die bereitgestellten Daten spezifiziert wird ARN, zum Signieren von Etappenteilnehmer-Token.

```

aws ivs-realtime get-public-key \
  --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2

```

Ausgabe:

```

{
  "publicKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
    "name": "",
    "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHkoZIZj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\nn9pmzcpWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcF00AzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPE\n-----END PUBLIC KEY-----",
    "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
    "tags": {}
  }
}

```

Weitere Informationen finden Sie unter [Verteilung von Teilnehmer-Token](#) im Amazon IVS Real-Time Streaming-Benutzerhandbuch.

- API-Einheiten finden Sie [GetPublicKey](#) in der AWS CLI Befehlsreferenz.

get-stage-session

Das folgende Codebeispiel zeigt die Verwendung `get-stage-session`.

AWS CLI

Um eine Bühnensitzung zu bekommen

Im folgenden `get-stage-session` Beispiel wird die Stage-Sitzung für eine angegebene Sitzungs-ID einer bestimmten Phase ARN (Amazon Resource Name) abgerufen.

```
aws ivs-realtime get-stage-session \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g
```

Ausgabe:

```
{
  "stageSession": {
    "endTime": "2023-04-26T20:36:29+00:00",
    "sessionId": "st-a1b2c3d4e5f6g",
    "startTime": "2023-04-26T20:30:29.602000+00:00"
  }
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStageSession](#) in der AWS CLI Befehlsreferenz.

get-stage

Das folgende Codebeispiel zeigt die Verwendung `get-stage`.

AWS CLI

Um die Konfigurationsinformationen einer Phase abzurufen

Im folgenden `get-stage` Beispiel wird die Stufenkonfiguration für eine angegebene Phase ARN (Amazon Resource Name) abgerufen.

```
aws ivs-realtime get-stage \  
--arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Ausgabe:

```
{  
  "stage": {  
    "activeSessionId": "st-a1b2c3d4e5f6g",  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",  
    "autoParticipantRecordingConfiguration": {  
      "mediaTypes": [  
        "AUDIO_VIDEO"  
      ],  
      "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-  
configuration/abcdABCDefgh",  
    },  
    "endpoints": {  
      "events": "wss://global.events.live-video.net",  
      "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"  
    },  
    "name": "test",  
    "tags": {}  
  }  
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStage](#) in der AWS CLI Befehlsreferenz.

get-storage-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-storage-configuration`.

AWS CLI

Um eine Speicherkonfiguration für Kompositionen zu erhalten

Im folgenden `get-storage-configuration` Beispiel wird die durch den angegebenen Wert ARN (Amazon-Ressourcenname) angegebene Konfiguration des Kompositionsspeichers abgerufen.

```
aws ivs-realtime get-storage-configuration \
  --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh"
```

Ausgabe:

```
{
  "storageConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
    "name": "test-sc",
    "s3": {
      "bucketName": "test-bucket-name"
    },
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStorageConfiguration](#) in der AWS CLI Befehlsreferenz.

import-public-key

Das folgende Codebeispiel zeigt die Verwendung `import-public-key`.

AWS CLI

Um einen vorhandenen öffentlichen Schlüssel zu importieren, der zum Signieren von Etappenteilnehmer-Token verwendet werden soll

Im folgenden `import-public-key` Beispiel wird ein öffentlicher Schlüssel aus einer Materialdatei importiert, der zum Signieren von Etappenteilnehmer-Token verwendet werden soll.

```
aws ivs-realtime import-public-key \
  --public-key-material="`cat public.pem`"
```

Ausgabe:

```
{
```

```

    "publicKey": {
      "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
      "name": "",
      "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHkoZIZj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\nn9pmzcpIWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcF00AzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPE\n-----END PUBLIC KEY-----",
      "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
      "tags": {}
    }
  }
}

```

Weitere Informationen finden Sie unter [Verteilung von Teilnehmer-Token](#) im Amazon IVS Real-Time Streaming-Benutzerhandbuch.

- API-Einheiten finden Sie [ImportPublicKey](#) in der AWS CLI Befehlsreferenz.

list-compositions

Das folgende Codebeispiel zeigt die Verwendung `list-compositions`.

AWS CLI

Um eine Liste von Kompositionen zu erhalten

Im Folgenden `list-compositions` sind alle Kompositionen für Ihr AWS Konto in der AWS Region aufgeführt, in der die API-Anfrage bearbeitet wird.

```
aws ivs-realtime list-compositions
```

Ausgabe:

```

{
  "compositions": [
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
abcdABCDefgh",
      "destinations": [
        {
          "id": "AabBCcdDEefF",
          "startTime": "2023-10-16T23:25:23+00:00",
          "state": "ACTIVE"
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
defgABCdabcd",
  "startTime": "2023-10-16T23:25:21+00:00",
  "state": "ACTIVE",
  "tags": {}
},
{
  "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
ABcdabCDefgh",
  "destinations": [
    {
      "endTime": "2023-10-16T23:25:00.786512+00:00",
      "id": "aABbcCDdeEff",
      "startTime": "2023-10-16T23:24:01+00:00",
      "state": "STOPPED"
    },
    {
      "endTime": "2023-10-16T23:25:00.786512+00:00",
      "id": "deEffaABbcCD",
      "startTime": "2023-10-16T23:24:01+00:00",
      "state": "STOPPED"
    }
  ],
  "endTime": "2023-10-16T23:25:00+00:00",
  "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
efghabcdABCD",
  "startTime": "2023-10-16T23:24:00+00:00",
  "state": "STOPPED",
  "tags": {}
}
]
}

```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListCompositions](#) in der AWS CLI Befehlsreferenz.

list-encoder-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-encoder-configurations`.

AWS CLI

Um die Konfigurationen des Kompositions-Encoders aufzulisten

Im Folgenden `list-encoder-configurations` sind alle Composition Encoder-Konfigurationen für Ihr AWS Konto in der AWS Region aufgeführt, in der die API Anfrage bearbeitet wird.

```
aws ivs-realtime list-encoder-configurations
```

Ausgabe:

```
{
  "encoderConfigurations": [
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
      "name": "test-ec-1",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABCefgEFGabc",
      "name": "test-ec-2",
      "tags": {}
    }
  ]
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListEncoderConfigurations](#) in der AWS CLI Befehlsreferenz.

list-participant-events

Das folgende Codebeispiel zeigt die Verwendung `list-participant-events`.

AWS CLI

Um eine Liste der Veranstaltungen der Bühnenteilnehmer zu erhalten

Das folgende `list-participant-events` Beispiel listet alle Teilnehmerereignisse für eine angegebene Teilnehmer-ID und Sitzungs-ID einer bestimmten Phase ARN (Amazon-Ressourcenname) auf.

```
aws ivs-realtime list-participant-events \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --session-id st-a1b2c3d4e5f6g \  
  --participant-id abCDEf12GHIj
```

Ausgabe:

```
{  
  "events": [  
    {  
      "eventTime": "2023-04-26T20:36:28+00:00",  
      "name": "LEFT",  
      "participantId": "abCDEf12GHIj"  
    },  
    {  
      "eventTime": "2023-04-26T20:36:28+00:00",  
      "name": "PUBLISH_STOPPED",  
      "participantId": "abCDEf12GHIj"  
    },  
    {  
      "eventTime": "2023-04-26T20:30:34+00:00",  
      "name": "JOINED",  
      "participantId": "abCDEf12GHIj"  
    },  
    {  
      "eventTime": "2023-04-26T20:30:34+00:00",  
      "name": "PUBLISH_STARTED",  
      "participantId": "abCDEf12GHIj"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListParticipantEvents](#) in der AWS CLI Befehlsreferenz.

list-participants

Das folgende Codebeispiel zeigt die Verwendung `list-participants`.

AWS CLI

Um eine Liste der Teilnehmer der Phase zu erhalten

Das folgende `list-participants` Beispiel listet alle Teilnehmer für eine angegebene Sitzungs-ID einer bestimmten Phase ARN (Amazon Resource Name) auf.

```
aws ivs-realtime list-participants \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g
```

Ausgabe:

```
{
  "participants": [
    {
      "firstJoinTime": "2023-04-26T20:30:34+00:00",
      "participantId": "abCDEf12GHIj"
      "published": true,
      "recordingState": "STOPPED",
      "state": "DISCONNECTED",
      "userId": ""
    }
  ]
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListParticipants](#) in der AWS CLI Befehlsreferenz.

list-public-keys

Das folgende Codebeispiel zeigt die Verwendung `list-public-keys`.

AWS CLI

Um vorhandene öffentliche Schlüssel aufzulisten, die zum Signieren von Etappenteilnehmer-Token verfügbar sind

Das folgende `list-public-keys` Beispiel listet alle öffentlichen Schlüssel auf, die in der AWS Region, in der die API Anfrage bearbeitet wird, zum Signieren von Etappenteilnehmer-Token verfügbar sind.

```
aws ivs-realtime list-public-keys
```

Ausgabe:

```
{
  "publicKeys": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
      "name": "",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/3bcdABCDefg4",
      "name": "",
      "tags": {}
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verteilung von Teilnehmer-Token](#) im Amazon IVS Real-Time Streaming-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPublicKeys](#) in der AWS CLI Befehlsreferenz.

list-stage-sessions

Das folgende Codebeispiel zeigt die Verwendung `list-stage-sessions`.

AWS CLI

Um eine Liste der Bühnensitzungen zu erhalten

Das folgende `list-stage-sessions` Beispiel listet alle Sitzungen für eine angegebene Phase ARN (Amazon-Ressourcenname) auf.

```
aws ivs-realtime list-stage-sessions \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Ausgabe:

```
{
  "stageSessions": [
    {
      "endTime": "2023-04-26T20:36:29+00:00",
      "sessionId": "st-a1b2c3d4e5f6g",
      "startTime": "2023-04-26T20:30:29.602000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStageSessions](#) in der AWS CLI Befehlsreferenz.

list-stages

Das folgende Codebeispiel zeigt die Verwendung `list-stages`.

AWS CLI

Um zusammenfassende Informationen zu allen Phasen zu erhalten

Das folgende `list-stages` Beispiel listet alle Phasen Ihres AWS Kontos in der AWS Region auf, in der die API-Anfrage bearbeitet wird.

```
aws ivs-realtime list-stages
```

Ausgabe:

```
{
  "stages": [
    {
      "activeSessionId": "st-a1b2c3d4e5f6g",
      "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
      "name": "stage1",
      "tags": {}
    },
    {
```

```
    "activeSessionId": "st-a123bcd456efg",
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcd1234ABCD",
    "name": "stage2",
    "tags": {}
  },
  {
    "activeSessionId": "st-abcDEF1234ghi",
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/ABCD1234efgh",
    "name": "stage3",
    "tags": {}
  }
]
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStages](#) in der AWS CLI Befehlsreferenz.

list-storage-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-storage-configurations`.

AWS CLI

Um die Konfigurationen des Kompositionsspeichers aufzulisten

Im Folgenden `list-storage-configurations` werden alle Konfigurationen des Kompositionsspeichers für Ihr AWS Konto in der AWS Region aufgeführt, in der die API-Anfrage bearbeitet wird.

```
aws ivs-realtime list-storage-configurations
```

Ausgabe:

```
{
  "storageConfigurations": [
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/abcdABCDefgh",
      "name": "test-sc-1",
      "s3": {
```

```

        "bucketName": "test-bucket-1-name"
    },
    "tags": {}
  },
  {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABCefgEFGabc",
    "name": "test-sc-2",
    "s3": {
      "bucketName": "test-bucket-2-name"
    },
    "tags": {}
  }
]
}

```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStorageConfigurations](#) in der AWS CLI Befehlsreferenz.

start-composition

Das folgende Codebeispiel zeigt die Verwendung `start-composition`.

AWS CLI

Beispiel 1: Um eine Komposition mit Standard-Layouteinstellungen zu beginnen

Im folgenden `start-composition` Beispiel wird eine Komposition für die angegebene Phase gestartet, die an die angegebenen Speicherorte gestreamt wird.

```

aws ivs-realtime start-composition \
  --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
  --destinations '["channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
    "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
    {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"}], \
    "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}]}]'

```

Ausgabe:

```

{
  "composition": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
    "destinations": [
      {
        "configuration": {
          "channel": {
            "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
            "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
          },
          "name": ""
        },
        "id": "AabBCcdDEefF",
        "state": "STARTING"
      },
      {
        "configuration": {
          "name": "",
          "s3": {
            "encoderConfigurationArns": [
              "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
            ],
            "recordingConfiguration": {
              "format": "HLS"
            },
            "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
          }
        },
        "detail": {
          "s3": {
            "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
          }
        },
        "id": "GHFabcgefABC",
        "state": "STARTING"
      }
    ],
  },
}

```



```

    "layout": {
      "grid": {
        "featuredParticipantAttribute": ""
        "gridGap": 2,
        "omitStoppedVideo": false,
        "videoAspectRatio": "VIDEO",
        "videoFillMode": ""
      }
    },
    "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
    "startTime": "2023-10-16T23:24:00+00:00",
    "state": "STARTING",
    "tags": {}
  }
}

```

Weitere Informationen finden Sie unter [Composite Recording \(Echtzeit-Streaming\)](#) im Amazon Interactive Video Service-Benutzerhandbuch.

Beispiel 2: Um eine Komposition mit PiP-Layout zu starten

Im folgenden `start-composition` Beispiel wird eine Komposition für die angegebene Bühne gestartet, die mithilfe des PiP-Layouts an die angegebenen Orte gestreamt wird.

```

aws ivs-realtime start-composition \
  --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd \
  --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg", \
  "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}}, \
  {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
  "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}}]' \
  --layout pip='{featuredParticipantAttribute="abcdefg"}'

```

Ausgabe:

```

{
  "composition": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
    "destinations": [
      {

```

```

        "configuration": {
            "channel": {
                "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
            },
            "name": ""
        },
        "id": "AabBCcdDEefF",
        "state": "STARTING"
    },
    {
        "configuration": {
            "name": "",
            "s3": {
                "encoderConfigurationArns": [
                    "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                ],
                "recordingConfiguration": {
                    "format": "HLS"
                },
                "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
            }
        },
        "detail": {
            "s3": {
                "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
            }
        },
        "id": "GHFabcgefABC",
        "state": "STARTING"
    }
],
"layout": {
    "pip": {
        "featuredParticipantAttribute": "abcdefg",
        "gridGap": 0,
        "omitStoppedVideo": false,
        "pipBehavior": "STATIC",
        "pipOffset": 0,

```

```
        "pipParticipantAttribute": "",
        "pipPosition": "BOTTOM_RIGHT",
        "videoFillMode": "COVER"
    }
},
"stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCdabcd",
"startTime": "2023-10-16T23:24:00+00:00",
"state": "STARTING",
"tags": {}
}
```

Weitere Informationen finden Sie unter [Composite Recording \(Echtzeit-Streaming\)](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [StartComposition](#) unter AWS CLI Befehlsreferenz.

stop-composition

Das folgende Codebeispiel zeigt die Verwendung `stop-composition`.

AWS CLI

Um eine Komposition zu beenden

Im Folgenden wird die durch den angegebenen ARN (Amazon-Ressourcennamen) angegebene Zusammensetzung `stop-composition` gestoppt.

```
aws ivs-realtime stop-composition \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [StopComposition](#) in der AWS CLI Befehlsreferenz.

update-stage

Das folgende Codebeispiel zeigt die Verwendung `update-stage`.

AWS CLI

Um die Konfiguration einer Phase zu aktualisieren

Im folgenden `update-stage` Beispiel wird eine Phase für eine angegebene Phase aktualisiert, um den Namen der Phase zu aktualisieren und die Aufzeichnung einzelner Teilnehmer zu konfigurieren.

```
aws ivs-realtime update-stage \
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --auto-participant-recording-configuration '{"mediaTypes":
["AUDIO_VIDEO"], "storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh"}' \
  --name stage1a
```

Ausgabe:

```
{
  "stage": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
    "autoParticipantRecordingConfiguration": {
      "mediaTypes": [
        "AUDIO_VIDEO"
      ],
      "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
    },
    "endpoints": {
      "events": "wss://global.events.live-video.net",
      "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
    },
    "name": "stage1a",
    "tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Aktivieren mehrerer Hosts auf einem Amazon IVS Stream](#) im Amazon Interactive Video Service-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateStage](#) in der AWS CLI Befehlsreferenz.

Amazon Kendra Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Kendra Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-data-source

Das folgende Codebeispiel zeigt die Verwendung `create-data-source`.

AWS CLI

So erstellen Sie einen Amazon Kendra Kendra-Datenquellen-Connector

Im Folgenden `create-data-source` wird ein Amazon Kendra-Datenquellen-Connector erstellt und konfiguriert. Sie können `describe-data-source` damit den Status eines Datenquellen-Connectors anzeigen und alle Fehlermeldungen lesen, falls der Status einen Datenquellen-Connector "FAILED" anzeigt, um ihn vollständig zu erstellen.

```
aws kendra create-data-source \
  --name "example data source 1" \
  --description "Example data source 1 for example index 1 contains the first set
of example documents" \
  --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",
"Value": "aws"}' \
  --role-arn "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource" \
  --index-id exampleindex1 \
  --language-code "es" \
  --schedule "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *" \
```

```
--configuration '{"TemplateConfiguration": {"Template": file://
s3schemaconfig.json}}' \
--type "TEMPLATE" \
--custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":
{"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
"S3Bucket": "s3://my-s3-bucket/scanned-image-text-example-docs"}, "RoleArn":
"arn:aws:iam:my-account-id:role/KendraRoleForCDE"}' \
--vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":
["subnet-1c234", "subnet-2b134"]}'
```

Ausgabe:

```
{
  "Id": "exampledatasource1"
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit einem Amazon Kendra-Index- und Datenquellen-Connector](#) im Amazon Kendra Developer Guide.

- API-Einheiten finden Sie [CreateDataSource](#) in der AWS CLI Befehlsreferenz.

create-index

Das folgende Codebeispiel zeigt die Verwendung `create-index`.

AWS CLI

So erstellen Sie einen Amazon Kendra Kendra-Index

Im Folgenden `create-index` wird ein Amazon Kendra Kendra-Index erstellt und konfiguriert. Sie können ihn verwenden `describe-index`, um den Status eines Indexes anzuzeigen und alle Fehlermeldungen zu lesen, falls der Status einen Index "FAILED" anzeigt, um den Index vollständig zu erstellen.

```
aws kendra create-index \
--name "example index 1" \
--description "Example index 1 contains the first set of example documents" \
--tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",
"Value": "aws"}' \
--role-arn "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex" \
--edition "DEVELOPER_EDITION" \
--server-side-encryption-configuration '{"KmsKeyId": "my-kms-key-id"}' \
```

```
--user-context-policy "USER_TOKEN" \  
--user-token-configurations '{"JsonTokenTypeConfiguration":  
{"GroupAttributeField": "groupNameField", "UserNameAttributeField":  
"userNameField"}}'
```

Ausgabe:

```
{  
  "Id": index1  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit einem Amazon Kendra-Index- und Datenquellen-Connector](#) im Amazon Kendra Developer Guide.

- API-Einheiten finden Sie [CreateIndex](#) in der AWS CLI Befehlsreferenz.

describe-data-source

Das folgende Codebeispiel zeigt die Verwendung `describe-data-source`.

AWS CLI

Um Informationen über einen Amazon Kendra Datenquellen-Connector zu erhalten

Im Folgenden finden `describe-data-source` Sie Informationen zu einem Amazon Kendra Datenquellen-Connector. Sie können die Konfiguration eines Datenquellen-Connectors einsehen und alle Fehlermeldungen lesen, falls der Status anzeigt, dass ein Datenquellen-Connector "FAILED" vollständig erstellt werden muss.

```
aws kendra describe-data-source \  
--id exampledatasource1 \  
--index-id exampleindex1
```

Ausgabe:

```
{  
  "Configuration": {  
    "TemplateConfiguration": {  
      "Template": {  
        "connectionConfiguration": {  
          "repositoryEndpointMetadata": {  
            "BucketName": "my-bucket"  
          }  
        }  
      }  
    }  
  }  
}
```

```
    }
  },
  "repositoryConfigurations": {
    "document":{
      "fieldMappings": [
        {
          "indexFieldName": "_document_title",
          "indexFieldType": "STRING",
          "dataSourceFieldName": "title"
        },
        {
          "indexFieldName": "_last_updated_at",
          "indexFieldType": "DATE",
          "dataSourceFieldName": "modified_date"
        }
      ]
    }
  },
  "additionalProperties": {
    "inclusionPatterns": [
      "*.txt",
      "*.doc",
      "*.docx"
    ],
    "exclusionPatterns": [
      "*.json"
    ],
    "inclusionPrefixes": [
      "PublicExampleDocsFolder"
    ],
    "exclusionPrefixes": [
      "PrivateDocsFolder/private"
    ],
    "aclConfigurationFilePath": "ExampleDocsFolder/AclConfig.json",
    "metadataFilesPrefix": "metadata"
  },
  "syncMode": "FULL_CRAWL",
  "type": "S3",
  "version": "1.0.0"
}
},
"CreatedAt": 2024-02-25T13:30:10+00:00,
"CustomDocumentEnrichmentConfiguration": {
```



```

    "PostExtractionHookConfiguration": {
      "LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
      "S3Bucket": "s3://my-s3-bucket/scanned-image-text-example-docs/function"
    },
    "RoleArn": "arn:aws:iam:my-account-id:role/KendraRoleForCDE"
  }
  "Description": "Example data source 1 for example index 1 contains the first set
of example documents",
  "Id": "exampledatasource1",
  "IndexId": "exampleindex1",
  "LanguageCode": "en",
  "Name": "example data source 1",
  "RoleArn": "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource",
  "Schedule": "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *",
  "Status": "ACTIVE",
  "Type": "TEMPLATE",
  "UpdatedAt": 1709163615,
  "VpcConfiguration": {
    "SecurityGroupIds": ["sg-1234567890abcdef0"],
    "SubnetIds": ["subnet-1c234", "subnet-2b134"]
  }
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit einem Amazon Kendra-Index- und Datenquellen-Connector](#) im Amazon Kendra Developer Guide.

- API-Einheiten finden Sie [DescribeDataSource](#) in der AWS CLI Befehlsreferenz.

describe-index

Das folgende Codebeispiel zeigt die Verwendung `describe-index`.

AWS CLI

Um Informationen über einen Amazon Kendra Kendra-Index zu erhalten

Im Folgenden finden `describe-index` Sie Informationen zu einem Amazon Kendra Kendra-Index. Sie können die Konfiguration eines Indexes einsehen und alle Fehlermeldungen lesen, falls der Status einen Index "FAILED" anzeigt, der vollständig erstellt werden soll.

```
aws kendra describe-index \
```

```
--id exampleindex1
```

Ausgabe:

```
{
  "CapacityUnits": {
    "QueryCapacityUnits": 0,
    "StorageCapacityUnits": 0
  },
  "CreatedAt": 2024-02-25T12:30:10+00:00,
  "Description": "Example index 1 contains the first set of example documents",
  "DocumentMetadataConfigurations": [
    {
      "Name": "_document_title",
      "Relevance": {
        "Importance": 8
      },
      "Search": {
        "Displayable": true,
        "Facetable": false,
        "Searchable": true,
        "Sortable": false
      },
      "Type": "STRING_VALUE"
    },
    {
      "Name": "_document_body",
      "Relevance": {
        "Importance": 5
      },
      "Search": {
        "Displayable": true,
        "Facetable": false,
        "Searchable": true,
        "Sortable": false
      },
      "Type": "STRING_VALUE"
    },
    {
      "Name": "_last_updated_at",
      "Relevance": {
        "Importance": 6,
        "Duration": "2628000s",

```

```

        "Freshness": true
    },
    "Search": {
        "Displayable": true,
        "Facetable": false,
        "Searchable": true,
        "Sortable": true
    },
    "Type": "DATE_VALUE"
},
{
    "Name": "department_custom_field",
    "Relevance": {
        "Importance": 7,
        "ValueImportanceMap": {
            "Human Resources" : 4,
            "Marketing and Sales" : 2,
            "Research and innvoation" : 3,
            "Admin" : 1
        }
    },
    "Search": {
        "Displayable": true,
        "Facetable": true,
        "Searchable": true,
        "Sortable": true
    },
    "Type": "STRING_VALUE"
}
],
"Edition": "DEVELOPER_EDITION",
"Id": "index1",
"IndexStatistics": {
    "FaqStatistics": {
        "IndexedQuestionAnswersCount": 10
    },
    "TextDocumentStatistics": {
        "IndexedTextBytes": 1073741824,
        "IndexedTextDocumentsCount": 1200
    }
},
"Name": "example index 1",
"RoleArn": "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex",
"ServerSideEncryptionConfiguration": {

```

```

    "KmsKeyId": "my-kms-key-id"
  },
  "Status": "ACTIVE",
  "UpdatedAt": 1709163615,
  "UserContextPolicy": "USER_TOKEN",
  "UserTokenConfigurations": [
    {
      "JsonTokenTypeConfiguration": {
        "GroupAttributeField": "groupNameField",
        "UserNameAttributeField": "userNameField"
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit einem Amazon Kendra-Index- und Datenquellen-Connector](#) im Amazon Kendra Developer Guide.

- API-Einheiten finden Sie [DescribeIndex](#) in der AWS CLI Befehlsreferenz.

update-data-source

Das folgende Codebeispiel zeigt die Verwendung `update-data-source`.

AWS CLI

So aktualisieren Sie einen Amazon Kendra Kendra-Datenquellen-Connector

Im Folgenden wird die Konfiguration eines Amazon Kendra Kendra-Datenquellen-Connectors `update-data-source` aktualisiert. Wenn die Aktion erfolgreich ist, sendet der Service entweder keine Ausgabe, den HTTP Statuscode 200 oder den AWS CLI Rückgabecode 0 zurück. Sie können `describe-data-source` verwenden, um die Konfiguration und den Status eines Datenquellenconnectors anzuzeigen.

```

aws kendra update-data-source \
  --id exampledatasource1 \
  --index-id exampleindex1 \
  --name "new name for example data source 1" \
  --description "new description for example data source 1" \
  --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForExampleDataSource \
  --configuration '{"TemplateConfiguration": {"Template": file://
s3schemanewconfig.json}}' \

```

```

--custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":
{"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
"S3Bucket": "s3://my-s3-bucket/scanned-image-text-example-docs"}, "RoleArn":
"arn:aws:iam:my-account-id:role/KendraNewRoleForCDE"}' \
--language-code "es" \
--schedule "0 0 18 ? * MON,WED,FRI *" \
--vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":
["subnet-1c234", "subnet-2b134"]}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit einem Amazon Kendra-Index- und Datenquellen-Connector](#) im Amazon Kendra Developer Guide.

- API-Einheiten finden Sie [UpdateDataSource](#) in der AWS CLI Befehlsreferenz.

update-index

Das folgende Codebeispiel zeigt die Verwendung `update-index`.

AWS CLI

So aktualisieren Sie einen Amazon Kendra Kendra-Index

Im Folgenden wird die Konfiguration eines Amazon Kendra-Indexes `update-index` aktualisiert. Wenn die Aktion erfolgreich ist, sendet der Service entweder keine Ausgabe, den HTTP Statuscode 200 oder den AWS CLI Rückgabecode 0 zurück. Sie können `describe-index es` verwenden, um die Konfiguration und den Status eines Indexes anzuzeigen.

```

aws kendra update-index \
--id enterpriseindex1 \
--name "new name for Enterprise Edition index 1" \
--description "new description for Enterprise Edition index 1" \
--role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForEnterpriseIndex \
--capacity-units '{"QueryCapacityUnits": 2, "StorageCapacityUnits": 1}' \
--document-metadata-configuration-updates '{"Name": "_document_title",
"Relevance": {"Importance": 6}}, {"Name": "_last_updated_at", "Relevance":
{"Importance": 8}}' \
--user-context-policy "USER_TOKEN" \
--user-token-configurations '{"JsonTokenTypeConfiguration":
{"GroupAttributeField": "groupNameField", "UserNameAttributeField":
"userNameField"}}'

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit einem Amazon Kendra-Index- und Datenquellen-Connector](#) im Amazon Kendra Developer Guide.

- API-Einheiten finden Sie [UpdateIndex](#) in der AWS CLI Befehlsreferenz.

Kinesis-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Kinesis Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-stream

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-stream`.

AWS CLI

Um Tags zu einem Datenstrom hinzuzufügen

Im folgenden `add-tags-to-stream` Beispiel wird dem angegebenen Stream ein Tag mit dem Schlüssel `samplekey` und `example` dem Wert zugewiesen.

```
aws kinesis add-tags-to-stream \  
  --stream-name samplestream \  
  --tags samplekey=example
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [AddTagsToStream AWS CLI](#) Befehlsreferenz.

create-stream

Das folgende Codebeispiel zeigt die Verwendung `create-stream`.

AWS CLI

Um einen Datenstrom zu erstellen

Im folgenden `create-stream` Beispiel wird ein Datenstrom namens `samplestream` mit 3 Shards erstellt.

```
aws kinesis create-stream \  
  --stream-name samplestream \  
  --shard-count 3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Creating a Stream](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [CreateStream AWS CLI](#) Befehlsreferenz.

decrease-stream-retention-period

Das folgende Codebeispiel zeigt die Verwendung `decrease-stream-retention-period`.

AWS CLI

Um die Aufbewahrungsdauer von Datenströmen zu verkürzen

Im folgenden `decrease-stream-retention-period` Beispiel wird die Aufbewahrungsdauer (die Dauer, für die Datensätze zugänglich sind, nachdem sie dem Stream hinzugefügt wurden) eines Streams mit dem Namen `samplestream` auf 48 Stunden verringert.

```
aws kinesis decrease-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 48
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ändern der Datenaufbewahrungsfrist](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [DecreaseStreamRetentionPeriod AWS CLI](#) Befehlsreferenz.

delete-stream

Das folgende Codebeispiel zeigt die Verwendung `delete-stream`.

AWS CLI

Um einen Datenstrom zu löschen

Im folgenden `delete-stream` Beispiel wird der angegebene Datenstrom gelöscht.

```
aws kinesis delete-stream \  
  --stream-name samplestream
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [DeleteStream AWS CLI](#) Befehlsreferenz.

deregister-stream-consumer

Das folgende Codebeispiel zeigt die Verwendung `deregister-stream-consumer`.

AWS CLI

Um einen Datenstromverbraucher abzumelden

Im folgenden `deregister-stream-consumer` Beispiel wird der angegebene Verbraucher vom angegebenen Datenstrom abgemeldet.

```
aws kinesis deregister-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Entwickeln von Verbrauchern mit erweitertem Fan-Out mithilfe der Kinesis Data Streams API](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [DeregisterStreamConsumer](#) in AWS CLI der Befehlsreferenz.

describe-limits

Das folgende Codebeispiel zeigt die Verwendung `describe-limits`.

AWS CLI

Um Shard-Limits zu beschreiben

Im folgenden `describe-limits` Beispiel werden die Shard-Limits und die Nutzung für das aktuelle AWS Konto angezeigt.

```
aws kinesis describe-limits
```

Ausgabe:

```
{
  "ShardLimit": 500,
  "OpenShardCount": 29
}
```

Weitere Informationen finden Sie unter [Resharding a Stream](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [DescribeLimits](#).AWS CLI

describe-stream-consumer

Das folgende Codebeispiel zeigt die Verwendung `describe-stream-consumer`.

AWS CLI

Um einen Datenstromverbraucher zu beschreiben

Das folgende `describe-stream-consumer` Beispiel gibt die Beschreibung des angegebenen Verbrauchers zurück, der für den angegebenen Datenstrom registriert ist.

```
aws kinesis describe-stream-consumer \
```

```
--stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \  
--consumer-name KinesisConsumerApplication
```

Ausgabe:

```
{  
  "ConsumerDescription": {  
    "ConsumerName": "KinesisConsumerApplication",  
    "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/  
consumer/KinesisConsumerApplication:1572383852",  
    "ConsumerStatus": "ACTIVE",  
    "ConsumerCreationTimestamp": 1572383852.0,  
    "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream"  
  }  
}
```

Weitere Informationen finden Sie unter [Lesen von Daten aus Amazon Kinesis Data Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [DescribeStreamConsumer AWS CLI Befehlsreferenz](#).

describe-stream-summary

Das folgende Codebeispiel zeigt die Verwendung `describe-stream-summary`.

AWS CLI

Um eine Zusammenfassung des Datenstroms zu beschreiben

Das folgende `describe-stream-summary` Beispiel enthält eine zusammengefasste Beschreibung (ohne die Shard-Liste) des angegebenen Datenstroms.

```
aws kinesis describe-stream-summary \  
--stream-name samplestream
```

Ausgabe:

```
{  
  "StreamDescriptionSummary": {  
    "StreamName": "samplestream",  
    "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",  
    "StreamStatus": "ACTIVE",
```

```

    "RetentionPeriodHours": 48,
    "StreamCreationTimestamp": 1572297168.0,
    "EnhancedMonitoring": [
      {
        "ShardLevelMetrics": []
      }
    ],
    "EncryptionType": "NONE",
    "OpenShardCount": 3,
    "ConsumerCount": 0
  }
}

```

Weitere Informationen finden Sie unter [Creating and Managing Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [DescribeStreamSummary AWS CLI Befehlsreferenz](#).

describe-stream

Das folgende Codebeispiel zeigt die Verwendung `describe-stream`.

AWS CLI

Um einen Datenstrom zu beschreiben

Das folgende `describe-stream` Beispiel gibt die Details des angegebenen Datenstroms zurück.

```

aws kinesis describe-stream \
  --stream-name samplestream

```

Ausgabe:

```

{
  "StreamDescription": {
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "StartingHashKey": "0",
          "EndingHashKey": "113427455640312821154458202477256070484"
        },
        "SequenceNumberRange": {

```

```
        "StartingSequenceNumber":
"49600871682957036442365024926191073437251060580128653314"
    }
  },
  {
    "ShardId": "shardId-000000000001",
    "HashKeyRange": {
      "StartingHashKey": "113427455640312821154458202477256070485",
      "EndingHashKey": "226854911280625642308916404954512140969"
    },
    "SequenceNumberRange": {
      "StartingSequenceNumber":
"49600871682979337187563555549332609155523708941634633746"
    }
  },
  {
    "ShardId": "shardId-000000000002",
    "HashKeyRange": {
      "StartingHashKey": "226854911280625642308916404954512140970",
      "EndingHashKey": "340282366920938463463374607431768211455"
    },
    "SequenceNumberRange": {
      "StartingSequenceNumber":
"49600871683001637932762086172474144873796357303140614178"
    }
  }
],
"StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
"StreamName": "samplestream",
"StreamStatus": "ACTIVE",
"RetentionPeriodHours": 24,
"EnhancedMonitoring": [
  {
    "ShardLevelMetrics": []
  }
],
"EncryptionType": "NONE",
"KeyId": null,
"StreamCreationTimestamp": 1572297168.0
}
}
```

Weitere Informationen finden Sie unter [Creating and Managing Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [DescribeStream AWS CLI](#) Befehlsreferenz.

disable-enhanced-monitoring

Das folgende Codebeispiel zeigt die Verwendung `disable-enhanced-monitoring`.

AWS CLI

Um die erweiterte Überwachung für Metriken auf Shard-Ebene zu deaktivieren

Im folgenden `disable-enhanced-monitoring` Beispiel wird die erweiterte Kinesis-Datenstream-Überwachung für Metriken auf Shard-Ebene deaktiviert.

```
aws kinesis disable-enhanced-monitoring \  
  --stream-name samplestream --shard-level-metrics ALL
```

Ausgabe:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ],  
  "DesiredShardLevelMetrics": []  
}
```

Weitere Informationen finden Sie unter [Monitoring Streams in Amazon Kinesis Data Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [DisableEnhancedMonitoring](#) in der AWS CLI Befehlsreferenz.

enable-enhanced-monitoring

Das folgende Codebeispiel zeigt die Verwendung `enable-enhanced-monitoring`.

AWS CLI

Um eine erweiterte Überwachung von Metriken auf Shard-Ebene zu ermöglichen

Das folgende `enable-enhanced-monitoring` Beispiel ermöglicht die erweiterte Kinesis-Datenstream-Überwachung für Metriken auf Shard-Ebene.

```
aws kinesis enable-enhanced-monitoring \  
  --stream-name samplestream \  
  --shard-level-metrics ALL
```

Ausgabe:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [],  
  "DesiredShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ]  
}
```

Weitere Informationen finden Sie unter [Monitoring Streams in Amazon Kinesis Data Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [EnableEnhancedMonitoring](#) in der AWS CLI Befehlsreferenz.

get-records

Das folgende Codebeispiel zeigt die Verwendung `get-records`.

AWS CLI

Um Datensätze von einem Shard abzurufen

Im folgenden `get-records` Beispiel werden Datensätze aus dem Shard eines Kinesis-Datenstreams mithilfe des angegebenen Shard-Iterators abgerufen.

```
aws kinesis get-records \
  --shard-iterator AAAAAAAAAAF7/0mWD7IuHj1yGv/
TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYe0ZyLZrvzze0FVf9iN99hUPw/w/
b0YWYeefNvnf1DYt5XpDJghLKr3DzgzknkTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/
GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWGE1
```

Ausgabe:

```
{
  "Records": [],
  "MillisBehindLatest": 80742000
}
```

Weitere Informationen finden Sie unter [Developing Consumers Using the Kinesis Data Streams API with the AWS SDK for Java](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [GetRecords](#) in der AWS CLI Befehlsreferenz.

get-shard-iterator

Das folgende Codebeispiel zeigt die Verwendung `get-shard-iterator`.

AWS CLI

Um einen Shard-Iterator zu erhalten

Das folgende `get-shard-iterator` Beispiel verwendet den `AT_SEQUENCE_NUMBER` Shard-Iteratortyp und generiert einen Shard-Iterator, um mit dem Lesen von Datensätzen genau an der Position zu beginnen, die durch die angegebene Sequenznummer gekennzeichnet ist.

```
aws kinesis get-shard-iterator \
  --stream-name samplestream \
  --shard-id shardId-000000000001 \
  --shard-iterator-type LATEST
```

Ausgabe:

```
{
```

```
"ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqqH9FifJ+n48XWTh/  
IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy  
+1UkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn9408SqeEqwhigwZxNwxID3Ej7YYYcxQi8Q/fIrCjGAy/  
n2r5Z9G864YpWDFn9upNNQAR/ii0Wks"  
}
```

Weitere Informationen finden Sie unter [Developing Consumers Using the Kinesis Data Streams API with the AWS SDK for Java](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [GetShardIterator](#) in der AWS CLI Befehlsreferenz.

increase-stream-retention-period

Das folgende Codebeispiel zeigt die Verwendung `increase-stream-retention-period`.

AWS CLI

Um die Aufbewahrungsdauer von Datenströmen zu verlängern

Im folgenden `increase-stream-retention-period` Beispiel wird die Aufbewahrungsdauer (die Dauer, für die Datensätze zugänglich sind, nachdem sie dem Stream hinzugefügt wurden) des angegebenen Streams auf 168 Stunden erhöht.

```
aws kinesis increase-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 168
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ändern der Datenaufbewahrungsfrist](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [IncreaseStreamRetentionPeriod AWS CLI](#) Befehlsreferenz.

list-shards

Das folgende Codebeispiel zeigt die Verwendung `list-shards`.

AWS CLI

Um Shards in einem Datenstrom aufzulisten

Das folgende `list-shards` Beispiel listet alle Shards im angegebenen Stream auf, beginnend mit dem Shard, dessen ID unmittelbar auf die angegebene von folgt. `exclusive-start-shard-id shardId-000000000000`

```
aws kinesis list-shards \  
  --stream-name samplestream \  
  --exclusive-start-shard-id shardId-000000000000
```

Ausgabe:

```
{  
  "Shards": [  
    {  
      "ShardId": "shardId-000000000001",  
      "HashKeyRange": {  
        "StartingHashKey": "113427455640312821154458202477256070485",  
        "EndingHashKey": "226854911280625642308916404954512140969"  
      },  
      "SequenceNumberRange": {  
        "StartingSequenceNumber":  
"49600871682979337187563555549332609155523708941634633746"  
      }  
    },  
    {  
      "ShardId": "shardId-000000000002",  
      "HashKeyRange": {  
        "StartingHashKey": "226854911280625642308916404954512140970",  
        "EndingHashKey": "340282366920938463463374607431768211455"  
      },  
      "SequenceNumberRange": {  
        "StartingSequenceNumber":  
"49600871683001637932762086172474144873796357303140614178"  
      }  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Shards auflisten](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [ListShards AWS CLI Befehlsreferenz](#).

list-streams

Das folgende Codebeispiel zeigt die Verwendung `list-streams`.

AWS CLI

Um Datenströme aufzulisten

Das folgende `list-streams` Beispiel listet alle aktiven Datenströme im aktuellen Konto und in der Region auf.

```
aws kinesys list-streams
```

Ausgabe:

```
{
  "StreamNames": [
    "samplestream",
    "samplestream1"
  ]
}
```

Weitere Informationen finden Sie unter [Streams auflisten](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [ListStreams AWS CLI Befehlsreferenz](#).

list-tags-for-stream

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-stream`.

AWS CLI

Um Tags für einen Datenstrom aufzulisten

Das folgende `list-tags-for-stream` Beispiel listet die Tags auf, die an den angegebenen Datenstrom angehängt sind.

```
aws kinesys list-tags-for-stream \
  --stream-name samplestream
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "samplekey",
      "Value": "example"
    }
  ],
  "HasMoreTags": false
}
```

Weitere Informationen finden Sie unter [Tagging Your Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [ListTagsForStream AWS CLI](#) Befehlsreferenz.

merge-shards

Das folgende Codebeispiel zeigt die Verwendung `merge-shards`.

AWS CLI

Um Shards zusammenzuführen

Im folgenden `merge-shards` Beispiel werden zwei benachbarte Shards mit IDs den Werten `shardId-000000000000` und `shardId-000000000001` im angegebenen Datenstrom zusammengeführt und zu einem einzigen Shard kombiniert.

```
aws kinesis merge-shards \
  --stream-name samplestream \
  --shard-to-merge shardId-000000000000 \
  --adjacent-shard-to-merge shardId-000000000001
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Merging Two Shards](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [MergeShards](#).AWS CLI

put-record

Das folgende Codebeispiel zeigt die Verwendung `put-record`.

AWS CLI

Um einen Datensatz in einen Datenstrom zu schreiben

Das folgende `put-record` Beispiel schreibt einen einzelnen Datensatz unter Verwendung des angegebenen Partitionsschlüssels in den angegebenen Datenstrom.

```
aws kinesis put-record \  
  --stream-name samplestream \  
  --data sampledatarecord \  
  --partition-key samplepartitionkey
```

Ausgabe:

```
{  
  "ShardId": "shardId-0000000000009",  
  "SequenceNumber": "49600902273357540915989931256901506243878407835297513618",  
  "EncryptionType": "KMS"  
}
```

Weitere Informationen finden Sie unter [Developing Producers Using the Amazon Kinesis Data Streams API with the AWS SDK for Java](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [PutRecord](#) in der AWS CLI Befehlsreferenz.

put-records

Das folgende Codebeispiel zeigt die Verwendung `put-records`.

AWS CLI

Um mehrere Datensätze in einen Datenstrom zu schreiben

Das folgende `put-records` Beispiel schreibt in einem einzigen Aufruf einen Datensatz mit dem angegebenen Partitionsschlüssel und einen anderen Datensatz mit einem anderen Partitionsschlüssel.

```
aws kinesis put-records \  
  --stream-name samplestream \  
  --  
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2
```

Ausgabe:

```
{
  "FailedRecordCount": 0,
  "Records": [
    {
      "SequenceNumber":
"49600883331171471519674795588238531498465399900093808706",
      "ShardId": "shardId-000000000004"
    },
    {
      "SequenceNumber":
"49600902273357540915989931256902715169698037101720764562",
      "ShardId": "shardId-000000000009"
    }
  ],
  "EncryptionType": "KMS"
}
```

Weitere Informationen finden Sie unter [Developing Producers Using the Amazon Kinesis Data Streams API with the AWS SDK for Java](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [PutRecords](#) in der AWS CLI Befehlsreferenz.

register-stream-consumer

Das folgende Codebeispiel zeigt die Verwendung `register-stream-consumer`.

AWS CLI

Um einen Datenstromverbraucher zu registrieren

Im folgenden `register-stream-consumer` Beispiel wird ein Consumer registriert, der `KinesisConsumerApplication` mit dem angegebenen Datenstrom aufgerufen wurde.

```
aws kinesis register-stream-consumer \
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
  --consumer-name KinesisConsumerApplication
```

Ausgabe:

```
{
```

```
"Consumer": {
  "ConsumerName": "KinesisConsumerApplication",
  "ConsumerARN": "arn:aws:kinesis:us-west-2: 123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
  "ConsumerStatus": "CREATING",
  "ConsumerCreationTimestamp": 1572383852.0
}
```

Weitere Informationen finden Sie unter [Entwickeln von Verbrauchern mit erweitertem Fan-Out mithilfe der Kinesis Data Streams API](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [RegisterStreamConsumer](#) in AWS CLI der Befehlsreferenz.

remove-tags-from-stream

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-stream`.

AWS CLI

Um Tags aus einem Datenstrom zu entfernen

Im folgenden `remove-tags-from-stream` Beispiel wird das Tag mit dem angegebenen Schlüssel aus dem angegebenen Datenstrom entfernt.

```
aws kinesis remove-tags-from-stream \
  --stream-name samplestream \
  --tag-keys samplekey
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Your Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter [RemoveTagsFromStream AWS CLI](#) Befehlsreferenz.

split-shard

Das folgende Codebeispiel zeigt die Verwendung `split-shard`.

AWS CLI

Um Shards zu teilen

Im folgenden `split-shard` Beispiel wird der angegebene Shard unter Verwendung eines neuen Start-Hash-Schlüssels von 10 in zwei neue Shards aufgeteilt.

```
aws kinesis split-shard \  
  --stream-name samplestream \  
  --shard-to-split shardId-000000000000 \  
  --new-starting-hash-key 10
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Splitting a Shard](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [SplitShard](#).AWS CLI

start-stream-encryption

Das folgende Codebeispiel zeigt die Verwendung `start-stream-encryption`.

AWS CLI

Um die Verschlüsselung von Datenströmen zu aktivieren

Das folgende `start-stream-encryption` Beispiel aktiviert die serverseitige Verschlüsselung für den angegebenen Stream unter Verwendung des angegebenen AWS KMS Schlüssels.

```
aws kinesis start-stream-encryption \  
  --encryption-type KMS \  
  --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-  
b334-4d3eb496e452 \  
  --stream-name samplestream
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Datenschutz in Amazon Kinesis Data Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [StartStreamEncryption](#) in der AWS CLI Befehlsreferenz.

stop-stream-encryption

Das folgende Codebeispiel zeigt die Verwendung `stop-stream-encryption`.

AWS CLI

Um die Datenstromverschlüsselung zu deaktivieren

Im folgenden `stop-stream-encryption` Beispiel wird die serverseitige Verschlüsselung für den angegebenen Stream mithilfe des angegebenen Schlüssels deaktiviert. AWS KMS

```
aws kinesis start-stream-encryption \  
  --encryption-type KMS \  
  --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-  
b334-4d3eb496e452 \  
  --stream-name samplestream
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Datenschutz in Amazon Kinesis Data Streams](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie [StopStreamEncryption](#) in der AWS CLI Befehlsreferenz.

update-shard-count

Das folgende Codebeispiel zeigt die Verwendung `update-shard-count`.

AWS CLI

Um die Anzahl der Shards in einem Datenstream zu aktualisieren

Im folgenden `update-shard-count` Beispiel wird die Anzahl der Shards des angegebenen Datenstroms auf 6 aktualisiert. In diesem Beispiel wird eine einheitliche Skalierung verwendet, wodurch Shards gleicher Größe erstellt werden.

```
aws kinesis update-shard-count \  
  --stream-name samplestream \  
  --scaling-type UNIFORM_SCALING \  
  --target-shard-count 6
```

Ausgabe:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardCount": 3,
```



```
"TargetShardCount": 6
}
```

Weitere Informationen finden Sie unter [Resharding a Stream](#) im Amazon Kinesis Data Streams Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [UpdateShardCount](#).AWS CLI

AWS KMS Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS KMS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-key-deletion

Das folgende Codebeispiel zeigt die Verwendung `cancel-key-deletion`.

AWS CLI

Um das geplante Löschen eines vom Kunden verwalteten KMS Schlüssels abzurechnen

Im folgenden `cancel-key-deletion` Beispiel wird das geplante Löschen eines vom Kunden verwalteten KMS Schlüssels storniert.

```
aws kms cancel-key-deletion \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Wenn der `cancel-key-deletion` Befehl erfolgreich ist, wird das geplante Löschen abgebrochen. Der Schlüssel hat jedoch den KMS Status `Disabled`, dass Sie ihn nicht für kryptografische KMS Operationen verwenden können. Verwenden Sie den `enable-key` Befehl, um die Funktionalität wiederherzustellen.

Weitere Informationen finden Sie im [Key Management Service Developer Guide unter Das Löschen von AWS Schlüsseln planen und abbrechen](#).

- API Einheiten finden Sie [CancelKeyDeletion](#) in der AWS CLI Befehlsreferenz.

connect-custom-key-store

Das folgende Codebeispiel zeigt die Verwendung `connect-custom-key-store`.

AWS CLI

Um einen benutzerdefinierten Schlüsselspeicher zu verbinden

Im folgenden `connect-custom-key-store` Beispiel wird die Verbindung zum angegebenen benutzerdefinierten Schlüsselspeicher erneut hergestellt. Sie können einen Befehl wie diesen verwenden, um zum ersten Mal eine Verbindung zu einem benutzerdefinierten Schlüsselspeicher herzustellen oder um einen getrennten Schlüsselspeicher wieder zu verbinden.

Sie können diesen Befehl verwenden, um eine Verbindung zu einem AWS HSM Cloud-Schlüsselspeicher oder einem externen Schlüsselspeicher herzustellen.

```
aws kms connect-custom-key-store \
  --custom-key-store-id cks-1234567890abcdef0
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie den Befehl, um zu überprüfen, ob der `describe-custom-key-stores` Befehl wirksam war.

Informationen zum Herstellen einer Verbindung mit einem AWS HSM Cloud-Schlüsselspeicher finden Sie unter [Verbinden und Trennen einer Verbindung zu einem AWS HSM Cloud-Schlüsselspeicher](#) im AWS Key Management Service Developer Guide.

Informationen zum Herstellen einer Verbindung mit einem externen Schlüsselspeicher finden Sie unter [Verbinden und Trennen eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [ConnectCustomKeyStore](#) in der AWS CLI Befehlsreferenz.

create-alias

Das folgende Codebeispiel zeigt die Verwendung `create-alias`.

AWS CLI

Um einen Alias für einen KMS Schlüssel zu erstellen

Der folgende `create-alias` Befehl erstellt einen Alias, der `example-alias` nach dem KMS Schlüssel benannt ist, der durch die Schlüssel-ID identifiziert wird `1234abcd-12ab-34cd-56ef-1234567890ab`.

Aliasnamen müssen mit `alias/` beginnen. Verwenden Sie keine Aliasnamen, die mit `alias/aws` beginnen. Diese sind für die Verwendung durch reserviert AWS.

```
aws kms create-alias \  
  --alias-name alias/example-alias \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie den `list-aliases` Befehl, um den neuen Alias zu sehen.

Weitere Informationen finden Sie unter [Verwenden von Aliasen](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [CreateAlias](#) in der AWS CLI Befehlsreferenz.

create-custom-key-store

Das folgende Codebeispiel zeigt die Verwendung `create-custom-key-store`.

AWS CLI

Beispiel 1: Um einen AWS HSM Cloud-Schlüsselspeicher zu erstellen

Im folgenden `create-custom-key-store` Beispiel wird mithilfe der erforderlichen Parameter ein AWS HSM Cloud-Schlüsselspeicher erstellt, der von einem AWS HSM Cloud-Cluster unterstützt wird. Sie können auch die hinzugefügte `custom-key-store-type` Parameter mit dem Standardwert `AWS_CLOUDHSM`.

Um die Dateieingabe für den `trust-anchor-certificate` Befehl in der anzugeben AWS CLI, ist das `file://` Präfix erforderlich.

```
aws kms create-custom-key-store \
  --custom-key-store-name ExampleCloudHSMKeyStore \
  --cloud-hsm-cluster-id cluster-1a23b4cdefg \
  --key-store-password kmsPswd \
  --trust-anchor-certificate file://customerCA.crt
```

Ausgabe:

```
{
  "CustomKeyId": cks-1234567890abcdef0
}
```

Weitere Informationen finden Sie unter [Erstellen eines AWS HSM Cloud-Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

Beispiel 2: So erstellen Sie einen externen Schlüsselspeicher mit öffentlicher Endpunktkonnektivität

Im folgenden `create-custom-key-store` Beispiel wird ein externer Schlüsselspeicher (XKS) erstellt, mit dem AWS KMS über das Internet kommuniziert.

In diesem Beispiel `XksProxyUriPath` verwendet der ein optionales Präfix von `example-prefix`.

NOTE: Wenn Sie AWS CLI Version 1.0 verwenden, führen Sie den folgenden Befehl aus, bevor Sie einen Parameter mit einem HTTP HTTPS OR-Wert angeben, z. B. den `XksProxyUriEndpoint` Parameter.

```
aws configure set cli_follow_urlparam false
```

Andernfalls ersetzt AWS CLI Version 1.0 den Parameterwert durch den Inhalt, der an dieser URI Adresse gefunden wurde.

```
aws kms create-custom-key-store \
  --custom-key-store-name ExamplePublicEndpointXKS \
  --custom-key-store-type EXTERNAL_KEY_STORE \
  --xks-proxy-connectivity PUBLIC_ENDPOINT \
  --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \
  --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Ausgabe:

```
{
  "CustomKeyId": cks-2234567890abcdef0
}
```

Weitere Informationen finden Sie unter [Erstellen eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

Beispiel 3: So erstellen Sie einen externen Schlüsselspeicher mit VPC Endpoint Service-Konnektivität

Das folgende `create-custom-key-store` Beispiel erstellt einen externen Schlüsselspeicher (XKS), mit dem ein VPC Amazon-Endpunktservice kommuniziert wird AWS KMS.

NOTE: Wenn Sie AWS CLI Version 1.0 verwenden, führen Sie den folgenden Befehl aus, bevor Sie einen Parameter mit einem HTTP HTTPS OR-Wert angeben, z. B. den `XksProxyUriEndpoint` Parameter.

```
aws configure set cli_follow_urlparam false
```

Andernfalls ersetzt AWS CLI Version 1.0 den Parameterwert durch den Inhalt, der an dieser URI Adresse gefunden wurde.

```
aws kms create-custom-key-store \
  --custom-key-store-name ExampleVPCEndpointXKS \
  --custom-key-store-type EXTERNAL_KEY_STORE \
  --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
  --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \
  --xks-proxy-uri-path "/kms/xks/v1" \
  --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-  
example1" \
```

```
--xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawneL2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Ausgabe:

```
{  
  "CustomKeyStoreId": cks-3234567890abcdef0  
}
```

Weitere Informationen finden Sie unter [Erstellen eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [CreateCustomKeyStore](#) unter AWS CLI Befehlsreferenz.

create-grant

Das folgende Codebeispiel zeigt die Verwendung `create-grant`.

AWS CLI

Um einen Zuschuss zu erstellen

Im folgenden `create-grant` Beispiel wird ein Grant erstellt, der es dem `exampleUser` Benutzer ermöglicht, den `decrypt` Befehl auf der `1234abcd-12ab-34cd-56ef-1234567890ab` KMS Beispieltaste zu verwenden. Der ausschließende Schulleiter ist die `adminRole` Rolle. Die Gewährung verwendet die `EncryptionContextSubset` Grant-Beschränkung, um diese Berechtigung nur dann zuzulassen, wenn der Verschlüsselungskontext in der `decrypt` Anforderung das `"Department": "IT"` Schlüssel-Wert-Paar enthält.

```
aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::123456789012:user/exampleUser \  
  --operations Decrypt \  
  --constraints EncryptionContextSubset={Department=IT} \  
  --retiring-principal arn:aws:iam::123456789012:role/adminRole
```

Ausgabe:

```
{  
  "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
```

```
"GrantToken": "<grant token here>"
}
```

Verwenden Sie den Befehl, um detaillierte Informationen zur Gewährung anzuzeigen. `list-grants`

Weitere Informationen finden Sie unter [Grants AWS KMS im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie [CreateGrant](#) in der AWS CLI Befehlsreferenz.

create-key

Das folgende Codebeispiel zeigt die Verwendung `create-key`.

AWS CLI

Beispiel 1: Um einen vom Kunden verwalteten KMS Schlüssel zu erstellen AWS KMS

Im folgenden `create-key` Beispiel wird ein symmetrischer KMS Verschlüsselungsschlüssel erstellt.

Um den KMS Basisschlüssel, einen symmetrischen Verschlüsselungsschlüssel, zu erstellen, müssen Sie keine Parameter angeben. Die Standardwerte für diese Parameter erzeugen einen symmetrischen Verschlüsselungsschlüssel.

Da dieser Befehl keine Schlüsselrichtlinie angibt, erhält der KMS Schlüssel die [Standardschlüsselrichtlinie für programmgesteuert erstellte Schlüssel](#). KMS Verwenden Sie den Befehl, um die Schlüsselrichtlinie anzuzeigen. `get-key-policy` Verwenden Sie den `put-key-policy` Befehl, um die Schlüsselrichtlinie zu ändern.

```
aws kms create-key
```

Der `create-key` Befehl gibt die Schlüsselmetadaten zurück, einschließlich der Schlüssel-ID und ARN des neuen KMS Schlüssels. Sie können diese Werte verwenden, um den KMS Schlüssel in anderen AWS KMS Operationen zu identifizieren. Die Ausgabe enthält die Tags nicht. Um die Tags für einen KMS Schlüssel anzuzeigen, verwenden Sie den `list-resource-tags` command.

Ausgabe:

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2017-07-05T14:04:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_KMS"
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Hinweis: `create-key` Mit dem Befehl können Sie keinen Alias angeben. Um einen Alias für den neuen KMS Schlüssel zu erstellen, verwenden Sie den `create-alias` Befehl.

Weitere Informationen finden Sie unter [Schlüssel erstellen](#) im AWS Key Management Service Developer Guide.

Beispiel 2: So erstellen Sie einen asymmetrischen RSA KMS Schlüssel für die Verschlüsselung und Entschlüsselung

Im folgenden `create-key` Beispiel wird ein KMS Schlüssel erstellt, der ein asymmetrisches RSA key pair für die Verschlüsselung und Entschlüsselung enthält.

```
aws kms create-key \
  --key-spec RSA_4096 \
  --key-usage ENCRYPT_DECRYPT
```

Ausgabe:

```
{
```



```

    "KeyMetadata": {
      "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "AWSAccountId": "111122223333",
      "CreationDate": "2021-04-05T14:04:55-07:00",
      "CustomerMasterKeySpec": "RSA_4096",
      "Description": "",
      "Enabled": true,
      "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
      ],
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyManager": "CUSTOMER",
      "KeySpec": "RSA_4096",
      "KeyState": "Enabled",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "MultiRegion": false,
      "Origin": "AWS_KMS"
    }
  }
}

```

Weitere Informationen finden Sie unter [Asymmetrische Schlüssel AWS KMS im AWS Key Management Service Developer Guide](#).

Beispiel 3: So erstellen Sie einen KMS Schlüssel mit asymmetrischer elliptischer Kurve zum Signieren und Überprüfen

Um einen asymmetrischen KMS Schlüssel zu erstellen, der ein key pair mit asymmetrischer elliptischer Kurve (ECC) zum Signieren und Überprüfen enthält. Der `--key-usage` Parameter ist erforderlich, obwohl er der einzig gültige Wert für Schlüssel `SIGN_VERIFY` ist. ECC KMS

```

aws kms create-key \
  --key-spec ECC_NIST_P521 \
  --key-usage SIGN_VERIFY

```

Ausgabe:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",

```

```

    "AWSAccountId": "111122223333",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "ECC_NIST_P521",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "ECC_NIST_P521",
    "KeyState": "Enabled",
    "KeyUsage": "SIGN_VERIFY",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "SigningAlgorithms": [
      "ECDSA_SHA_512"
    ]
  }
}

```

Weitere Informationen finden Sie unter [Asymmetrische Schlüssel AWS KMS im AWS Key Management Service Developer Guide](#).

Beispiel 4: So erstellen Sie einen Schlüssel HMAC KMS

Das folgende `create-key` Beispiel erstellt einen 384-Bit-Schlüssel HMACKMS. Der `GENERATE_VERIFY_MAC` Wert für den `--key-usage` Parameter ist erforderlich, obwohl er der einzig gültige Wert für HMAC KMS Schlüssel ist.

```

aws kms create-key \
  --key-spec HMAC_384 \
  --key-usage GENERATE_VERIFY_MAC

```

Ausgabe:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-04-05T14:04:55-07:00",
    "CustomerMasterKeySpec": "HMAC_384",
    "Description": "",
    "Enabled": true,

```

```

    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "HMAC_384",
    "KeyState": "Enabled",
    "KeyUsage": "GENERATE_VERIFY_MAC",
    "MacAlgorithms": [
        "HMAC_SHA_384"
    ],
    "MultiRegion": false,
    "Origin": "AWS_KMS"
}
}

```

Weitere Informationen finden Sie unter [HMACKeys AWS KMS in](#) im AWS Key Management Service Developer Guide.

Beispiel 4: So erstellen Sie einen KMS Primärschlüssel mit mehreren Regionen

Im folgenden `create-key` Beispiel wird ein primärer symmetrischer Verschlüsselungsschlüssel für mehrere Regionen erstellt. Da die Standardwerte für alle Parameter einen symmetrischen Verschlüsselungsschlüssel erzeugen, ist nur der `--multi-region` Parameter für diesen Schlüssel erforderlich. KMS Um anzugeben AWS CLI, dass ein boolescher Parameter wahr ist, geben Sie in der einfach den Parameternamen an.

```

aws kms create-key \
  --multi-region

```

Ausgabe:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2021-09-02T016:15:21-09:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",

```

```

    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": true,
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "PRIMARY",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
        "Region": "us-west-2"
      },
      "ReplicaKeys": []
    },
    "Origin": "AWS_KMS"
  }
}

```

Weitere Informationen finden Sie unter [Asymmetrische Schlüssel AWS KMS im AWS Key Management Service Developer Guide](#).

Beispiel 5: So erstellen Sie einen KMS Schlüssel für importiertes Schlüsselmaterial

Im folgenden `create-key` Beispiel wird ein KMS Schlüssel ohne Schlüsselmaterial erstellt. Wenn der Vorgang abgeschlossen ist, können Sie Ihr eigenes Schlüsselmaterial in den KMS Schlüssel importieren. Um diesen KMS Schlüssel zu erstellen, setzen Sie den `--origin` Parameter auf `EXTERNAL`.

```

aws kms create-key \
  --origin EXTERNAL

```

Ausgabe:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": false,

```

```

    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "PendingImport",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL"
  }
}

```

Weitere Informationen finden Sie unter [Importieren von Schlüsselmaterial in AWS KMS Schlüsseln](#) im AWS Key Management Service Developer Guide.

Beispiel 6: So erstellen Sie einen KMS Schlüssel in einem AWS HSM Cloud-Schlüsselspeicher

Das folgende `create-key` Beispiel erstellt einen KMS Schlüssel im angegebenen AWS HSM Cloud-Schlüsselspeicher. Der Vorgang erstellt den KMS Schlüssel und seine Metadaten AWS KMS und erstellt das Schlüsselmaterial im AWS HSM Cloud-Cluster, der dem benutzerdefinierten Schlüsselspeicher zugeordnet ist. Die Parameter `--custom-key-store-id` und `--origin` müssen angegeben werden.

```

aws kms create-key \
  --origin AWS_CLOUDHSM \
  --custom-key-store-id cks-1234567890abcdef0

```

Ausgabe:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [

```

```

        "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_CLOUDHSM"
}
}

```

Weitere Informationen finden Sie unter [AWS HSMCloud-Schlüsselspeicher](#) im AWS Key Management Service Developer Guide.

Beispiel 7: So erstellen Sie einen KMS Schlüssel in einem externen Schlüsselspeicher

Das folgende `create-key` Beispiel erstellt einen KMS Schlüssel im angegebenen externen Schlüsselspeicher. Die `--xks-key-id` Parameter `--custom-key-store-id--origin`, und sind in diesem Befehl erforderlich.

Der `--xks-key-id` Parameter gibt die ID eines vorhandenen symmetrischen Verschlüsselungsschlüssels in Ihrem externen Schlüsselmanager an. Dieser Schlüssel dient als externes Schlüsselmaterial für den Schlüssel. KMS Der Wert des `--origin` Parameters muss sein. Der Parameter muss `EXTERNAL_KEY_STORE` einen externen Schlüsselspeicher identifizieren, der mit seinem externen `custom-key-store-id` Schlüsselspeicher-Proxy verbunden ist.

```

aws kms create-key \
  --origin EXTERNAL_KEY_STORE \
  --custom-key-store-id cks-9876543210fedcba9 \
  --xks-key-id bb8562717f809024

```

Ausgabe:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",

```

```

    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyStoreId": "cks-9876543210fedcba9",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}

```

Weitere Informationen finden Sie unter [Externe Schlüsselspeicher im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie [CreateKey](#) in der AWS CLI Befehlsreferenz.

decrypt

Das folgende Codebeispiel zeigt die Verwendung `decrypt`.

AWS CLI

Beispiel 1: Um eine verschlüsselte Nachricht mit einem symmetrischen KMS Schlüssel zu entschlüsseln (Linux und macOS)

Das folgende `decrypt` Befehlsbeispiel zeigt die empfohlene Methode zum Entschlüsseln von Daten mit dem AWS CLI. Diese Version zeigt, wie Daten unter einem symmetrischen KMS Schlüssel entschlüsselt werden.

Geben Sie den Chiffretext in einer Datei an. Verwenden Sie im Wert des `--ciphertext-blob` Parameters das `fileb://` Präfix, das angibt, die Daten aus einer Binärdatei CLI zu lesen. Wenn sich die Datei nicht im aktuellen Verzeichnis befindet, geben Sie den vollständigen Dateipfad ein. Weitere Informationen zum Lesen von AWS CLI Parameterwerten aus einer Datei finden Sie unter

Loading AWS CLI parameters from a file < <https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html>> im AWS Command Line Interface User Guide und Best Practices for Local File Parameters< <https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>> im AWS Command Line Tool Blog .Geben Sie den KMS Schlüssel zum Entschlüsseln des Chiffretextes an. Der `--key-id` Parameter ist bei der Entschlüsselung mit einem symmetrischen KMS Schlüssel nicht erforderlich. AWS KMS kann die Schlüssel-ID des KMS Schlüssels, der zur Verschlüsselung der Daten verwendet wurde, aus den Metadaten im Chiffretext abrufen. Es hat sich jedoch immer bewährt, den KMS Schlüssel anzugeben, den Sie verwenden. Diese Vorgehensweise stellt sicher, dass Sie den KMS Schlüssel verwenden, den Sie beabsichtigen, und verhindert, dass Sie versehentlich einen Chiffretext mit einem KMS Schlüssel entschlüsseln, dem Sie nicht vertrauen. Fordern Sie die Klartext-Ausgabe als Textwert an. Der `--query` Parameter weist an, dass nur der Wert des Felds aus der Ausgabe abgerufen werden soll. Plaintext Der `--output` Parameter gibt die Ausgabe als Text zurück. Base64-dekodiert den Klartext und speichert ihn in einer Datei. Im folgenden Beispiel wird der Wert des Plaintext Parameters über die Pipeline (`|`) an das Base64-Hilfsprogramm übergeben, das ihn dekodiert. Anschließend leitet er die dekodierte Ausgabe in die Datei um (`>`). `ExamplePlaintext`

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto.

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text \
  --query Plaintext | base64 \
  --decode > ExamplePlaintextFile
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Die Ausgabe des `decrypt` Befehls wird base64-dekodiert und in einer Datei gespeichert.

Weitere Informationen finden Sie unter [Decrypt](#) in der AWS Key Management Service-Referenz-API

Beispiel 2: So entschlüsseln Sie eine verschlüsselte Nachricht mit einem symmetrischen KMS Schlüssel (Windows-Eingabeaufforderung)

Das folgende Beispiel ist dasselbe wie das vorherige, außer dass es das `certutil` Hilfsprogramm zur Base64-Decodierung der Klartextdaten verwendet. Für dieses Verfahren sind zwei Befehle erforderlich, wie in den folgenden Beispielen gezeigt.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto.

```
aws kms decrypt ^
  --ciphertext-blob fileb://ExampleEncryptedFile ^
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
  --output text ^
  --query Plaintext > ExamplePlaintextFile.base64
```

Führen Sie den Befehl `certutil` aus.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Ausgabe:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Weitere Informationen finden Sie unter [Decrypt](#) in der AWS Key Management API Service-Referenz.

Beispiel 3: Um eine verschlüsselte Nachricht mit einem asymmetrischen KMS Schlüssel zu entschlüsseln (Linux und macOS)

Das folgende `decrypt` Befehlsbeispiel zeigt, wie Daten entschlüsselt werden, die mit einem RSA asymmetrischen Schlüssel verschlüsselt wurden. KMS

Bei Verwendung eines asymmetrischen KMS Schlüssels ist der `encryption-algorithm` Parameter erforderlich, der den Algorithmus angibt, der zur Verschlüsselung des Klartextes verwendet wird.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem Konto. AWS

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 0987dcb-a09fe-87dc-65ba-ab0987654321 \
  --encryption-algorithm RSAES_OAEP_SHA_256 \
  --output text \
  --query Plaintext | base64 \
```

```
--decode > ExamplePlaintextFile
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Die Ausgabe des decrypt Befehls wird base64-dekodiert und in einer Datei gespeichert.

Weitere Informationen finden Sie unter [Asymmetrische Schlüssel AWS KMS im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie unter [Decrypt](#) in der AWS CLI Befehlsreferenz.

delete-alias

Das folgende Codebeispiel zeigt die Verwendung `delete-alias`.

AWS CLI

Um einen AWS KMS Alias zu löschen

Im folgenden `delete-alias` Beispiel wird der Alias `alias/example-alias` gelöscht. Der Aliasname muss mit `alias/` beginnen.

```
aws kms delete-alias \  
  --alias-name alias/example-alias
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den Befehl, um den Alias zu finden. `list-aliases`

Weitere Informationen finden Sie unter [Löschen eines Alias](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [DeleteAlias](#) in der AWS CLI Befehlsreferenz.

delete-custom-key-store

Das folgende Codebeispiel zeigt die Verwendung `delete-custom-key-store`.

AWS CLI

Um einen benutzerdefinierten Schlüsselspeicher zu löschen

Im folgenden `delete-custom-key-store` Beispiel wird der angegebene benutzerdefinierte Schlüsselspeicher gelöscht.

Das Löschen eines AWS HSM Cloud-Schlüsselspeichers hat keine Auswirkungen auf den zugehörigen HSM Cloud-Cluster. Das Löschen eines externen Schlüsselspeichers hat keine Auswirkungen auf den zugehörigen externen Schlüsselspeicher-Proxy, den externen Schlüsselmanager oder die externen Schlüssel.

NOTE: Bevor Sie einen benutzerdefinierten Schlüsselspeicher löschen können, müssen Sie das Löschen aller KMS Schlüssel im benutzerdefinierten Schlüsselspeicher planen und dann warten, bis diese KMS Schlüssel gelöscht sind. Anschließend müssen Sie die Verbindung zum benutzerdefinierten Schlüsselspeicher trennen. Hilfe bei der Suche nach den KMS Schlüsseln in Ihrem benutzerdefinierten Schlüsselspeicher finden [Sie unter Löschen eines AWS HSM Cloud-Schlüsselspeichers \(API\)](#) im AWS Key Management Service Developer Guide.

```
delete-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie den `describe-custom-key-stores` Befehl, um zu überprüfen, ob der benutzerdefinierte Schlüsselspeicher gelöscht wurde.

Informationen zum Löschen eines AWS HSM Cloud-Schlüsselspeichers finden Sie unter [Löschen eines AWS HSM Cloud-Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

Informationen zum Löschen externer Schlüsselspeicher finden Sie unter [Löschen eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [DeleteCustomKeyStore](#) unter AWS CLI Befehlsreferenz.

delete-imported-key-material

Das folgende Codebeispiel zeigt die Verwendung `delete-imported-key-material`.

AWS CLI

Um importiertes Schlüsselmaterial aus einem KMS Schlüssel zu löschen

Im folgenden `delete-imported-key-material` Beispiel wird Schlüsselmaterial gelöscht, das in einen KMS Schlüssel importiert wurde.

```
aws kms delete-imported-key-material \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um zu überprüfen, ob das Schlüsselmaterial gelöscht wurde, verwenden Sie den `describe-key` Befehl, um nach dem Schlüsselstatus `PendingImport` oder `PendingDeletion` zu suchen.

Weitere Informationen finden Sie unter Löschen von importiertem Schlüsselmaterial < <https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key-material.html> > im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [DeleteImportedKeyMaterial](#) in AWS CLI der Befehlsreferenz.

derive-shared-secret

Das folgende Codebeispiel zeigt die Verwendung `derive-shared-secret`.

AWS CLI

Um ein gemeinsames Geheimnis abzuleiten

Im folgenden `derive-shared-secret` Beispiel wird ein gemeinsamer geheimer Schlüssel mithilfe eines Schlüsselvereinbarungsalgorithmus abgeleitet.

Sie müssen ein NIST asymmetrisches KMS key pair mit elliptischer Kurve (ECC) oder SM2 (nur Regionen China) mit dem `KeyUsage` Wert `to call` verwenden. `KEY_AGREEMENT` `DeriveSharedSecret`

```
aws kms derive-shared-secret \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --key-agreement-algorithm ECDH \
  --public-
key "MIIBIjANBgqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAvH3Yj0wbkLEpU195Cv1cJVjsVNSjwGq3tCLnzXfhVwV
```

Ausgabe:

```
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "SharedSecret": "MEYCIQCKZLWyTk5runarx6XiAkU9gv31bwP0/pHa
+DXFehzdDwIhANwpsIV2g/9SPWLLsF6p/hiSskuIXMTRwqrMdVKWTMHG",
  "KeyAgreementAlgorithm": "ECDH",
  "KeyOrigin": "AWS_KMS"
}
```

Weitere Informationen finden Sie [DeriveSharedSecret](#) in der AWS Key Management Service Reference. API

- API Einheiten finden Sie [DeriveSharedSecret](#) unter AWS CLI Befehlsreferenz.

describe-custom-key-stores

Das folgende Codebeispiel zeigt die Verwendung `describe-custom-key-stores`.

AWS CLI

Beispiel 1: Um Details zu einem AWS HSM Cloud-Schlüsselspeicher abzurufen

Im folgenden `describe-custom-key-store` Beispiel werden Details zum angegebenen AWS HSM Cloud-Schlüsselspeicher angezeigt. Der Befehl ist für alle Typen von benutzerdefinierten Schlüsselspeichern derselbe, aber die Ausgabe unterscheidet sich je nach Schlüsselspeichertyp und, bei einem externen Schlüsselspeicher, dessen Konnektivitätsoption.

Standardmäßig zeigt dieser Befehl Informationen zu allen benutzerdefinierten Schlüsselspeichern im Konto und in der Region an. Verwenden Sie den `custom-key-store-id` Parameter `custom-key-store-name` oder, um Informationen zu einem bestimmten benutzerdefinierten Schlüsselspeicher anzuzeigen.

```
aws kms describe-custom-key-stores \  
  --custom-key-store-name ExampleCloudHSMKeyStore
```

Die Ausgabe dieses Befehls enthält nützliche Informationen über den AWS HSM Cloud-Schlüsselspeicher, einschließlich seines Verbindungsstatus (`ConnectionState`). Wenn der Verbindungsstatus lautet `FAILED`, enthält die Ausgabe ein `ConnectionErrorCode` Feld, das das Problem beschreibt.

Ausgabe:

```
{  
  "CustomKeyStores": [  
    {  
      "CloudHsmClusterId": "cluster-1a23b4cdefg",  
      "ConnectionState": "CONNECTED",  
      "CreationDate": "2022-04-05T14:04:55-07:00",  
      "CustomKeyStoreId": "cks-1234567890abcdef0",  
      "CustomKeyStoreName": "ExampleExternalKeyStore",
```

```

    "TrustAnchorCertificate": "<certificate appears here>"
  }
]
}

```

Weitere Informationen finden Sie unter [AWS HSMCloud-Schlüsselspeicher anzeigen](#) im AWS Key Management Service Developer Guide.

Beispiel 2: Um Details zu einem externen Schlüsselspeicher mit öffentlicher Endpunktkonnektivität abzurufen

Im folgenden `describe-custom-key-store` Beispiel werden Details zum angegebenen externen Schlüsselspeicher angezeigt. Der Befehl ist für alle Typen von benutzerdefinierten Schlüsselspeichern derselbe, aber die Ausgabe unterscheidet sich je nach Schlüsselspeichertyp und, bei einem externen Schlüsselspeicher, dessen Konnektivitätsoption.

Standardmäßig zeigt dieser Befehl Informationen zu allen benutzerdefinierten Schlüsselspeichern im Konto und in der Region an. Verwenden Sie den `custom-key-store-id` Parameter `custom-key-store-name` oder, um Informationen zu einem bestimmten benutzerdefinierten Schlüsselspeicher anzuzeigen.

```

aws kms describe-custom-key-stores \
  --custom-key-store-id cks-9876543210fedcba9

```

Die Ausgabe dieses Befehls enthält nützliche Informationen über den externen Schlüsselspeicher, einschließlich seines Verbindungsstatus (`ConnectionState`). Wenn der Verbindungsstatus lautet `FAILED`, enthält die Ausgabe ein `ConnectionErrorCode` Feld, das das Problem beschreibt.

Ausgabe:

```

{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-9876543210fedcba9",
      "CustomKeyName": "ExampleXKS",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-02T07:48:55-07:00",
      "CustomKeyType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",

```

```
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://myproxy.xks.example.com",
        "UriPath": "/example-prefix/kms/xks/v1"
    }
}
]
```

Weitere Informationen finden Sie unter [Einen externen Schlüsselspeicher anzeigen](#) im AWS Key Management Service Developer Guide.

Beispiel 3: So rufen Sie Details zu einem externen Schlüsselspeicher mit VPC Endpoint Service-Konnektivität ab

Im folgenden `describe-custom-key-store` Beispiel werden Details zum angegebenen externen Schlüsselspeicher angezeigt. Der Befehl ist für alle Typen von benutzerdefinierten Schlüsselspeichern derselbe, aber die Ausgabe unterscheidet sich je nach Schlüsselspeichertyp und, bei einem externen Schlüsselspeicher, dessen Konnektivitätsoption.

Standardmäßig zeigt dieser Befehl Informationen zu allen benutzerdefinierten Schlüsselspeichern im Konto und in der Region an. Verwenden Sie den `custom-key-store-id` Parameter `custom-key-store-name` oder, um Informationen zu einem bestimmten benutzerdefinierten Schlüsselspeicher anzuzeigen.

```
aws kms describe-custom-key-stores \
  --custom-key-store-id cks-2234567890abcdef0
```

Die Ausgabe dieses Befehls enthält nützliche Informationen über den externen Schlüsselspeicher, einschließlich seines Verbindungsstatus (`ConnectionState`). Wenn der Verbindungsstatus lautet `FAILED`, enthält die Ausgabe ein `ConnectionErrorCode` Feld, das das Problem beschreibt.

Ausgabe:

```
{
  "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-3234567890abcdef0",
      "CustomKeyStoreName": "ExampleVPCEExternalKeyStore",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-22T07:48:55-07:00",
```

```

    "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
    "XksProxyConfiguration": {
      "AccessKeyId": "ABCDE12345670EXAMPLE",
      "Connectivity": "VPC_ENDPOINT_SERVICE",
      "UriEndpoint": "https://myproxy-private.xks.example.com",
      "UriPath": "/kms/xks/v1",
      "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-
example1"
    }
  }
]
}

```

Weitere Informationen finden Sie unter [Einen externen Schlüsselspeicher anzeigen](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [DescribeCustomKeyStores](#) unter AWS CLI Befehlsreferenz.

describe-key

Das folgende Codebeispiel zeigt die Verwendung `describe-key`.

AWS CLI

Beispiel 1: Um detaillierte Informationen zu einem KMS Schlüssel zu finden

Im folgenden `describe-key` Beispiel werden detaillierte Informationen zum AWS verwalteten Schlüssel für Amazon S3 im Beispielkonto und in der Region abgerufen. Sie können diesen Befehl verwenden, um Details zu AWS verwalteten Schlüsseln und kundenverwalteten Schlüsseln zu finden.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird ein Aliasnamenwert verwendet, aber Sie können ARN in diesem Befehl eine Schlüssel-ID, einen SchlüsselARN, einen Aliasnamen oder einen Alias verwenden.

```

aws kms describe-key \
  --key-id alias/aws/s3

```

Ausgabe:

```
{
```



```

    "KeyMetadata": {
      "AWSAccountId": "846764612917",
      "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
      "Arn": "arn:aws:kms:us-west-2:846764612917:key/
b8a9477d-836c-491f-857e-07937918959b",
      "CreationDate": 2017-06-30T21:44:32.140000+00:00,
      "Enabled": true,
      "Description": "Default KMS key that protects my S3 objects when no other
key is defined",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "KeyState": "Enabled",
      "Origin": "AWS_KMS",
      "KeyManager": "AWS",
      "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
      ]
    }
  }
}

```

Weitere Informationen finden Sie unter [Schlüssel anzeigen](#) im AWS Key Management Service Developer Guide.

Beispiel 2: Um Details zu einem RSA asymmetrischen KMS Schlüssel abzurufen

Im folgenden `describe-key` Beispiel werden detaillierte Informationen zu einem asymmetrischen RSA KMS Schlüssel abgerufen, der zum Signieren und Überprüfen verwendet wird.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Ausgabe:

```

{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2019-12-02T19:47:14.861000+00:00",
    "CustomerMasterKeySpec": "RSA_2048",

```

```

    "Enabled": false,
    "Description": "",
    "KeyState": "Disabled",
    "Origin": "AWS_KMS",
    "MultiRegion": false,
    "KeyManager": "CUSTOMER",
    "KeySpec": "RSA_2048",
    "KeyUsage": "SIGN_VERIFY",
    "SigningAlgorithms": [
      "RSASSA_PKCS1_V1_5_SHA_256",
      "RSASSA_PKCS1_V1_5_SHA_384",
      "RSASSA_PKCS1_V1_5_SHA_512",
      "RSASSA_PSS_SHA_256",
      "RSASSA_PSS_SHA_384",
      "RSASSA_PSS_SHA_512"
    ]
  }
}

```

Beispiel 3: Um Details zu einem Replikatschlüssel für mehrere Regionen abzurufen

Im folgenden `describe-key` Beispiel werden Metadaten für einen Replikatschlüssel mit mehreren Regionen abgerufen. Dieser Schlüssel für mehrere Regionen ist ein symmetrischer Verschlüsselungsschlüssel. Die Ausgabe eines `describe-key` Befehls für einen beliebigen Schlüssel mit mehreren Regionen gibt Informationen über den Primärschlüssel und alle zugehörigen Replikate zurück.

```

aws kms describe-key \
  --key-id arn:aws:kms:ap-northeast-1:111122223333:key/  

mrk-1234abcd12ab34cd56ef1234567890ab

```

Ausgabe:

```

{
  "KeyMetadata": {
    "MultiRegion": true,
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab",
    "CreationDate": "2021-06-28T21:09:16.114000+00:00",
    "Description": "",
    "Enabled": true,

```

```

    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_KMS",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "PRIMARY",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "us-west-2"
      },
      "ReplicaKeys": [
        {
          "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
          "Region": "eu-west-1"
        },
        {
          "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
          "Region": "ap-northeast-1"
        },
        {
          "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
          "Region": "sa-east-1"
        }
      ]
    }
  }
}

```

Beispiel 4: Um Details zu einem Schlüssel abzurufen HMAC KMS

Im folgenden `describe-key` Beispiel werden detaillierte Informationen zu einem HMAC KMS Schlüssel abgerufen.

```
aws kms describe-key \
```

```
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{
  "KeyMetadata": {
    "AWSAccountId": "123456789012",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2022-04-03T22:23:10.194000+00:00",
    "Enabled": true,
    "Description": "Test key",
    "KeyUsage": "GENERATE_VERIFY_MAC",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "CUSTOMER",
    "CustomerMasterKeySpec": "HMAC_256",
    "MacAlgorithms": [
      "HMAC_SHA_256"
    ],
    "MultiRegion": false
  }
}
```

- API-Einheiten finden Sie [DescribeKey](#) unter AWS CLI Befehlsreferenz.

disable-key-rotation

Das folgende Codebeispiel zeigt die Verwendung `disable-key-rotation`.

AWS CLI

Um die automatische Drehung eines KMS Schlüssels zu deaktivieren

Im folgenden `disable-key-rotation` Beispiel wird die automatische Rotation eines vom Kunden verwalteten KMS Schlüssels deaktiviert. Verwenden Sie den Befehl, um die automatische Rotation wieder zu aktivieren. `enable-key-rotation`

```
aws kms disable-key-rotation \
  --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `get-key-rotation-status` Befehl, um zu überprüfen, ob die automatische Drehung für den KMS Schlüssel deaktiviert ist.

Weitere Informationen finden Sie unter [Rotieren von Schlüsseln](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [DisableKeyRotation](#) unter AWS CLI Befehlsreferenz.

disable-key

Das folgende Codebeispiel zeigt die Verwendung `disable-key`.

AWS CLI

Um einen KMS Schlüssel vorübergehend zu deaktivieren

Im folgenden Beispiel wird der `disable-key` Befehl verwendet, um einen vom Kunden verwalteten KMS Schlüssel zu deaktivieren. Verwenden Sie den `enable-key` Befehl, um den KMS Schlüssel wieder zu aktivieren.

```
aws kms disable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Enabling and Disabling Keys](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie unter [DisableKey AWS CLI](#) Befehlsreferenz.

disconnect-custom-key-store

Das folgende Codebeispiel zeigt die Verwendung `disconnect-custom-key-store`.

AWS CLI

Um die Verbindung zu einem benutzerdefinierten Schlüsselspeicher zu trennen

Im folgenden `disconnect-custom-key-store` Beispiel wird die Verbindung eines benutzerdefinierten Schlüsselspeichers von seinem AWS HSM Cloud-Cluster getrennt. Sie

können die Verbindung zu einem Schlüsselspeicher trennen, um ein Problem zu beheben, seine Einstellungen zu aktualisieren oder um zu verhindern, dass KMS Schlüssel im Schlüsselspeicher für kryptografische Operationen verwendet werden.

Dieser Befehl ist für alle benutzerdefinierten Schlüsselspeicher, einschließlich AWS Cloud-Schlüsselspeicher und externer HSM Schlüsselspeicher, derselbe.

Vor der Ausführung dieses Befehls müssen Sie die Beispiel-ID des benutzerdefinierten Schlüsselspeichers durch eine gültige ID ersetzen.

```
$ aws kms disconnect-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Dieser Befehl erzeugt keine Ausgabe. Vergewissern Sie sich, dass der Befehl wirksam war, und verwenden Sie den `describe-custom-key-stores` Befehl.

Weitere Informationen zum Trennen der Verbindung mit einem AWS HSM Cloud-Schlüsselspeicher finden Sie unter [Einen AWS HSM Cloud-Schlüsselspeicher verbinden und trennen im AWS Key Management Service Developer Guide](#).

Weitere Informationen zum Trennen eines externen Schlüsselspeichers finden Sie unter [Verbinden und Trennen eines externen Schlüsselspeichers im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie unter [DisconnectCustomKeyStore AWS CLI Befehlsreferenz](#).

enable-key-rotation

Das folgende Codebeispiel zeigt die Verwendung von `enable-key-rotation`.

AWS CLI

Um die automatische Rotation eines KMS Schlüssels zu aktivieren

Das folgende `enable-key-rotation` Beispiel ermöglicht die automatische Rotation eines vom Kunden verwalteten KMS Schlüssels mit einem Rotationszeitraum von 180 Tagen. Der KMS Schlüssel wird ab dem Datum, an dem dieser Befehl abgeschlossen wurde, ein Jahr (ungefähr 365 Tage) rotiert, und danach jedes Jahr.

Der `--key-id` Parameter identifiziert den KMS Schlüssel. In diesem Beispiel wird ein ARN Schlüsselwert verwendet, Sie können jedoch entweder die Schlüssel-ID oder die ARN des

Schlüssels verwenden. KMS Der `--rotation-period-in-days` Parameter gibt die Anzahl der Tage zwischen den einzelnen Rotationsdaten an. Geben Sie einen Wert zwischen 90 und 2560 Tagen an. Wenn kein Wert angegeben ist, beträgt der Standardwert 365 Tage.

```
aws kms enable-key-rotation \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --rotation-period-in-days 180
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `get-key-rotation-status` Befehl, um zu überprüfen, ob der KMS Schlüssel aktiviert ist.

Weitere Informationen finden Sie unter [Rotieren von Schlüsseln](#) im AWS Key Management Service Developer Guide.

- API Einzelheiten finden Sie [EnableKeyRotation](#) unter AWS CLI Befehlsreferenz.

enable-key

Das folgende Codebeispiel zeigt die Verwendungen `enable-key`.

AWS CLI

Um einen KMS Schlüssel zu aktivieren

Das folgende `enable-key` Beispiel aktiviert einen vom Kunden verwalteten Schlüssel. Sie können einen Befehl wie diesen verwenden, um eine KMS Taste zu aktivieren, die Sie mit dem `disable-key` Befehl vorübergehend deaktiviert haben. Sie können ihn auch verwenden, um einen KMS Schlüssel zu aktivieren, der deaktiviert ist, weil für ihn ein Löschvorgang geplant war und der Löschvorgang abgebrochen wurde.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird ein Schlüssel-ID-Wert verwendet, aber Sie können in diesem Befehl auch eine Schlüssel-ID oder ARN einen Schlüsselwert verwenden.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige.

```
aws kms enable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `describe-key` Befehl, um zu überprüfen, ob der KMS Schlüssel aktiviert ist. Sehen Sie sich die Werte der `Enabled` Felder `KeyState` und in der `describe-key` Ausgabe an.

Weitere Informationen finden Sie unter [Enabling and Disabling Keys](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie unter [EnableKey AWS CLI](#) Befehlsreferenz.

encrypt

Das folgende Codebeispiel zeigt die Verwendung `encrypt`.

AWS CLI

Beispiel 1: Um den Inhalt einer Datei unter Linux oder macOS zu verschlüsseln

Der folgende `encrypt` Befehl demonstriert die empfohlene Methode zum Verschlüsseln von Daten mit dem AWS CLI

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

Der Befehl macht mehrere Dinge:

Verwendet den `--plaintext` Parameter, um die zu verschlüsselnden Daten anzugeben. Dieser Parameterwert muss Base64-kodiert sein. Der Wert des `plaintext` Parameters muss Base64-kodiert sein, oder Sie müssen das `fileb://` Präfix verwenden, das den mitteilt, Binärdaten aus der Datei AWS CLI zu lesen. Wenn sich die Datei nicht im aktuellen Verzeichnis befindet, geben Sie den vollständigen Dateipfad ein. Beispiel: `fileb:///var/tmp/ExamplePlaintextFile` oder `fileb://C:\Temp\ExamplePlaintextFile`. Weitere Informationen zum Lesen von AWS CLI Parameterwerten aus einer Datei finden Sie unter [Laden von Parametern aus einer Datei](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle und [Best Practices für lokale Dateiparameter](#) im Blog des AWS Befehlszeilentools. Verwendet die `--query` Parameter `--output` und, um die Befehlsausgabe zu steuern. Diese Parameter extrahieren die verschlüsselten Daten, den so genannten Chiffretext, aus der Befehlsausgabe. Weitere Informationen zur

Steuerung der Ausgabe finden Sie unter [Steuern Befehlsausgabe](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle. Verwendet das base64 Hilfsprogramm, um die extrahierte Ausgabe in Binärdaten zu dekodieren. Der Chiffretext, der von einem erfolgreichen encrypt Befehl zurückgegeben wird, ist Base64-codierter Text. Sie müssen diesen Text dekodieren, bevor Sie ihn mit dem entschlüsseln können. Speichert den AWS CLI binären Chiffretext in einer Datei. Der letzte Teil des Befehls () speichert den binären Chiffretext in einer Datei, um die Entschlüsselung zu vereinfachen. > `ExampleEncryptedFile` Einen Beispielbefehl, der zum Entschlüsseln von Daten verwendet, finden Sie in den Entschlüsselungsbeispielen. AWS CLI

Beispiel 2: Verwenden von AWS CLI zum Verschlüsseln von Daten unter Windows

Dieses Beispiel ist dasselbe wie das vorherige, außer dass es das `certutil` Tool anstelle von `base64` verwendet. Für dieses Verfahren sind zwei Befehle erforderlich, wie im folgenden Beispiel gezeigt.

```
aws kms encrypt \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --plaintext fileb://ExamplePlaintextFile \
  --output text \
  --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64

certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

Beispiel 3: Verschlüsselung mit einem asymmetrischen Schlüssel KMS

Der folgende `encrypt` Befehl zeigt, wie Klartext mit einem asymmetrischen Schlüssel verschlüsselt wird. KMS Der Parameter `--encryption-algorithm` muss angegeben werden. Wie bei allen `encrypt` CLI Befehlen muss der `plaintext` Parameter Base64-kodiert sein, oder Sie müssen das `fileb://` Präfix verwenden, das den mitteilt, Binärdaten aus der Datei AWS CLI zu lesen.

```
aws kms encrypt \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --encryption-algorithm RSAES_OAEP_SHA_256 \
  --plaintext fileb://ExamplePlaintextFile \
  --output text \
  --query CiphertextBlob | base64 \
  --decode > ExampleEncryptedFile
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [Verschlüsseln](#) in der Befehlsreferenz.AWS CLI

generate-data-key-pair-without-plaintext

Das folgende Codebeispiel zeigt die Verwendung `generate-data-key-pair-without-plaintext`.

AWS CLI

Um ein asymmetrisches ECC NIST P384-Datenschlüsselpaar zu generieren

Im folgenden `generate-data-key-pair-without-plaintext` Beispiel wird ein ECC NIST P384-Schlüsselpaar zur Verwendung außerhalb von AWS angefordert.

Der Befehl gibt einen öffentlichen Klartext-Schlüssel und eine Kopie des privaten Schlüssels zurück, der unter dem angegebenen KMS Schlüssel verschlüsselt wurde. Es wird kein privater Klartext-Schlüssel zurückgegeben. Sie können den verschlüsselten privaten Schlüssel zusammen mit den verschlüsselten Daten sicher speichern und bei Bedarf aufrufen, AWS KMS um den privaten Schlüssel zu entschlüsseln.

Um ein asymmetrisches ECC NIST P384-Datenschlüsselpaar anzufordern, verwenden Sie den `key-pair-spec` Parameter mit dem Wert `ECC_NIST_P384`.

Bei dem von Ihnen angegebenen KMS Schlüssel muss es sich um einen symmetrischen KMS Verschlüsselungsschlüssel handeln, d. h. um einen KMS Schlüssel mit dem Wert `KeySpec SYMMETRIC_DEFAULT`.

NOTE: Die Werte in der Ausgabe dieses Beispiels sind zur Anzeige gekürzt.

```
aws kms generate-data-key-pair-without-plaintext \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --key-pair-spec ECC_NIST_P384
```

Ausgabe:

```
{
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
  "PublicKey":
  "MIIBoJANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrivSn7+Ld1JE1oUoQV5HpEuHAVbd0yND
+NmYDH/mL10SIEuLrcdZ5hrMH4pk83r401...",
```

```
"KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyPairSpec": "ECC_NIST_P384"  
}
```

Die `PublicKey` und `PrivateKeyCiphertextBlob` werden im Base64-kodierten Format zurückgegeben.

Weitere Informationen finden Sie unter [Datenschlüsselpaare im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie [GenerateDataKeyPairWithoutPlaintext](#) in der AWS CLI Befehlsreferenz.

generate-data-key-pair

Das folgende Codebeispiel zeigt die Verwendung `generate-data-key-pair`.

AWS CLI

Um ein RSA asymmetrisches 2048-Bit-Datenschlüsselpaar zu generieren

Im folgenden `generate-data-key-pair` Beispiel wird ein RSA asymmetrisches 2048-Bit-Datenschlüsselpaar zur Verwendung außerhalb von `aws kms` angefordert. Der Befehl gibt einen öffentlichen Klartext-Schlüssel und einen privaten Klartext-Schlüssel zur sofortigen Verwendung und Löschung sowie eine Kopie des privaten Schlüssels zurück, die unter dem angegebenen Schlüssel verschlüsselt wurde. Sie können den verschlüsselten privaten Schlüssel zusammen mit den verschlüsselten Daten sicher speichern.

Um ein RSA asymmetrisches 2048-Bit-Datenschlüsselpaar anzufordern, verwenden Sie den `key-pair-spec` Parameter mit dem Wert `RSA_2048`.

Bei dem von Ihnen angegebenen KMS Schlüssel muss es sich um einen symmetrischen KMS Verschlüsselungsschlüssel handeln, d. h. um einen KMS Schlüssel mit dem Wert `KeySpec SYMMETRIC_DEFAULT`.

NOTE: Die Werte in der Ausgabe dieses Beispiels sind zur Anzeige gekürzt.

```
aws kms generate-data-key-pair \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-spec RSA_2048
```

```
--key-pair-spec RSA_2048
```

Ausgabe:

```
{
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
  "PrivateKeyPlaintext": "MIIG/
QIBADANBgqhkiG9w0BAQEFAASCBUcwggbjAgEAAoIBgQDcDd4YzI
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P42ZgMf+YvU5IgS4ut...",
  "PublicKey":
  "MIIBojANBgqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrivSn7+Ld1JE1oUoQV5HpEuHAVbd0yND
+NmYDH/mL10SIEuLrcdZ5hrMH4pk83r401...",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyPairSpec": "RSA_2048"
}
```

Die `PublicKeyPrivateKeyPlaintext`, und `PrivateKeyCiphertextBlob` werden im Base64-kodierten Format zurückgegeben.

Weitere Informationen finden Sie unter [Datenschlüsselpaare im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie [GenerateDataKeyPair](#) in der AWS CLI Befehlsreferenz.

generate-data-key-without-plaintext

Das folgende Codebeispiel zeigt die Verwendung `generate-data-key-without-plaintext`.

AWS CLI

Um einen symmetrischen 256-Bit-Datenschlüssel ohne Klartextschlüssel zu generieren

Im folgenden `generate-data-key-without-plaintext` Beispiel wird eine verschlüsselte Kopie eines symmetrischen 256-Bit-Datenschlüssels zur Verwendung außerhalb von angefordert. AWS Sie können aufrufen AWS KMS, um den Datenschlüssel zu entschlüsseln, wenn Sie bereit sind, ihn zu verwenden.

Um einen 256-Bit-Datenschlüssel anzufordern, verwenden Sie den `key-spec` Parameter mit dem Wert `AES_256`. Um einen 128-Bit-Datenschlüssel anzufordern, verwenden Sie den `key-spec`

Parameter mit dem Wert. AES_128 Verwenden Sie für alle anderen Datenschlüssellängen den `number-of-bytes` Parameter.

Der von Ihnen angegebene KMS Schlüssel muss ein symmetrischer KMS Verschlüsselungsschlüssel sein, d. h. ein KMS Schlüssel mit dem Schlüsselpezifikationswert `SYMMETRIC DEFAULT`

```
aws kms generate-data-key-without-plaintext \
  --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
  --key-spec AES_256
```

Ausgabe:

```
{
  "CiphertextBlob":
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhkiG9w0BAQsD
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Der `CiphertextBlob` (verschlüsselte Datenschlüssel) wird im Base64-codierten Format zurückgegeben.

Weitere Informationen finden Sie unter [Datenschlüssel](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [GenerateDataKeyWithoutPlaintext](#) in der AWS CLI Befehlsreferenz.

generate-data-key

Das folgende Codebeispiel zeigt die Verwendung `generate-data-key`.

AWS CLI

Beispiel 1: Um einen symmetrischen 256-Bit-Datenschlüssel zu generieren

Im folgenden `generate-data-key` Beispiel wird ein symmetrischer 256-Bit-Datenschlüssel zur Verwendung außerhalb von `aws kms generate-data-key` angefordert. AWS Der Befehl gibt einen Klartext-Datenschlüssel zur sofortigen Verwendung und Löschung sowie eine Kopie dieses Datenschlüssels zurück, die

unter dem angegebenen Schlüssel verschlüsselt wurde. KMS Sie können den verschlüsselten Datenschlüssel sicher neben den verschlüsselten Daten speichern.

Um einen 256-Bit-Datenschlüssel anzufordern, verwenden Sie den `key-spec` Parameter mit dem Wert. `AES_256` Um einen 128-Bit-Datenschlüssel anzufordern, verwenden Sie den `key-spec` Parameter mit dem Wert. `AES_128` Verwenden Sie für alle anderen Datenschlüssellängen den `number-of-bytes` Parameter.

Der von Ihnen angegebene KMS Schlüssel muss ein symmetrischer KMS Verschlüsselungsschlüssel sein, d. h. ein KMS Schlüssel mit dem Schlüssel spezifikationswert `_SYMMETRIC_DEFAULT`.

```
aws kms generate-data-key \
  --key-id alias/ExampleAlias \
  --key-spec AES_256
```

Ausgabe:

```
{
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "CiphertextBlob":
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhkiG9w0YDhV8MrikBQPeac0ReRVNDt9qlAt+SHgIRF8P0H+7U="
}
```

Der `Plaintext` (Klartext-Datenschlüssel) und der `CiphertextBlob` (verschlüsselte Datenschlüssel) werden im Base64-codierten Format zurückgegeben.

Weitere Informationen finden Sie unter Datenschlüssel < <https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys> im AWS Key Management Service Developer Guide.

Beispiel 2: So generieren Sie einen symmetrischen 512-Bit-Datenschlüssel

Im folgenden `generate-data-key` Beispiel wird ein symmetrischer 512-Bit-Datenschlüssel für die Verschlüsselung und Entschlüsselung angefordert. Der Befehl gibt einen Klartext-Datenschlüssel zur sofortigen Verwendung und Löschung sowie eine Kopie dieses

Datenschlüssels zurück, die mit dem angegebenen Schlüssel verschlüsselt wurde. KMS Sie können den verschlüsselten Datenschlüssel sicher neben den verschlüsselten Daten speichern.

Verwenden Sie den `number-of-bytes` Parameter, um eine andere Schlüssellänge als 128 oder 256 Bit anzufordern. Um einen 512-Bit-Datenschlüssel anzufordern, verwendet das folgende Beispiel den `number-of-bytes` Parameter mit einem Wert von 64 (Byte).

Bei dem von Ihnen angegebenen KMS Schlüssel muss es sich um einen symmetrischen KMS Verschlüsselungsschlüssel handeln, d. h. um einen KMS Schlüssel mit dem Schlüssel spezifikationswert `_`. `SYMMETRIC DEFAULT`

NOTE: Die Werte in der Ausgabe dieses Beispiels sind zur Anzeige gekürzt.

```
aws kms generate-data-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --number-of-bytes 64
```

Ausgabe:

```
{  
  "CiphertextBlob": "AQIBAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y+hAEnX/  
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIB3DQEHbqCCA1UwggJRAgEAMIICSgYJKoZ...",  
  "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB  
+G00ZLtf5MSEb4a13R2UKWG0p06njAwy2n72VRm2m7z/  
Pm9Wpbvttz6a41So9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

`Plaintext`(Klartext-Datenschlüssel) und `CiphertextBlob` (verschlüsselter Datenschlüssel) werden im Base64-codierten Format zurückgegeben.

Weitere Informationen finden Sie unter [Datenschlüssel < https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [GenerateDataKey](#) in der AWS CLI Befehlsreferenz.

generate-random

Das folgende Codebeispiel zeigt die Verwendung `generate-random`.

AWS CLI

Beispiel 1: Um eine 256-Bit-Zufallsbytezeichenfolge zu generieren (Linux oder) macOS

Das folgende `generate-random` Beispiel generiert eine Base64-kodierte 256-Bit-Zufallsbytezeichenfolge (32 Byte). Das Beispiel dekodiert die Bytezeichenfolge und speichert sie in der Zufallsdatei.

Wenn Sie diesen Befehl ausführen, müssen Sie den `number-of-bytes` Parameter verwenden, um die Länge des Zufallswerts in Byte anzugeben.

Sie geben keinen KMS Schlüssel an, wenn Sie diesen Befehl ausführen. Die zufällige Byte-Zeichenfolge hat nichts mit einem KMS Schlüssel zu tun.

AWS KMSGeneriert standardmäßig die Zufallszahl. Wenn Sie jedoch einen benutzerdefinierten Schlüsselspeicher angeben < <https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html>>, wird die zufällige Byte-Zeichenfolge in dem AWS HSM Cloud-Cluster generiert, der dem benutzerdefinierten Schlüsselspeicher zugeordnet ist.

In diesem Beispiel werden die folgenden Parameter und Werte verwendet:

Es verwendet den erforderlichen `--number-of-bytes` Parameter mit dem Wert von, 32 um eine 32-Byte-Zeichenfolge (256-Bit) anzufordern.Es verwendet den `--output` Parameter mit dem Wert von, `text` AWS CLI um die Ausgabe als Text zurückzugeben, anstatt vonJSON. Es verwendet den, `--query` parameter um den Wert der `Plaintext` Eigenschaft aus der Antwort zu extrahieren.Es leitet (`|`) die Ausgabe des Befehls an das `base64` Hilfsprogramm weiter, das die extrahierte Ausgabe dekodiert. Es verwendet den Umleitungsoperator (`>`), um die dekodierte Bytezeichenfolge in der `ExampleRandom` Datei zu speichern.IT verwendet die Umleitungsoperator (`>`), um den binären Chiffretext in einer Datei zu speichern.

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext | base64 --decode > ExampleRandom
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [GenerateRandom](#) in der AWS Key Management Service Reference. API

Beispiel 2: Generieren einer 256-Bit-Zufallszahl (Windows-Eingabeaufforderung)

Im folgenden Beispiel wird der `generate-random` Befehl verwendet, um eine Base64-kodierte 256-Bit- (32-Byte) -Bit-Zufallsbytezeichenfolge zu generieren. Das Beispiel dekodiert die Bytezeichenfolge und speichert sie in der Zufallsdatei. Dieses Beispiel entspricht dem vorherigen Beispiel, außer dass es das `certutil` Hilfsprogramm in Windows verwendet, um die zufällige Bytezeichenfolge base64-dekodieren zu lassen, bevor sie in einer Datei gespeichert wird.

Generieren Sie zunächst eine Base64-kodierte Zufallsbytezeichenfolge und speichern Sie sie in einer temporären Datei, `ExampleRandom.base64`

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext > ExampleRandom.base64
```

Da die Ausgabe des `generate-random` Befehls in einer Datei gespeichert wird, erzeugt dieses Beispiel keine Ausgabe.

Verwenden Sie nun den `certutil -decode` Befehl, um die Base64-kodierte Bytezeichenfolge in der Datei zu dekodieren. `ExampleRandom.base64` Anschließend wird die dekodierte Bytezeichenfolge in der Datei gespeichert. `ExampleRandom`

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Ausgabe:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Weitere Informationen finden Sie [GenerateRandom](#) in der APIReferenz zum AWS Key Management Service.

- APIEinzelheiten finden Sie [GenerateRandom](#) unter AWS CLI Befehlsreferenz.

get-key-policy

Das folgende Codebeispiel zeigt die Verwendung `get-key-policy`.

AWS CLI

Um eine Schlüsselrichtlinie von einem Schlüssel auf einen anderen KMS Schlüssel zu kopieren

Im folgenden `get-key-policy` Beispiel wird die Schlüsselrichtlinie von einem KMS Schlüssel abgerufen und in einer Textdatei gespeichert. Anschließend wird die Richtlinie eines anderen KMS Schlüssels ersetzt, wobei die Textdatei als Richtlinieneingabe verwendet wird.

Da für den `--policy` Parameter von eine Zeichenfolge `put-key-policy` erforderlich ist, müssen Sie die `--output text` Option verwenden, um die Ausgabe als Textzeichenfolge anstatt als zurückzugebenJSON.

```
aws kms get-key-policy \  
  --policy-name default \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --query Policy \  
  --output text > policy.txt  
  
aws kms put-key-policy \  
  --policy-name default \  
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --policy file://policy.txt
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [PutKeyPolicy](#) in der AWS KMSAPIReferenz.

- APIEinzelheiten finden Sie [GetKeyPolicy](#) in der AWS CLI Befehlsreferenz.

get-key-rotation-status

Das folgende Codebeispiel zeigt die Verwendung `get-key-rotation-status`.

AWS CLI

Um den Rotationsstatus für einen KMS Schlüssel abzurufen.

Das folgende `get-key-rotation-status` Beispiel gibt Informationen über den Rotationsstatus des angegebenen KMS Schlüssels zurück, einschließlich der Frage, ob die automatische Rotation aktiviert ist, den Rotationszeitraum und das nächste geplante Rotationsdatum. Sie können

diesen Befehl für vom Kunden verwaltete KMS Schlüssel und AWS verwaltete KMS Schlüssel verwenden. Alle AWS verwalteten KMS Schlüssel werden jedoch jedes Jahr automatisch rotiert.

```
aws kms get-key-rotation-status \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyRotationEnabled": true,  
  "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",  
  "RotationPeriodInDays": 365  
}
```

Weitere Informationen finden Sie unter [Rotation von Schlüsseln](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [GetKeyRotationStatus](#) unter AWS CLI Befehlsreferenz.

get-parameters-for-import

Das folgende Codebeispiel zeigt die Verwendung `get-parameters-for-import`.

AWS CLI

Um die Elemente abzurufen, die zum Importieren von Schlüsselmaterial in einen KMS Schlüssel erforderlich sind

Im folgenden `get-parameters-for-import` Beispiel werden der öffentliche Schlüssel und das Import-Token abgerufen, die Sie benötigen, um Schlüsselmaterial in einen KMS Schlüssel zu importieren. Achten Sie bei der Verwendung des `import-key-material` Befehls darauf, dass Sie das Import-Token und das mit dem öffentlichen Schlüssel verschlüsselte Schlüsselmaterial verwenden, die im selben `get-parameters-for-import` Befehl zurückgegeben wurden. Außerdem muss es sich bei dem in diesem Befehl angegebenen Wrapping-Algorithmus um einen Algorithmus handeln, den Sie verwenden, um das Schlüsselmaterial mit dem öffentlichen Schlüssel zu verschlüsseln.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird eine Schlüssel-ID verwendet, aber Sie können ARN in diesem Befehl eine Schlüssel-ID oder einen Schlüssel verwenden.

```
aws kms get-parameters-for-import \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --wrapping-algorithm RSAES_OAEP_SHA_256 \  
  --wrapping-key-spec RSA_2048
```

Ausgabe:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "PublicKey": "<public key base64 encoded data>",  
  "ImportToken": "<import token base64 encoded data>",  
  "ParametersValidTo": 1593893322.32  
}
```

Weitere Informationen finden [Sie unter Herunterladen des öffentlichen Schlüssels und Importieren des Tokens](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [GetParametersForImport](#) in der AWS CLI Befehlsreferenz.

get-public-key

Das folgende Codebeispiel zeigt die Verwendung `get-public-key`.

AWS CLI

Beispiel 1: Um den öffentlichen Schlüssel eines asymmetrischen KMS Schlüssels herunterzuladen

Im folgenden `get-public-key` Beispiel wird der öffentliche Schlüssel eines asymmetrischen KMS Schlüssels heruntergeladen.

Zusätzlich zur Rückgabe des öffentlichen Schlüssels enthält die Ausgabe Informationen, die Sie benötigen, um den öffentlichen Schlüssel sicher außerhalb zu verwenden AWS KMS, einschließlich der Schlüsselverwendung und der unterstützten Verschlüsselungsalgorithmen.

```
aws kms get-public-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "PublicKey": "jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA15epvg1/
QtJhxSi2g9SDEVg8QV/...",
  "CustomerMasterKeySpec": "RSA_4096",
  "KeyUsage": "ENCRYPT_DECRYPT",
  "EncryptionAlgorithms": [
    "RSAES_OAEP_SHA_1",
    "RSAES_OAEP_SHA_256"
  ]
}
```

Weitere Informationen zur Verwendung asymmetrischer KMS Schlüssel finden Sie unter [AWS KMS Using symmetric and Asymmetric Keys](#) in der AWS Key Management Service Reference. API

Beispiel 2: Um einen öffentlichen Schlüssel in ein DER Format zu konvertieren (Linux und macOS)

Im folgenden `get-public-key` Beispiel wird der öffentliche Schlüssel eines asymmetrischen KMS Schlüssels heruntergeladen und in einer DER Datei gespeichert.

Wenn Sie den `get-public-key` Befehl in verwenden AWS CLI, gibt er einen mit DER -codierten öffentlichen X.509-Schlüssel zurück, der Base64-kodiert ist. In diesem Beispiel wird der Wert der Eigenschaft als Text abgerufen. `PublicKey` Es dekodiert das `PublicKey` Base64- und speichert es in der Datei. `public_key.der` Der output Parameter gibt die Ausgabe als Text zurück, anstatt als JSON. Der `--query` Parameter ruft nur die `PublicKey` Eigenschaft ab, nicht die Eigenschaften, die Sie benötigen, um den öffentlichen Schlüssel sicher außerhalb zu verwenden AWS KMS.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto.

```
aws kms get-public-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text \
  --query PublicKey | base64 --decode > public_key.der
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zur Verwendung asymmetrischer KMS Schlüssel finden Sie unter [Verwenden symmetrischer und asymmetrischer Schlüssel](#) in der AWS Key Management Service-Referenz. AWS KMS API

- API-Einheiten finden Sie unter Befehlsreferenz [GetPublicKey](#).AWS CLI

import-key-material

Das folgende Codebeispiel zeigt die Verwendung `import-key-material`.

AWS CLI

Um Schlüsselmaterial in einen KMS Schlüssel zu importieren

Im folgenden `import-key-material` Beispiel wird Schlüsselmaterial in einen KMS Schlüssel hochgeladen, der ohne Schlüsselmaterial erstellt wurde. Der Schlüsselstatus des KMS Schlüssels muss sein. `PendingImport`

Dieser Befehl verwendet Schlüsselmaterial, das Sie mit dem öffentlichen Schlüssel verschlüsselt haben, den der `get-parameters-for-import` Befehl zurückgegeben hat. Außerdem wird das Import-Token aus demselben `get-parameters-for-import` Befehl verwendet.

Der `expiration-model` Parameter gibt an, dass das Schlüsselmaterial automatisch an dem vom `valid-to` Parameter angegebenen Datum und der Uhrzeit abläuft. Wenn das Schlüsselmaterial abläuft und das Schlüsselmaterial AWS KMS gelöscht wird, ändert sich der Schlüsselstatus des KMS Schlüssels `Pending import` und der KMS Schlüssel wird unbrauchbar. Um den KMS Schlüssel wiederherzustellen, müssen Sie dasselbe Schlüsselmaterial erneut importieren. Um anderes Schlüsselmaterial zu verwenden, müssen Sie einen neuen KMS Schlüssel erstellen.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID oder einen gültigen Schlüssel ARN aus Ihrem AWS Konto.

```
aws kms import-key-material \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
  --import-token fileb://ImportToken.bin \
  --expiration-model KEY_MATERIAL_EXPIRES \
  --valid-to 2021-09-21T19:00:00Z
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zum Importieren von Schlüsselmaterial finden Sie unter [Importieren von Schlüsselmaterial](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [ImportKeyMaterial](#) unter AWS CLI Befehlsreferenz.

list-aliases

Das folgende Codebeispiel zeigt die Verwendung `list-aliases`.

AWS CLI

Beispiel 1: Um alle Aliase in einem AWS Konto und einer Region aufzulisten

Im folgenden Beispiel wird der `list-aliases` Befehl verwendet, um alle Aliase in der Standardregion des AWS Kontos aufzulisten. Die Ausgabe umfasst Aliase, die AWS verwalteten KMS Schlüsseln und KMS kundenverwalteten Schlüsseln zugeordnet sind.

```
aws kms list-aliases
```

Ausgabe:

```
{
  "Aliases": [
    {
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",
      "AliasName": "alias/testKey",
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",
      "AliasName": "alias/FinanceDept",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
    },
    {
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
      "AliasName": "alias/aws/dynamodb",
      "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    },
    {
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
      "AliasName": "alias/aws/ebs",
      "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
    }
  ]
}
```

```
    },  
    ...  
  ]  
}
```

Beispiel 2: Um alle Aliase für einen bestimmten Schlüssel aufzulisten KMS

Im folgenden Beispiel werden der `list-aliases` Befehl und sein `key-id` Parameter verwendet, um alle Aliase aufzulisten, die einem bestimmten KMS Schlüssel zugeordnet sind.

Jeder Alias ist nur einem KMS Schlüssel zugeordnet, aber ein KMS Schlüssel kann mehrere Aliase haben. Dieser Befehl ist sehr nützlich, da die AWS KMS Konsole für jeden KMS Schlüssel nur einen Alias auflistet. Um alle Aliase für einen KMS Schlüssel zu finden, müssen Sie den `list-aliases` Befehl verwenden.

In diesem Beispiel wird die Schlüssel-ID des KMS Schlüssels für den `--key-id` Parameter verwendet, aber Sie können ARN in diesem Befehl eine Schlüssel-ID, einen SchlüsselARN, einen Aliasnamen oder einen Alias verwenden.

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{  
  "Aliases": [  
    {  
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-key",  
      "AliasName": "alias/oregon-test-key"  
    },  
    {  
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-test",  
      "AliasName": "alias/project121-test"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Aliasen](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie unter [ListAliases AWS CLI Befehlsreferenz](#).

list-grants

Das folgende Codebeispiel zeigt die Verwendung `list-grants`.

AWS CLI

Um die Zuschüsse auf einem AWS KMS Schlüssel anzuzeigen

Im folgenden `list-grants` Beispiel werden alle Grants für den angegebenen AWS verwalteten KMS Schlüssel für Amazon DynamoDB in Ihrem Konto angezeigt. Dieser Zuschuss ermöglicht es DynamoDB, den KMS Schlüssel in Ihrem Namen zu verwenden, um eine DynamoDB-Tabelle zu verschlüsseln, bevor sie auf die Festplatte geschrieben wird. Sie können einen Befehl wie diesen verwenden, um die Zuschüsse für die verwalteten Schlüssel und die vom Kunden AWS verwalteten KMS Schlüssel im Konto und in der KMS Region anzuzeigen. AWS

Dieser Befehl verwendet den `key-id` Parameter mit einer Schlüssel-ID, um den KMS Schlüssel zu identifizieren. Sie können eine Schlüssel-ID oder einen Schlüssel verwenden ARN, um den KMS Schlüssel zu identifizieren. Verwenden Sie den `list-aliases` Befehl oder, um die Schlüssel-ID oder den KMS `list-keys` Schlüssel ARN eines AWS verwalteten Schlüssels abzurufen.

```
aws kms list-grants \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Die Ausgabe zeigt, dass Amazon DynamoDB durch den Zuschuss die Erlaubnis erhält, den KMS Schlüssel für kryptografische Operationen zu verwenden und Details über den KMS Schlüssel (`DescribeKey`) einzusehen und Grants zurückzuziehen (`RetireGrant`). Die `EncryptionContextSubset` Einschränkung beschränkt diese Berechtigungen auf Anfragen, die die angegebenen Verschlüsselungskontextpaare enthalten. Daher sind die Berechtigungen in der Gewährung nur für das angegebene Konto und die angegebene DynamoDB-Tabelle wirksam.

```
{  
  "Grants": [  
    {  
      "Constraints": {  
        "EncryptionContextSubset": {  
          "aws:dynamodb:subscriberId": "123456789012",  
          "aws:dynamodb:tableName": "Services"  
        }  
      }  
    }  
  ]  
}
```

```

    }
  },
  "IssuingAccount": "arn:aws:iam::123456789012:root",
  "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
  "Operations": [
    "Decrypt",
    "Encrypt",
    "GenerateDataKey",
    "ReEncryptFrom",
    "ReEncryptTo",
    "RetireGrant",
    "DescribeKey"
  ],
  "GrantId":
"1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
  "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
  "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
  "CreationDate": "2021-05-13T18:32:45.144000+00:00"
}
]
}

```

Weitere Informationen finden Sie unter [Grants AWS KMS im AWS](#) Key Management Service Developer Guide.

- API-Einheiten finden Sie [ListGrants](#) in der AWS CLI Befehlsreferenz.

list-key-policies

Das folgende Codebeispiel zeigt die Verwendung `list-key-policies`.

AWS CLI

Um die Namen der wichtigsten Richtlinien für einen KMS Schlüssel abzurufen

Im folgenden `list-key-policies` Beispiel werden die Namen der wichtigsten Richtlinien für einen vom Kunden verwalteten Schlüssel im Beispiellokonto und in der Region abgerufen. Sie können diesen Befehl verwenden, um die Namen der wichtigsten Richtlinien für AWS verwaltete Schlüssel und vom Kunden verwaltete Schlüssel zu finden.

Da der einzig gültige Name der Schlüsselrichtlinie lautet `default`, ist dieser Befehl nicht nützlich.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird ein Schlüssel-ID-Wert verwendet, aber Sie können ARN in diesem Befehl auch eine Schlüssel-ID oder einen Schlüssel verwenden.

```
aws kms list-key-policies \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{  
  "PolicyNames": [  
    "default"  
  ]  
}
```

Weitere Informationen zu AWS KMS wichtigen Richtlinien finden Sie unter [Verwenden von Schlüsselrichtlinien AWS KMS im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie [ListKeyPolicies](#) in der AWS CLI Befehlsreferenz.

list-key-rotations

Das folgende Codebeispiel zeigt die Verwendung `list-key-rotations`.

AWS CLI

Um Informationen über alle abgeschlossenen wichtigen Materialrotationen abzurufen

Im folgenden `list-key-rotations` Beispiel werden Informationen zu allen abgeschlossenen Materialrotationen für den angegebenen KMS Schlüssel aufgeführt.

```
aws kms list-key-rotations \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{  
  "Rotations": [  
    {  
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
      "RotationDate": "2024-03-02T10:11:36.564000+00:00",  
      "RotationType": "AUTOMATIC"  
    }  
  ]  
}
```

```
    },
    {
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "RotationDate": "2024-04-05T15:14:47.757000+00:00",
      "RotationType": "ON_DEMAND"
    }
  ],
  "Truncated": false
}
```

Weitere Informationen finden Sie unter [Rotation von Schlüsseln](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [ListKeyRotations](#) unter AWS CLI Befehlsreferenz.

list-keys

Das folgende Codebeispiel zeigt die Verwendung `list-keys`.

AWS CLI

Um die KMS Schlüssel für ein Konto und eine Region zu erhalten

Im folgenden `list-keys` Beispiel werden die KMS Schlüssel für ein Konto und eine Region abgerufen. Dieser Befehl gibt sowohl AWS verwaltete Schlüssel als auch vom Kunden verwaltete Schlüssel zurück.

```
aws kms list-keys
```

Ausgabe:

```
{
  "Keys": [
    {
      "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
    }
  ],
}
```

```
{
  "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
  "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
}
]
```

Weitere Informationen finden Sie unter [Schlüssel anzeigen](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [ListKeys](#) in der AWS CLI Befehlsreferenz.

list-resource-tags

Das folgende Codebeispiel zeigt die Verwendung `list-resource-tags`.

AWS CLI

Um die Tags auf einem KMS Schlüssel abzurufen

Im folgenden `list-resource-tags` Beispiel werden die Tags für einen KMS Schlüssel abgerufen. Verwenden Sie den `tag-resource` Befehl, um Ressourcen-Tags zu KMS Schlüsseln hinzuzufügen oder zu ersetzen. Die Ausgabe zeigt, dass dieser KMS Schlüssel zwei Ressourcen-Tags hat, von denen jedes einen Schlüssel und einen Wert hat.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird ein Schlüssel-ID-Wert verwendet, aber Sie können ARN in diesem Befehl auch eine Schlüssel-ID oder einen Schlüssel verwenden.

```
aws kms list-resource-tags \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{
  "Tags": [
    {
      "TagKey": "Dept",
      "TagValue": "IT"
    },
    {
```

```
    "TagKey": "Purpose",
    "TagValue": "Test"
  }
],
"Truncated": false
}
```

Weitere Informationen zur Verwendung von Tags in AWS KMS finden Sie unter [Tagging Keys](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [ListResourceTags](#) in der AWS CLI Befehlsreferenz.

list-retirable-grants

Das folgende Codebeispiel zeigt die Verwendung `list-retirable-grants`.

AWS CLI

Um die Zuschüsse anzuzeigen, die ein Schulleiter in den Ruhestand schicken kann

Im folgenden `list-retirable-grants` Beispiel werden alle Zuschüsse angezeigt, die der `ExampleAdmin` Benutzer mit den KMS Schlüsseln eines AWS Kontos und einer Region zurückziehen kann. Sie können einen Befehl wie diesen verwenden, um die Zuschüsse anzuzeigen, die jeder Kontoinhaber aufgrund von KMS Schlüsseln für das AWS Konto und die Region zurückziehen kann.

Der Wert des erforderlichen `retiring-principal` Parameters muss der Amazon-Ressourcenname (ARN) eines Kontos, Benutzers oder einer Rolle sein.

Sie können `retiring-principal` in diesem Befehl keinen Service für den Wert von angeben, auch wenn ein Service der ausscheidende Principal sein kann. Verwenden Sie den Befehl, um die Zuschüsse zu ermitteln, in denen ein bestimmter Dienst der ausscheidende Schulleiter ist. `list-grants`

Die Ausgabe zeigt, dass der `ExampleAdmin` Benutzer berechtigt ist, Grants für zwei verschiedene KMS Schlüssel im Konto und in der Region zurückzuziehen. Zusätzlich zum ausscheidenden Hauptbetrag ist das Konto berechtigt, alle Zuschüsse auf dem Konto zurückzuziehen.

```
aws kms list-retirable-grants \  
  --retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

Ausgabe:

```
{
  "Grants": [
    {
      "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "GrantId":
"156b69c63cb154aa21f59929ff19760717be8d9d82b99df53e18b94a15a5e88e",
      "Name": "",
      "CreationDate": 2021-01-14T20:17:36.419000+00:00,
      "GranteePrincipal": "arn:aws:iam::111122223333:user/ExampleUser",
      "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "Operations": [
        "Encrypt"
      ],
      "Constraints": {
        "EncryptionContextSubset": {
          "Department": "IT"
        }
      }
    },
    {
      "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "GrantId":
"8c94d1f12f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
      "Name": "",
      "CreationDate": "2021-02-02T19:49:49.638000+00:00",
      "GranteePrincipal": "arn:aws:iam::111122223333:role/ExampleRole",
      "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "Operations": [
        "Decrypt"
      ],
      "Constraints": {
        "EncryptionContextSubset": {
          "Department": "IT"
        }
      }
    }
  ],
  "Truncated": false
}
```

```
}
```

Weitere Informationen finden Sie unter [Grants AWS KMS im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie [ListRetirableGrants](#) in der AWS CLI Befehlsreferenz.

put-key-policy

Das folgende Codebeispiel zeigt die Verwendung `put-key-policy`.

AWS CLI

Um die Schlüsselrichtlinie für einen KMS Schlüssel zu ändern

Im folgenden `put-key-policy` Beispiel wird die Schlüsselrichtlinie für einen vom Kunden verwalteten Schlüssel geändert.

Erstellen Sie zunächst eine Schlüsselrichtlinie und speichern Sie sie in einer lokalen JSON Datei. In diesem Beispiel ist die Datei `key_policy.json`. Sie können die Schlüsselrichtlinie auch als Zeichenfolgenwert des `policy` Parameters angeben.

Die erste Anweisung in dieser Schlüsselrichtlinie erteilt dem AWS Konto die Erlaubnis, IAM Richtlinien zur Steuerung des Zugriffs auf den KMS Schlüssel zu verwenden. Die zweite Anweisung erteilt dem `test-user` Benutzer die Erlaubnis, die `list-keys` Befehle `describe-key` und auf dem KMS Schlüssel auszuführen.

Inhalt von `key_policy.json`:

```
{
  "Version" : "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [
    {
      "Sid" : "Enable IAM User Permissions",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : "kms:*",
      "Resource" : "*"
    },
    {
```



```

    "Sid" : "Allow Use of Key",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:user/test-user"
    },
    "Action" : [
      "kms:DescribeKey",
      "kms:ListKeys"
    ],
    "Resource" : "*"
  }
]
}

```

Um den KMS Schlüssel zu identifizieren, verwendet dieses Beispiel die Schlüssel-ID, Sie können aber auch einen Schlüssel verwendenARN. Um die Schlüsselrichtlinie anzugeben, verwendet der Befehl den `policy` Parameter. Um anzugeben, dass sich die Richtlinie in einer Datei befindet, wird das erforderliche `file://` Präfix verwendet. Dieses Präfix ist erforderlich, um Dateien auf allen unterstützten Betriebssystemen zu identifizieren. Schließlich verwendet der Befehl den `policy-name` Parameter mit dem Wert `default`. Wenn kein Richtlinienname angegeben ist, ist der Standardwert `default`. Der einzige gültige Wert ist `default`.

```

aws kms put-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --policy file://key_policy.json

```

Dieser Befehl erzeugt keine Ausgabe. Verwenden Sie den Befehl, um zu überprüfen, ob der `get-key-policy` Befehl wirksam war. Mit dem folgenden Beispielbefehl wird die Schlüsselrichtlinie für denselben KMS Schlüssel abgerufen. Der `output` Parameter mit dem Wert von `text` gibt ein Textformat zurück, das leicht zu lesen ist.

```

aws kms get-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text

```

Ausgabe:

```

{
  "Version" : "2012-10-17",

```

```

    "Id" : "key-default-1",
    "Statement" : [
      {
        "Sid" : "Enable IAM User Permissions",
        "Effect" : "Allow",
        "Principal" : {
          "AWS" : "arn:aws:iam::111122223333:root"
        },
        "Action" : "kms:*",
        "Resource" : "*"
      },
      {
        "Sid" : "Allow Use of Key",
        "Effect" : "Allow",
        "Principal" : {
          "AWS" : "arn:aws:iam::111122223333:user/test-user"
        },
        "Action" : [ "kms:Describe", "kms:List" ],
        "Resource" : "*"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Ändern einer Schlüsselrichtlinie](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [PutKeyPolicy](#) unter AWS CLI Befehlsreferenz.

re-encrypt

Das folgende Codebeispiel zeigt die Verwendung `re-encrypt`.

AWS CLI

Beispiel 1: Um eine verschlüsselte Nachricht unter einem anderen symmetrischen KMS Schlüssel (Linux und macOS) erneut zu verschlüsseln.

Das folgende `re-encrypt` Befehlsbeispiel zeigt die empfohlene Methode zum erneuten Verschlüsseln von Daten mit dem AWS CLI

Geben Sie den Geheimtext in einer Datei an. Verwenden Sie im Wert des `--ciphertext-blob` Parameters das `fileb://` Präfix, das angibt, die Daten aus einer Binärdatei CLI zu lesen. Wenn sich die Datei nicht im aktuellen Verzeichnis befindet, geben Sie den vollständigen Dateipfad ein.

Weitere Informationen zum Lesen von AWS CLI Parameterwerten aus einer Datei finden Sie unter [Loading AWS CLI parameters from a file < https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file .html>](https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html) im AWS Command Line Interface User Guide und [Best Practices for Local File Parameters< https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>](https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/) im AWS Command Line Tool Blog .Geben Sie den KMS Quellschlüssel an, der den Chiffretext entschlüsselt. Der `--source-key-id` Parameter ist bei der Entschlüsselung mit symmetrischer Verschlüsselung nicht erforderlich. `KMSSchlüssel`. AWS KMS kann den KMS Schlüssel, der zur Verschlüsselung der Daten verwendet wurde, aus den Metadaten im Chiffretext-Blob abrufen. Es hat sich jedoch immer bewährt, den KMS Schlüssel anzugeben, den Sie verwenden. Diese Vorgehensweise stellt sicher, dass Sie den KMS Schlüssel verwenden, den Sie beabsichtigen, und verhindert, dass Sie versehentlich einen Chiffretext mit einem KMS Schlüssel entschlüsseln, dem Sie nicht vertrauen. Geben Sie den KMS Zielschlüssel an, mit dem die Daten erneut verschlüsselt werden. Der Parameter ist immer erforderlich. `--destination-key-id` In diesem Beispiel wird ein Schlüssel verwendet `ARN`, Sie können jedoch jeden gültigen Schlüsselbezeichner verwenden. Fordern Sie die Klartextausgabe als Textwert an. Der Parameter weist an, nur den Wert des Felds aus der Ausgabe abzurufen. `--query CLI Plaintext` Der `--output` Parameter gibt die Ausgabe als Text zurück. Base64-dekodieren Sie den Klartext und speichern Sie ihn in einer Datei. Im folgenden Beispiel wird der Wert des Parameters (`()`) an das Base64-Hilfsprogramm übergeben, das ihn dekodiert. `Plaintext` Anschließend leitet er die dekodierte Ausgabe in die Datei um (`>`). `ExamplePlaintext`

Bevor Sie diesen Befehl ausführen, ersetzen Sie den Beispielschlüssel IDs durch gültige Schlüsselkennungen aus Ihrem AWS Konto.

```
aws kms re-encrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
  --query CiphertextBlob \
  --output text | base64 --decode > ExampleReEncryptedFile
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Die Ausgabe des `re-encrypt` Befehls wird base64-dekodiert und in einer Datei gespeichert.

Weitere Informationen finden Sie unter [ReEncrypt < https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt .html](https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html) in der AWS Key Management Service Reference. `API`

Beispiel 2: Um eine verschlüsselte Nachricht unter einem anderen symmetrischen KMS Schlüssel erneut zu verschlüsseln (Windows-Eingabeaufforderung).

Das folgende `re-encrypt` Befehlsbeispiel ist dasselbe wie das vorherige, außer dass es das `certutil` Hilfsprogramm zur Base64-Decodierung der Klartextdaten verwendet. Für dieses Verfahren sind zwei Befehle erforderlich, wie in den folgenden Beispielen gezeigt.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto.

```
aws kms re-encrypt ^
  --ciphertext-blob fileb://ExampleEncryptedFile ^
  --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
  --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^
  --query CiphertextBlob ^
  --output text > ExampleReEncryptedFile.base64
```

Verwenden Sie dann das `certutil` Hilfsprogramm

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Ausgabe:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Weitere Informationen finden Sie unter `ReEncrypt` < https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html in der AWS Key Management Service API Reference.

- API-Einheiten finden Sie [ReEncrypt](#) in der AWS CLI Befehlsreferenz.

retire-grant

Das folgende Codebeispiel zeigt die Verwendung `retire-grant`.

AWS CLI

Um einen Zuschuss für einen Kundenhauptschlüssel zurückzuziehen

Im folgenden `retire-grant` Beispiel wird ein Zuschuss aus einem KMS Schlüssel gelöscht.

Der folgende Beispielbefehl spezifiziert die `key-id` Parameter `grant-id` und. Der Wert des `key-id` Parameters muss der Schlüssel ARN des KMS Schlüssels sein.

```
aws kms retire-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `list-grants` Befehl, um zu bestätigen, dass der Zuschuss zurückgezogen wurde.

Weitere Informationen finden Sie im AWS Key Management Service Developer Guide unter [Zurückziehen und Widerrufen von Zuschüssen](#).

- API-Einheiten finden Sie unter [RetireGrant AWS CLI](#) Befehlsreferenz.

revoke-grant

Das folgende Codebeispiel zeigt die Verwendung `revoke-grant`.

AWS CLI

Um eine Erteilung für einen Kundenhauptschlüssel zu widerrufen

Im folgenden `revoke-grant` Beispiel wird ein Zuschuss aus einem KMS Schlüssel gelöscht. Der folgende Beispielbefehl spezifiziert die `key-id` Parameter `grant-id` und. Der Wert des `key-id` Parameters kann die Schlüssel-ID oder der Schlüssel ARN des KMS Schlüssels sein.

```
aws kms revoke-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `list-grants` Befehl, um zu bestätigen, dass der Zuschuss widerrufen wurde.

Weitere Informationen finden Sie im AWS Key Management Service Developer Guide unter [Zurückziehen und Widerrufen von Zuschüssen](#).

- API-Einheiten finden Sie unter [RevokeGrant AWS CLI](#) Befehlsreferenz.

rotate-key-on-demand

Das folgende Codebeispiel zeigt die Verwendung `rotate-key-on-demand`.

AWS CLI

So führen Sie die Rotation eines KMS Schlüssels bei Bedarf durch

Im folgenden `rotate-key-on-demand` Beispiel wird sofort die Rotation des Schlüsselmaterials für den angegebenen KMS Schlüssel initiiert.

```
aws kms rotate-key-on-demand \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Ausgabe:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Weitere Informationen finden Sie unter [So führen Sie die Schlüsselrotation bei Bedarf durch](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [RotateKeyOnDemand](#) unter AWS CLI Befehlsreferenz.

schedule-key-deletion

Das folgende Codebeispiel zeigt die Verwendung `schedule-key-deletion`.

AWS CLI

Um das Löschen eines vom Kunden verwalteten KMS Schlüssels zu planen.

Im folgenden `schedule-key-deletion` Beispiel wird geplant, dass der angegebene vom Kunden verwaltete KMS Schlüssel innerhalb von 15 Tagen gelöscht wird.

Der `--key-id` Parameter identifiziert den KMS Schlüssel. In diesem Beispiel wird ein ARN Schlüsselwert verwendet, Sie können jedoch entweder die Schlüssel-ID oder die ARN des Schlüssels verwenden. Der `--pending-window-in-days` Parameter gibt KMS die Länge der Wartezeit von 7 bis 30 Tagen an. Standardmäßig beträgt die Wartezeit 30 Tage. In diesem Beispiel wird der Wert 15 angegeben, der angibt, dass der KMS Schlüssel 15 Tage nach Abschluss des Befehls dauerhaft gelöscht werden AWS soll.

```
aws kms schedule-key-deletion \  
  --key-id arn:aws:kms:us-east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab  
  --pending-window-in-days 15
```

```
--key-id arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
--pending-window-in-days 15
```

Die Antwort umfasst den SchlüsselARN, den Schlüsselstatus, die Wartezeit (PendingWindowInDays) und das Löschdatum in Unix-Zeit. Verwenden Sie die AWS KMS Konsole, um das Löschdatum in Ortszeit anzuzeigen. KMS-Schlüssel im PendingDeletion Schlüsselstatus können nicht für kryptografische Operationen verwendet werden.

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "DeletionDate": "2022-06-18T23:43:51.272000+00:00",  
  "KeyState": "PendingDeletion",  
  "PendingWindowInDays": 15  
}
```

Weitere Informationen finden Sie unter [Löschen von Schlüsseln](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [ScheduleKeyDeletion](#) in der AWS CLI Befehlsreferenz.

sign

Das folgende Codebeispiel zeigt die Verwendung `sign`.

AWS CLI

Beispiel 1: Um eine digitale Signatur für eine Nachricht zu generieren

Das folgende `sign` Beispiel generiert eine kryptografische Signatur für eine Kurznachricht. Die Ausgabe des Befehls enthält ein Base-64-codiertes Signature Feld, das Sie mithilfe des Befehls `verify` überprüfen können.

Sie müssen eine zu signierende Nachricht und einen Signierungsalgorithmus angeben, den Ihr asymmetrischer KMS Schlüssel unterstützt. Verwenden Sie den `describe-key` Befehl, um die Signaturalgorithmen für Ihren KMS Schlüssel abzurufen.

In AWS CLI 2.0 muss der Wert des `message` Parameters Base64-kodiert sein. Oder Sie können die Nachricht in einer Datei speichern und das `fileb://` Präfix verwenden, das angibt, Binärdaten aus der Datei AWS CLI zu lesen.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto. Die Schlüssel-ID muss einen asymmetrischen KMS Schlüssel mit der Schlüsselverwendung SIGN_VERIFY darstellen.

```
msg=(echo 'Hello World' | base64)

aws kms sign \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://UnsignedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

Ausgabe:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Weitere Informationen zur Verwendung asymmetrischer KMS Schlüssel in AWS KMS finden Sie unter [Asymmetrische Schlüssel in AWS KMS im AWS](#) Key Management Service Developer Guide.

Beispiel 2: Um eine digitale Signatur in einer Datei zu speichern (Linux und) macOS

Das folgende `sign` Beispiel generiert eine kryptografische Signatur für eine Kurznachricht, die in einer lokalen Datei gespeichert ist. Der Befehl ruft auch die `Signature` Eigenschaft aus der Antwort ab, dekodiert sie mit Base64 und speichert sie in der Datei. `ExampleSignature` Sie können die Signaturdatei in einem `verify` Befehl verwenden, der die Signatur überprüft.

Für den `sign` Befehl sind eine Base64-codierte Nachricht und ein Signaturalgorithmus erforderlich, den Ihr asymmetrischer Schlüssel unterstützt. KMS Verwenden Sie den Befehl, um die Signaturalgorithmen abzurufen, die Ihr KMS Schlüssel unterstützt. `describe-key`

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto. Die Schlüssel-ID muss einen asymmetrischen KMS Schlüssel mit der Schlüsselverwendung SIGN_VERIFY darstellen.


```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://EncodedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
  --output text \
  --query Signature | base64 --decode > ExampleSignature
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. In diesem Beispiel wird die `Signature` Eigenschaft der Ausgabe extrahiert und in einer Datei gespeichert.

Weitere Informationen zur Verwendung asymmetrischer KMS Schlüssel in AWS KMS finden Sie unter [Asymmetrische Schlüssel in AWS KMS im AWS Key Management Service Developer Guide](#).

- API-Einheiten finden Sie unter AWS CLI Befehlsreferenz für die [Anmeldung](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einem KMS Schlüssel ein Tag hinzuzufügen

Das folgende `tag-resource` Beispiel fügt einem vom Kunden verwalteten KMS Schlüssel "Dept": "IT" Tags hinzu "Purpose": "Test" und fügt ihm Tags hinzu. Sie können solche Tags verwenden, um KMS Schlüssel zu beschriften und KMS Schlüsselkategorien für Berechtigungen und Prüfungen zu erstellen.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird ein Schlüssel-ID-Wert verwendet, aber Sie können ARN in diesem Befehl auch eine Schlüssel-ID oder einen Schlüssel verwenden.

```
aws kms tag-resource \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `list-resource-tags` Befehl, um die Tags eines AWS KMS Schlüssels anzuzeigen.

Weitere Informationen zur Verwendung von Tags finden Sie unter [Tagging Keys](#) im AWS Key Management Service Developer Guide. AWS KMS

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einem KMS Schlüssel zu löschen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem "Purpose" Schlüssel aus einem vom Kunden verwalteten KMS Schlüssel gelöscht.

Verwenden Sie den `key-id` Parameter, um den KMS Schlüssel anzugeben. In diesem Beispiel wird ein Schlüssel-ID-Wert verwendet, aber Sie können ARN in diesem Befehl auch eine Schlüssel-ID oder einen Schlüssel verwenden. Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto.

```
aws kms untag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tag-key 'Purpose'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `list-resource-tags` Befehl, um die Tags auf einem AWS KMS Schlüssel anzuzeigen.

Weitere Informationen zur Verwendung von Tags finden Sie unter [Tagging Keys](#) im AWS Key Management Service Developer Guide. AWS KMS

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-alias

Das folgende Codebeispiel zeigt die Verwendung `update-alias`.

AWS CLI

Um einen Alias einem anderen KMS Schlüssel zuzuordnen

Im folgenden `update-alias` Beispiel wird der Alias `alias/test-key` einem anderen KMS Schlüssel zugeordnet.

Der `--alias-name` Parameter gibt den Alias an. Der Wert des Aliasnamens muss mit `alias/` beginnen. Der `--target-key-id` Parameter gibt den KMS Schlüssel an, der dem Alias zugeordnet werden soll. Sie müssen den aktuellen KMS Schlüssel für den Alias nicht angeben.

```
aws kms update-alias \  
  --alias-name alias/test-key \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `list-aliases` Befehl, um den Alias zu finden.

Weitere Informationen finden Sie unter [Aliase aktualisieren](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [UpdateAlias](#) in der AWS CLI Befehlsreferenz.

update-custom-key-store

Das folgende Codebeispiel zeigt die Verwendung `update-custom-key-store`.

AWS CLI

Beispiel 1: Um den Anzeigenamen eines benutzerdefinierten Schlüsselspeichers zu bearbeiten

Im folgenden `update-custom-key-store` Beispiel wird der Name des benutzerdefinierten Schlüsselspeichers geändert. Dieses Beispiel funktioniert für einen AWS HSM Cloud-Schlüsselspeicher oder einen externen Schlüsselspeicher.

Verwenden Sie `custom-key-store-id`, um den Schlüsselspeicher zu identifizieren. Verwenden Sie den `new-custom-key-store-name` Parameter, um den neuen Anzeigenamen anzugeben.

Um den Anzeigenamen eines AWS HSM Cloud-Schlüsselspeichers zu aktualisieren, müssen Sie zuerst die Verbindung zum Schlüsselspeicher trennen, z. B. mit dem `disconnect-`

`custom-key-store` Befehl. Sie können den Anzeigenamen eines externen Schlüsselspeichers aktualisieren, während er verbunden oder getrennt ist. Verwenden Sie den `describe-custom-key-store` Befehl, um den Verbindungsstatus Ihres benutzerdefinierten Schlüsselspeichers zu ermitteln.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --new-custom-key-store-name ExampleKeyStore
```

Dieser Befehl gibt keine Daten zurück. Verwenden Sie einen Befehl, um zu überprüfen, ob der `describe-custom-key-stores` Befehl funktioniert hat.

Weitere Informationen zum Aktualisieren eines AWS HSM Cloud-Schlüsselspeichers finden Sie unter [Bearbeiten von AWS HSM Cloud-Schlüsselspeichereinstellungen](#) im AWS Key Management Service Developer Guide.

Weitere Informationen zum Aktualisieren eines externen Schlüsselspeichers finden Sie unter [Bearbeiten der Eigenschaften eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

Beispiel 2: So bearbeiten Sie das `kmsuser`-Passwort eines AWS Cloud-Schlüsselspeichers HSM

Im folgenden `update-custom-key-store` Beispiel wird der Wert des `kmsuser` Kennworts auf das aktuelle Passwort für den `kmsuser` HSM Cloud-Cluster aktualisiert, der dem angegebenen Schlüsselspeicher zugeordnet ist. Dieser Befehl ändert das `kmsuser` Passwort im Cluster nicht. Es gibt nur AWS KMS das aktuelle Passwort an. Wenn KMS das aktuelle `kmsuser` Passwort nicht vorhanden ist, kann es keine Verbindung zum AWS Cloud HSM Key Store herstellen.

NOTE: Bevor Sie einen AWS Cloud HSM Key Store aktualisieren, müssen Sie die Verbindung trennen. Verwenden Sie den `disconnect-custom-key-store`-Befehl. Nachdem der Befehl abgeschlossen ist, können Sie den AWS HSM Cloud-Schlüsselspeicher erneut verbinden. Verwenden Sie den `connect-custom-key-store`-Befehl.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --key-store-password ExamplePassword
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie einen `describe-custom-key-stores` Befehl, um zu überprüfen, ob die Änderung wirksam war.

Weitere Informationen zum Aktualisieren eines AWS HSM Cloud-Schlüsselspeichers finden Sie unter [Bearbeiten von AWS HSM Cloud-Schlüsselspeicher-Einstellungen](#) im AWS Key Management Service Developer Guide.

Beispiel 3: So bearbeiten Sie den AWS HSM Cloud-Cluster eines AWS HSM Cloud-Schlüsselspeichers

Im folgenden Beispiel wird der AWS HSM Cloud-Cluster, der einem AWS HSM Cloud-Schlüsselspeicher zugeordnet ist, in einen verwandten Cluster geändert, z. B. in ein anderes Backup desselben Clusters.

NOTE: Bevor Sie einen AWS HSM Cloud-Schlüsselspeicher aktualisieren, müssen Sie die Verbindung trennen. Verwenden Sie den `disconnect-custom-key-store`-Befehl. Nachdem der Befehl abgeschlossen ist, können Sie den AWS HSM Cloud-Schlüsselspeicher erneut verbinden. Verwenden Sie den `connect-custom-key-store`-Befehl.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie einen `describe-custom-key-stores` Befehl, um zu überprüfen, ob die Änderung wirksam war.

Weitere Informationen zum Aktualisieren eines AWS HSM Cloud-Schlüsselspeichers finden Sie unter [Bearbeiten von AWS HSM Cloud-Schlüsselspeicher-Einstellungen](#) im AWS Key Management Service Developer Guide.

Beispiel 4: So bearbeiten Sie die Anmeldeinformationen für die Proxyauthentifizierung eines externen Schlüsselspeichers

Im folgenden Beispiel werden die Anmeldeinformationen für die Proxyauthentifizierung für Ihren externen Schlüsselspeicher aktualisiert. Sie müssen `raw-secret-access-key` sowohl den als auch den `access-key-id`, auch wenn Sie nur einen der Werte ändern. Sie können diese Funktion verwenden, um ungültige Anmeldeinformationen zu korrigieren oder um die Anmeldeinformationen zu ändern, wenn der externe Schlüsselspeicher-Proxy sie wechselt.

Richten Sie die Anmeldeinformationen für die Proxyauthentifizierung in AWS KMS Ihrem externen Schlüsselspeicher ein. Verwenden Sie dann diesen Befehl, um die Anmeldeinformationen für bereitzustellen. AWS KMS verwendet diese Anmeldeinformationen, um seine Anfragen an Ihren externen Schlüsselspeicher-Proxy zu signieren.

Sie können die Anmeldeinformationen für die Proxyauthentifizierung aktualisieren, während der externe Schlüsselspeicher verbunden oder getrennt ist. Verwenden Sie den Befehl, um den Verbindungsstatus Ihres benutzerdefinierten Schlüsselspeichers zu ermitteln. `describe-custom-key-store`

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo=""
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie einen `describe-custom-key-stores` Befehl, um zu überprüfen, ob die Änderung wirksam war.

Weitere Informationen zum Aktualisieren eines externen Schlüsselspeichers finden Sie unter [Bearbeiten der Eigenschaften eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

Beispiel 5: So bearbeiten Sie die Proxykonnektivität eines externen Schlüsselspeichers

Im folgenden Beispiel wird die Option für die Proxykonnektivität des externen Schlüsselspeichers von öffentlicher Endpunktkonnektivität zu VPC Endpunktdienstkonnektivität geändert. Sie müssen nicht nur den `xks-proxy-connectivity` Wert ändern, sondern auch den `xks-proxy-uri-endpoint` Wert so ändern, dass er den privaten DNS Namen wiedergibt, der dem VPC Endpunktdienst zugeordnet ist. Sie müssen auch einen `xks-proxy-vpc-endpoint-service-name` Wert hinzufügen.

NOTE: Bevor Sie die Proxykonnektivität eines externen Speichers aktualisieren, müssen Sie die Verbindung trennen. Verwenden Sie den `disconnect-custom-key-store`-Befehl. Nach Abschluss des Befehls können Sie den externen Schlüsselspeicher mithilfe des `connect-custom-key-store` Befehls erneut verbinden.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \  
  --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \  
  --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-  
example"
```

Dieser Befehl gibt keine Ausgabe zurück. Verwenden Sie einen `describe-custom-key-stores` Befehl, um zu überprüfen, ob die Änderung wirksam war.

Weitere Informationen zum Aktualisieren eines externen Schlüsselspeichers finden Sie unter [Bearbeiten der Eigenschaften eines externen Schlüsselspeichers](#) im AWS Key Management Service Developer Guide.

- API-Einheiten finden Sie [UpdateCustomKeyStore](#) in der AWS CLI Befehlsreferenz.

update-key-description

Das folgende Codebeispiel zeigt die Verwendung `update-key-description`.

AWS CLI

Beispiel 1: Um einem vom Kunden verwalteten KMS Schlüssel eine Beschreibung hinzuzufügen oder zu ändern

Im folgenden `update-key-description` Beispiel wird einem vom Kunden verwalteten KMS Schlüssel eine Beschreibung hinzugefügt. Sie können denselben Befehl verwenden, um eine bestehende Beschreibung zu ändern.

Der `--key-id` Parameter identifiziert den KMS Schlüssel im Befehl. In diesem Beispiel wird ein ARN Schlüsselwert verwendet, Sie können jedoch entweder die Schlüssel-ID oder den Schlüssel ARN des Schlüssels verwenden. Der `--description` Parameter gibt die neue Beschreibung an. Der Wert dieses Parameters ersetzt die aktuelle Beschreibung des KMS Schlüssels, falls vorhanden.

```
aws kms update-key-description \
  --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
  --description "IT Department test key"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Verwenden Sie den `describe-key` Befehl, um die Beschreibung eines KMS Schlüssels anzuzeigen.

Weitere Informationen finden Sie [UpdateKeyDescription](#) in der APIReferenz AWS zum Key Management Service.

Beispiel 2: So löschen Sie die Beschreibung eines vom Kunden verwalteten KMS Schlüssels

Im folgenden `update-key-description` Beispiel wird die Beschreibung eines vom Kunden verwalteten KMS Schlüssels gelöscht.

Der `--key-id` Parameter identifiziert den KMS Schlüssel im Befehl. In diesem Beispiel wird ein Schlüssel-ID-Wert verwendet, Sie können jedoch entweder die Schlüssel-ID oder den Schlüssel ARN des Schlüssels verwenden. KMS Der `--description` Parameter mit einem leeren Zeichenfolgenwert (`,`) löscht die vorhandene Beschreibung.

```
aws kms update-key-description \  
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --description ''
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um die Beschreibung eines KMS Schlüssels anzuzeigen, verwenden Sie den Befehl `describe-key`.

Weitere Informationen finden Sie [UpdateKeyDescription](#) in der Referenz zum AWS Key Management Service API.

- API-Einheiten finden Sie [UpdateKeyDescription](#) unter AWS CLI Befehlsreferenz.

verify

Das folgende Codebeispiel zeigt die Verwendung `verify`.

AWS CLI

Um eine digitale Signatur zu verifizieren

Im folgenden `verify` Beispiel wird eine kryptografische Signatur für eine kurze, Base64-kodierte Nachricht überprüft. Die Schlüssel-ID, die Nachricht, der Nachrichtentyp und der Signaturalgorithmus müssen dieselben sein, die zum Signieren der Nachricht verwendet wurden. Die von Ihnen angegebene Signatur kann nicht Base64-codiert sein. Hilfe zur Dekodierung der Signatur, die der `sign` Befehl zurückgibt, finden Sie in den Befehlsbeispielen. `sign`

Die Ausgabe des Befehls enthält ein boolesches `SignatureValid` Feld, das angibt, dass die Signatur verifiziert wurde. Wenn die Signaturüberprüfung fehlschlägt, schlägt auch der `verify` Befehl fehl.

Bevor Sie diesen Befehl ausführen, ersetzen Sie die Beispielschlüssel-ID durch eine gültige Schlüssel-ID aus Ihrem AWS Konto.

```
aws kms verify \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --message-type TEXT \  
  --signature ...
```



```
--message fileb://EncodedMessage \  
--message-type RAW \  
--signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \  
--signature fileb://ExampleSignature
```

Ausgabe:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "SignatureValid": true,  
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"  
}
```

Weitere Informationen zur Verwendung asymmetrischer KMS Schlüssel finden Sie unter [Verwenden asymmetrischer Schlüssel](#) im AWS Key Management Service Developer Guide. [AWS KMS](#)

- API-Einheiten finden Sie unter [Überprüfen](#) in der AWS CLI Befehlsreferenz.

Beispiele für Lake Formation mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Lake Formation Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-lf-tags-to-resource

Das folgende Codebeispiel zeigt die Verwendung `add-lf-tags-to-resource`.

AWS CLI

Um ein oder mehrere LF-Tags an eine bestehende Ressource anzuhängen

Im folgenden `add-lf-tags-to-resource` Beispiel wird ein bestimmtes LF-Tag an die Tabellenressource angehängt.

```
aws lakeformation add-lf-tags-to-resource \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  },  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "usergroup",  
    "TagValues": [  
      "analyst"  
    ]  
  }]  
}
```

Ausgabe:

```
{  
  "Failures": []  
}
```

Weitere Informationen finden Sie unter [Zuweisen von LF-Tags zu Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [AddLfTagsToResource](#).AWS CLI

batch-grant-permissions

Das folgende Codebeispiel zeigt die Verwendung `batch-grant-permissions`.

AWS CLI

Um den Prinzipalen massenweise Berechtigungen für Ressourcen zu gewähren

Im folgenden `batch-grant-permissions` Beispiel wird den Prinzipalen Massenzugriff auf bestimmte Ressourcen gewährt.

```
aws lakeformation batch-grant-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Entries": [{  
    "Id": "1",  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-  
developer"  
    },  
    "Resource": {  
      "Table": {  
        "CatalogId": "123456789111",  
        "DatabaseName": "tpc",  
        "Name": "dl_tpc_promotion"  
      }  
    },  
    "Permissions": [  
      "ALL"  
    ],  
    "PermissionsWithGrantOption": [  
      "ALL"  
    ]  
  },  
  {  
    "Id": "2",  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-  
developer"
```

```
    },
    "Resource": {
      "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_customer"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": [
      "ALL"
    ]
  },
  {
    "Id": "3",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
    },
    "Resource": {
      "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": [
      "ALL"
    ]
  },
  {
    "Id": "4",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
    },
    "Resource": {
      "DataCellsFilter": {
        "TableCatalogId": "123456789111",
```

```

        "DatabaseName": "tpc",
        "TableName": "dl_tpc_item",
        "Name": "developer_item"
      }
    },
    "Permissions": [
      "SELECT"
    ],
    "PermissionsWithGrantOption": []
  }
]
}

```

Ausgabe:

```

{
  "Failures": []
}

```

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [BatchGrantPermissions AWS CLI Befehlsreferenz](#).

batch-revoke-permissions

Das folgende Codebeispiel zeigt die Verwendung `batch-revoke-permissions`.

AWS CLI

Um den Principals massenweise Berechtigungen für Ressourcen zu entziehen

Im folgenden `batch-revoke-permissions` Beispiel wird den Prinzipalen der Zugriff auf bestimmte Ressourcen massenweise entzogen.

```

aws lakeformation batch-revoke-permissions \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "CatalogId": "123456789111",

```

```
"Entries": [{
  "Id": "1",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
  },
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_promotion"
    }
  },
  "Permissions": [
    "ALL"
  ],
  "PermissionsWithGrantOption": [
    "ALL"
  ]
},
{
  "Id": "2",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
  },
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_promotion"
    }
  },
  "Permissions": [
    "ALL"
  ],
  "PermissionsWithGrantOption": [
    "ALL"
  ]
}
]
```

Ausgabe:

```
{
  "Failures": []
}
```

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [BatchRevokePermissions AWS CLI](#) Befehlsreferenz.

cancel-transaction

Das folgende Codebeispiel zeigt die Verwendung `cancel-transaction`.

AWS CLI

Um eine Transaktion zu stornieren

Im folgenden `cancel-transaction` Beispiel wird die Transaktion storniert.

```
aws lakeformation cancel-transaction \
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [CancelTransaction](#) in der AWS CLI Befehlsreferenz.

commit-transaction

Das folgende Codebeispiel zeigt die Verwendung `commit-transaction`.

AWS CLI

Um die Transaktion festzuschreiben

Im folgenden `commit-transaction` Beispiel wird die Transaktion festgeschrieben.

```
aws lakeformation commit-transaction \
```

```
--transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Ausgabe:

```
{
  "TransactionStatus": "committed"
}
```

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [CommitTransaction](#) in der AWS CLI Befehlsreferenz.

create-data-cells-filter

Das folgende Codebeispiel zeigt die Verwendung `create-data-cells-filter`.

AWS CLI

Beispiel 1: Um einen Datenzellenfilter zu erstellen

Im folgenden `create-data-cells-filter` Beispiel wird ein Datenzellenfilter erstellt, der es ermöglicht, Zugriff auf bestimmte Spalten auf der Grundlage der Zeilenbedingungen zu gewähren.

```
aws lakeformation create-data-cells-filter \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "TableData": {
    "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],
    "DatabaseName": "tpc",
    "Name": "developer_promotion",
    "RowFilter": {
      "FilterExpression": "p_promo_name='ese'"
    },
    "TableCatalogId": "123456789111",
    "TableName": "dl_tpc_promotion"
  }
}
```


Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Datenfilterung und Sicherheit auf Zellebene in Lake Formation im AWS Lake Formation Developer Guide](#).

Beispiel 2: So erstellen Sie einen Spaltenfilter

Im folgenden `create-data-cells-filter` Beispiel wird ein Datenfilter erstellt, mit dem Zugriff auf bestimmte Spalten gewährt werden kann.

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "TableData": {  
    "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],  
    "DatabaseName": "tpc",  
    "Name": "developer_promotion_allrows",  
    "RowFilter": {  
      "AllRowsWildcard": {}  
    },  
    "TableCatalogId": "123456789111",  
    "TableName": "dl_tpc_promotion"  
  }  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Datenfilterung und Sicherheit auf Zellebene in Lake Formation im AWS Lake Formation Developer Guide](#).

Beispiel 3: So erstellen Sie einen Datenfilter mit Ausschlussspalten

Das folgende `create-data-cells-filter` Beispiel erstellt einen Datenfilter, der es ermöglicht, Zugriff auf alle Spalten mit Ausnahme der genannten Spalten zu gewähren.

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "TableData": {
    "ColumnWildcard": {
      "ExcludedColumnNames": ["p_channel_details", "p_start_date_sk"]
    },
    "DatabaseName": "tpc",
    "Name": "developer_promotion_excludedcolumn",
    "RowFilter": {
      "AllRowsWildcard": {}
    },
    "TableCatalogId": "123456789111",
    "TableName": "dl_tpc_promotion"
  }
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Datenfilterung und Sicherheit auf Zellebene in Lake Formation im AWS Lake Formation Developer Guide](#).

- API-Einheiten finden Sie [CreateDataCellsFilter](#) in der AWS CLI Befehlsreferenz.

create-lf-tag

Das folgende Codebeispiel zeigt die Verwendung `create-lf-tag`.

AWS CLI

Um ein LF-Tag zu erstellen

Im folgenden `create-lf-tag` Beispiel wird ein LF-Tag mit dem angegebenen Namen und den angegebenen Werten erstellt.

```
aws lakeformation create-lf-tag \
  --catalog-id '123456789111' \
  --tag-key 'usergroup' \
  --tag-values '["developer","analyst","campaign"]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing LF-Tags for Metadata Access Control](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [CreateLfTag AWS CLI](#) Befehlsreferenz.

delete-data-cells-filter

Das folgende Codebeispiel zeigt die Verwendung `delete-data-cells-filter`.

AWS CLI

Um den Datenzellenfilter zu löschen

Im folgenden `delete-data-cells-filter` Beispiel wird der angegebene Datenzellenfilter gelöscht.

```
aws lakeformation delete-data-cells-filter \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "TableCatalogId": "123456789111",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_promotion",  
  "Name": "developer_promotion"  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Datenfilterung und Sicherheit auf Zellebene in Lake Formation im AWS Lake Formation Developer Guide](#).

- API-Einheiten finden Sie [DeleteDataCellsFilter](#) in der AWS CLI Befehlsreferenz.

delete-lf-tag

Das folgende Codebeispiel zeigt die Verwendung `delete-lf-tag`.

AWS CLI

Um die LF-Tag-Definition zu löschen

Im folgenden `delete-lf-tag` Beispiel wird die LF-Tag-Definition gelöscht.

```
aws lakeformation delete-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing LF-Tags for Metadata Access Control](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [DeleteLfTag AWS CLI Befehlsreferenz](#).

delete-objects-on-cancel

Das folgende Codebeispiel zeigt die Verwendung `delete-objects-on-cancel`.

AWS CLI

Um ein Objekt zu löschen, wenn die Transaktion abgebrochen wird

Im folgenden `delete-objects-on-cancel` Beispiel wird das aufgelistete S3-Objekt gelöscht, wenn die Transaktion abgebrochen wird.

```
aws lakeformation delete-objects-on-cancel \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_household_demographics_gov",  
  "TransactionId": "1234d972ca8347b89825e33c5774aec4",  
  "Objects": [{  
    "Uri": "s3://lf-data-lake-012345678901/target/  
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-  
ff26b17504414fe88b302cd795eabd00.parquet",  
    "ETag": "1234ab1fc50a316b149b4e1f21a73800"  
  }]  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [DeleteObjectsOnCancel](#) in der AWS CLI Befehlsreferenz.

deregister-resource

Das folgende Codebeispiel zeigt die Verwendung `deregister-resource`.

AWS CLI

Um den Data Lake-Speicher zu deregistrieren

Im folgenden `deregister-resource` Beispiel wird die Registrierung der Ressource aufgehoben, so wie sie von der Lake Formation verwaltet wird.

```
aws lakeformation deregister-resource \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen eines Amazon S3 S3-Standorts zu Ihrem Data Lake](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [DeregisterResource](#) in der AWS CLI Befehlsreferenz.

describe-transaction

Das folgende Codebeispiel zeigt die Verwendung `describe-transaction`.

AWS CLI

Um Transaktionsdetails abzurufen

Das folgende `describe-transaction` Beispiel gibt die Details einer einzelnen Transaktion zurück.

```
aws lakeformation describe-transaction \  
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Ausgabe:

```
{  
  "TransactionDescription": {  
    "TransactionId": "12345972ca8347b89825e33c5774aec4",  
    "TransactionStatus": "committed",  
    "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",  
    "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [DescribeTransaction](#) in der AWS CLI Befehlsreferenz.

extend-transaction

Das folgende Codebeispiel zeigt die Verwendung `extend-transaction`.

AWS CLI

Um eine Transaktion zu verlängern

Das folgende `extend-transaction` Beispiel erweitert die Transaktion.

```
aws lakeformation extend-transaction \  
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [ExtendTransaction](#) in der AWS CLI Befehlsreferenz.

get-data-lake-settings

Das folgende Codebeispiel zeigt die Verwendung `get-data-lake-settings`.

AWS CLI

So rufen Sie von AWS Lake Formation verwaltete Data Lake-Einstellungen ab

Im folgenden `get-data-lake-settings` Beispiel wird die Liste der Data Lake-Administratoren und anderer Data Lake-Einstellungen abgerufen.

```
aws lakeformation get-data-lake-settings \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111"  
}
```

Ausgabe:

```
{  
  "DataLakeSettings": {  
    "DataLakeAdmins": [{  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"  
    }],  
    "CreateDatabaseDefaultPermissions": [],  
    "CreateTableDefaultPermissions": [  
      {  
        "Principal": {  
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"  
        },  
        "Permissions": [  
          "ALL"  
        ]  
      }  
    ],  
    "TrustedResourceOwners": [],  
    "AllowExternalDataFiltering": true,  
    "ExternalDataFilteringAllowList": [{  
      "DataLakePrincipalIdentifier": "123456789111"  
    }],  
    "AuthorizedSessionTagValueList": [  
      "Amazon EMR"  
    ]  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Ändern der Standardsicherheitseinstellungen für Ihren Data Lake im AWS Lake Formation Developer Guide](#).

- API-Einheiten finden Sie [GetDataLakeSettings](#) unter AWS CLI Befehlsreferenz.

get-effective-permissions-for-path

Das folgende Codebeispiel zeigt die Verwendung `get-effective-permissions-for-path`.

AWS CLI

Um Berechtigungen für Ressourcen abzurufen, die sich in einem bestimmten Pfad befinden

Das folgende `get-effective-permissions-for-path` Beispiel gibt die Lake Formation Formation-Berechtigungen für eine angegebene Tabelle oder Datenbankressource zurück, die sich in einem Pfad in Amazon S3 befindet.

```
aws lakeformation get-effective-permissions-for-path \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"  
}
```

Ausgabe:

```
{  
  "Permissions": [{  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-campaign-manager"  
    },  
    "Resource": {  
      "Database": {  
        "Name": "tpc"  
      }  
    },  
  },  
]
```



```
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/EMR-
RuntimeRole"
    },
    "Resource": {
      "Database": {
        "Name": "tpc"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": []
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:saml-
provider/oktaSAMLProvider:user/emr-developer"
    },
    "Resource": {
      "Database": {
        "Name": "tpc"
      }
    },
    "Permissions": [
      "ALL",
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    },
    "Resource": {
      "Database": {
        "Name": "tpc"
      }
    }
  }
}
```

```

    }
  },
  "Permissions": [
    "ALL",
    "ALTER",
    "CREATE_TABLE",
    "DESCRIBE",
    "DROP"
  ],
  "PermissionsWithGrantOption": [
    "ALL",
    "ALTER",
    "CREATE_TABLE",
    "DESCRIBE",
    "DROP"
  ]
},
{
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/LF-
GlueServiceRole"
  },
  "Resource": {
    "Database": {
      "Name": "tpc"
    }
  },
  "Permissions": [
    "CREATE_TABLE"
  ],
  "PermissionsWithGrantOption": []
}
],
"NextToken":
"E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjBUYwYVc5dU1qcDdJbk5sWTI5dVpITW1P
}

```

Weitere Informationen finden Sie unter [Managing Lake Formation-Berechtigungen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [GetEffectivePermissionsForPath](#) in der AWS CLI Befehlsreferenz.

get-lf-tag

Das folgende Codebeispiel zeigt die Verwendung `get-lf-tag`.

AWS CLI

Um die LF-Tag-Definition abzurufen

Im folgenden `get-lf-tag` Beispiel wird die LF-Tag-Definition abgerufen.

```
aws lakeformation get-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup'
```

Ausgabe:

```
{  
  "CatalogId": "123456789111",  
  "TagKey": "usergroup",  
  "TagValues": [  
    "analyst",  
    "campaign",  
    "developer"  
  ]  
}
```

Weitere Informationen finden Sie unter [Managing LF-Tags for Metadata Access Control](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [GetLfTag AWS CLI](#) Befehlsreferenz.

get-query-state

Das folgende Codebeispiel zeigt die Verwendung `get-query-state`.

AWS CLI

Um den Status einer gesendeten Abfrage abzurufen

Das folgende `get-query-state` Beispiel gibt den Status einer zuvor übermittelten Abfrage zurück.

```
aws lakeformation get-query-state \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Ausgabe:

```
{  
  "State": "FINISHED"  
}
```

Weitere Informationen finden Sie unter [Transactional Data Operations](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [GetQueryState](#) in der AWS CLI Befehlsreferenz.

get-query-statistics

Das folgende Codebeispiel zeigt die Verwendung `get-query-statistics`.

AWS CLI

Um Abfragestatistiken abzurufen

Im folgenden `get-query-statistics` Beispiel werden Statistiken zur Planung und Ausführung einer Abfrage abgerufen.

```
aws lakeformation get-query-statistics \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Ausgabe:

```
{  
  "ExecutionStatistics": {  
    "AverageExecutionTimeMillis": 0,  
    "DataScannedBytes": 0,  
    "WorkUnitsExecutedCount": 0  
  },  
  "PlanningStatistics": {  
    "EstimatedDataToScanBytes": 43235,  
    "PlanningTimeMillis": 2377,  
    "QueueTimeMillis": 440,  
  }  
}
```

```
    "WorkUnitsGeneratedCount": 1
  },
  "QuerySubmissionTime": "2022-08-11T02:14:38.641870+00:00"
}
```

Weitere Informationen finden Sie unter [Transactional Data Operations](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [GetQueryStatistics](#) in der AWS CLI Befehlsreferenz.

get-resource-lf-tags

Das folgende Codebeispiel zeigt die Verwendung `get-resource-lf-tags`.

AWS CLI

Um LF-Tags aufzulisten

Das folgende `list-lf-tags` Beispiel gibt eine Liste von LF-Tags zurück, zu deren Anzeige der Anforderer berechtigt ist.

```
aws lakeformation list-lf-tags \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "ResourceShareType": "ALL",  
  "MaxResults": 2  
}
```

Ausgabe:

```
{  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "category",  
    "TagValues": [  
      "private",
```

```
        "public"
      ]
    },
    {
      "CatalogId": "123456789111",
      "TagKey": "group",
      "TagValues": [
        "analyst",
        "campaign",
        "developer"
      ]
    }
  ]],
  "NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWVbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"
}
```

Weitere Informationen finden Sie unter [Managing LF-Tags for Metadata Access Control](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [GetResourceLfTags AWS CLI-Befehlsreferenz](#).

get-table-objects

Das folgende Codebeispiel zeigt die Verwendung von `get-table-objects`.

AWS CLI

Um Objekte einer verwalteten Tabelle aufzulisten

Das folgende `get-table-objects` Beispiel gibt den Satz von Amazon S3 S3-Objekten zurück, aus denen die angegebene verwaltete Tabelle besteht.

```
aws lakeformation get-table-objects \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_household_demographics_gov",  
  "QueryAsOfTime": "2022-08-10T15:00:00"
```

```
}

```

Ausgabe:

```
{
  "Objects": [{
    "PartitionValues": [],
    "Objects": [{
      "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
      "ETag": "12345b1fc50a316b149b4e1f21a73800",
      "Size": 43235
    }]
  }]
}
```

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [GetTableObjects](#) in der AWS CLI Befehlsreferenz.

get-work-unit-results

Das folgende Codebeispiel zeigt die Verwendung `get-work-unit-results`.

AWS CLI

Um Arbeitseinheiten einer bestimmten Abfrage abzurufen

Das folgende `get-work-unit-results` Beispiel gibt die Arbeitseinheiten zurück, die sich aus der Abfrage ergeben.

```
aws lakeformation get-work-units \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \
  --work-unit-id '0' \
  --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

Ausgabe:

```
outfile with Blob content.
```

Weitere Informationen finden Sie unter [Transactional Data Operations](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [GetWorkUnitResults](#) in der AWS CLI Befehlsreferenz.

get-work-units

Das folgende Codebeispiel zeigt die Verwendung von `get-work-units`.

AWS CLI

Um Arbeitseinheiten abzurufen

Im folgenden `get-work-units` Beispiel werden die durch den StartQueryPlanning Vorgang generierten Arbeitseinheiten abgerufen.

```
aws lakeformation get-work-units \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Ausgabe:

```
{
  "WorkUnitRanges": [{
    "WorkUnitIdMax": 0,
    "WorkUnitIdMin": 0,
    "WorkUnitToken":
      "1234eMAk4kL04umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
      +f88jzGrYq22gE6jkQlp0B
      +0et2eqNUMFudAAAAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBg1ghkgBZQMEAS4wEQQMCOEWRda
      wAAAAEAAAAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmB1
      +VSojiG0tdsUk7vcjYXUUboYm3dvqRqX2s4gROM0n
      +Ij8R0/8jYmnHkpvyAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytm0hozxD0Ei/
      F2ZoXz6w1GDfGUo/2WxCKY0hTyNaw6TM
      +7drTM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzB1AjBH8okRtYP8R7AY2W1s/
      hqFBhG0V4142AC0LxsuZbMQrE2SzwZUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCNk1ExMR
      +btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
    }],
  "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
}
```

Weitere Informationen finden Sie unter [Transactional Data Operations](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [GetWorkUnits](#) in der AWS CLI Befehlsreferenz.

grant-permissions

Das folgende Codebeispiel zeigt die Verwendung `grant-permissions`.

AWS CLI

Beispiel 1: Um dem Prinzipal mithilfe von LF-Tags Berechtigungen für Ressourcen zu erteilen

Im folgenden `grant-permissions` Beispiel werden dem Prinzipal ALL Berechtigungen für eine Datenbankressource erteilt, die der LF-Tag-Richtlinie entspricht.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"  
  },  
  "Resource": {  
    "LFTagPolicy": {  
      "CatalogId": "123456789111",  
      "ResourceType": "DATABASE",  
      "Expression": [{  
        "TagKey": "usergroup",  
        "TagValues": [  
          "analyst",  
          "developer"  
        ]  
      }]  
    }  
  },  
  "Permissions": [  
    "ALL"  
  ],  
  "PermissionsWithGrantOption": [  
    "ALL"  
  ]  
}
```

```
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

Beispiel 2: So gewähren Sie dem Prinzipal Berechtigungen auf Spaltenebene

Im folgenden `grant-permissions` Beispiel wird dem Prinzipal die Berechtigung erteilt, eine bestimmte Spalte auszuwählen.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "TableWithColumns": {  
      "CatalogId": "123456789111",  
      "ColumnNames": ["p_end_date_sk"],  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  },  
  "Permissions": [  
    "SELECT"  
  ],  
  "PermissionsWithGrantOption": []  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

Beispiel 3: So gewähren Sie dem Prinzipal Tabellenberechtigungen

Im folgenden `grant-permissions` Beispiel wird dem Prinzipal die `Select`-Berechtigung für alle Tabellen der angegebenen Datenbank erteilt.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "TableWildcard": {}  
    }  
  },  
  "Permissions": [  
    "SELECT"  
  ],  
  "PermissionsWithGrantOption": []  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

Beispiel 4: So erteilen Sie dem Prinzipal Berechtigungen für LF-Tags

Im folgenden `grant-permissions` Beispiel wird dem Prinzipal die Zugriffsberechtigung für LF-Tags erteilt.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
```

```

    "CatalogId": "123456789111",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
      "LFTag": {
        "CatalogId": "123456789111",
        "TagKey": "category",
        "TagValues": [
          "private", "public"
        ]
      }
    },
    "Permissions": [
      "ASSOCIATE"
    ],
    "PermissionsWithGrantOption": []
  }
}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

Beispiel 5: So erteilen Sie dem Prinzipal Berechtigungen für Datenspeicherorte

Im folgenden `grant-permissions` Beispiel wird dem Prinzipal die Erlaubnis zum Speicherort von Daten erteilt.

```

aws lakeformation grant-permissions \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "CatalogId": "123456789111",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
  },
  "Resource": {
    "DataLocation": {

```

```
        "CatalogId": "123456789111",
        "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
    }
},
"Permissions": [
    "DATA_LOCATION_ACCESS"
],
"PermissionsWithGrantOption": []
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [GrantPermissions AWS CLI Befehlsreferenz](#).

list-data-cells-filter

Das folgende Codebeispiel zeigt die Verwendung `list-data-cells-filter`.

AWS CLI

Um Datenzellenfilter aufzulisten

Das folgende `list-data-cells-filter` Beispiel listet den Datenzellenfilter für eine bestimmte Tabelle auf.

```
aws lakeformation list-data-cells-filter \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "MaxResults": 2,
  "Table": {
    "CatalogId": "123456789111",
    "DatabaseName": "tpc",
    "Name": "dl_tpc_promotion"
  }
}
```

Ausgabe:

```
{
  "DataCellsFilters": [{
    "TableCatalogId": "123456789111",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_promotion",
    "Name": "developer_promotion",
    "RowFilter": {
      "FilterExpression": "p_promo_name='ese'"
    },
    "ColumnNames": [
      "p_channel_details",
      "p_start_date_sk",
      "p_purpose",
      "p_promo_id",
      "p_promo_name",
      "p_end_date_sk",
      "p_discount_active"
    ]
  },
  {
    "TableCatalogId": "123456789111",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_promotion",
    "Name": "developer_promotion_allrows",
    "RowFilter": {
      "FilterExpression": "TRUE",
      "AllRowsWildcard": {}
    },
    "ColumnNames": [
      "p_channel_details",
      "p_start_date_sk",
      "p_promo_name"
    ]
  }
],
  "NextToken": "2MDA2MTgwNiwibmFub3MiOjE0MDAwMDAwMH19"
}
```

Weitere Informationen finden Sie unter [Datenfilterung und Sicherheit auf Zellebene in Lake Formation im AWS Lake Formation Developer Guide](#).

- API-Einheiten finden Sie [ListDataCellsFilter](#) in der AWS CLI Befehlsreferenz.

list-permissions

Das folgende Codebeispiel zeigt die Verwendung `list-permissions`.

AWS CLI

Beispiel 1: Um eine Liste der Hauptberechtigungen für die Ressource abzurufen

Das folgende `list-permissions` Beispiel gibt eine Liste der Hauptberechtigungen für die Datenbankressourcen zurück.

```
aws lakeformation list-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "ResourceType": "DATABASE",  
  "MaxResults": 2  
}
```

Ausgabe:

```
{  
  "PrincipalResourcePermissions": [{  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-  
campaign-manager"  
    },  
    "Resource": {  
      "Database": {  
        "CatalogId": "123456789111",  
        "Name": "tpc"  
      }  
    },  
    "Permissions": [  
      "DESCRIBE"  
    ],  
    "PermissionsWithGrantOption": []  
  }],  
  "NextToken":  
  "E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dU1qcDdJbk5sWTI5dVpITWlP"
```

```
}
```

Weitere Informationen finden Sie unter [Managing Lake Formation-Berechtigungen](#) im AWS Lake Formation Developer Guide.

Beispiel 2: So rufen Sie eine Liste der Hauptberechtigungen für die Tabelle mit Datenfiltern ab

Im folgenden `list-permissions` Beispiel werden die Berechtigungen für die Tabelle mit zugehörigen Datenfiltern aufgeführt, die dem Prinzipal gewährt wurden.

```
aws lakeformation list-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_customer"  
    }  
  },  
  "IncludeRelated": "TRUE",  
  "MaxResults": 10  
}
```

Ausgabe:

```
{  
  "PrincipalResourcePermissions": [{  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/  
Admin"  
    },  
    "Resource": {  
      "Table": {  
        "CatalogId": "123456789111",  
        "DatabaseName": "customer",  
        "Name": "customer_invoice"  
      }  
    }  
  }  
}
```



```
    },
    "Permissions": [
      "ALL",
      "ALTER",
      "DELETE",
      "DESCRIBE",
      "DROP",
      "INSERT"
    ],
    "PermissionsWithGrantOption": [
      "ALL",
      "ALTER",
      "DELETE",
      "DESCRIBE",
      "DROP",
      "INSERT"
    ]
  ],
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
    },
    "Resource": {
      "TableWithColumns": {
        "CatalogId": "123456789111",
        "DatabaseName": "customer",
        "Name": "customer_invoice",
        "ColumnWildcard": {}
      }
    },
    "Permissions": [
      "SELECT"
    ],
    "PermissionsWithGrantOption": [
      "SELECT"
    ]
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
    },
    "Resource": {
```

```

        "DataCellsFilter": {
            "TableCatalogId": "123456789111",
            "DatabaseName": "customer",
            "TableName": "customer_invoice",
            "Name": "dl_us_customer"
        }
    },
    "Permissions": [
        "DESCRIBE",
        "SELECT",
        "DROP"
    ],
    "PermissionsWithGrantOption": []
}
],
"NextToken": "VyeUFjY291bnRQZXJtaXNzaW9ucyI6ZmFsc2V9"
}

```

Weitere Informationen finden Sie unter [Managing Lake Formation-Berechtigungen](#) im AWS Lake Formation Developer Guide.

Beispiel 3: Um eine Liste der Hauptberechtigungen für die LF-Tags abzurufen

Das folgende `list-permissions` Beispiel listet die Berechtigungen für die LF-Tags auf, die dem Principal gewährt wurden.

```

aws lakeformation list-permissions \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "CatalogId": "123456789111",
  "Resource": {
    "LFTag": {
      "CatalogId": "123456789111",
      "TagKey": "category",
      "TagValues": [
        "private"
      ]
    }
  }
},

```

```
"MaxResults": 10
}
```

Ausgabe:

```
{
  "PrincipalResourcePermissions": [{
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    },
    "Resource": {
      "LFTag": {
        "CatalogId": "123456789111",
        "TagKey": "category",
        "TagValues": [
          "*"
        ]
      }
    },
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": [
      "DESCRIBE"
    ]
  }],
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    },
    "Resource": {
      "LFTag": {
        "CatalogId": "123456789111",
        "TagKey": "category",
        "TagValues": [
          "*"
        ]
      }
    },
    "Permissions": [
      "ASSOCIATE"
    ]
  }
]
```

```

    ],
    "PermissionsWithGrantOption": [
        "ASSOCIATE"
    ]
}
],
"NextToken": "EJwY21GMGF0XVJanA3SW50cm1pc3Npb25zIjpmYWxzZX0="
}

```

Weitere Informationen finden Sie unter [Managing Lake Formation-Berechtigungen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [ListPermissions](#) in der AWS CLI Befehlsreferenz.

list-resources

Das folgende Codebeispiel zeigt die Verwendung `list-resources`.

AWS CLI

Um die von der Lake Formation verwalteten Ressourcen aufzulisten

Das folgende `list-resources` Beispiel listet die Ressourcen auf, die der Bedingung entsprechen, die von der Lake Formation verwaltet wird.

```

aws lakeformation list-resources \
  --cli-input-json file://input.json

```

Inhalt von `input.json`:

```

{
  "FilterConditionList": [{
    "Field": "ROLE_ARN",
    "ComparisonOperator": "CONTAINS",
    "StringValueList": [
      "123456789111"
    ]
  }],
  "MaxResults": 10
}

```

Ausgabe:

```
{
  "ResourceInfoList": [{
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111",
    "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole",
    "LastModified": "2022-07-21T02:12:46.669000+00:00"
  },
  {
    "ResourceArn": "arn:aws:s3:::lf-emr-test-123456789111",
    "RoleArn": "arn:aws:iam::123456789111:role/EMRIFS3Role",
    "LastModified": "2022-07-29T16:22:03.211000+00:00"
  }
]
}
```

Weitere Informationen finden Sie unter [Managing Lake Formation-Berechtigungen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [ListResources](#) in der AWS CLI Befehlsreferenz.

list-transactions

Das folgende Codebeispiel zeigt die Verwendung `list-transactions`.

AWS CLI

Um alle Transaktionsdetails aufzulisten

Im folgenden `list-transactions` Beispiel werden Metadaten zu Transaktionen und deren Status zurückgegeben.

```
aws lakeformation list-transactions \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "CatalogId": "123456789111",
  "StatusFilter": "ALL",
  "MaxResults": 3
}
```

Ausgabe:

```
{
  "Transactions": [{
    "TransactionId": "1234569f08804cb790d950d4d0fe485e",
    "TransactionStatus": "committed",
    "TransactionStartTime": "2022-08-10T14:32:29.220000+00:00",
    "TransactionEndTime": "2022-08-10T14:32:33.751000+00:00"
  },
  {
    "TransactionId": "12345972ca8347b89825e33c5774aec4",
    "TransactionStatus": "committed",
    "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
    "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
  },
  {
    "TransactionId": "12345daf6cb047dbba8ad9b0414613b2",
    "TransactionStatus": "committed",
    "TransactionStartTime": "2022-08-10T13:56:51.261000+00:00",
    "TransactionEndTime": "2022-08-10T13:56:51.547000+00:00"
  }
  ],
  "NextToken": "77X1ebypsI7os+X2lhHsZLGNC DK3nNGpwRdFpicS0HgcX1/
QMoniUAKcpR3kj3ts3PVdMA=="
}
```

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [ListTransactions](#) in der AWS CLI Befehlsreferenz.

put-data-lake-settings

Das folgende Codebeispiel zeigt die Verwendung `put-data-lake-settings`.

AWS CLI

So legen Sie von AWS Lake Formation verwaltete Data Lake-Einstellungen fest

Im folgenden `put-data-lake-settings` Beispiel werden die Liste der Data Lake-Administratoren und anderer Data Lake-Einstellungen festgelegt.

```
aws lakeformation put-data-lake-settings \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "DataLakeSettings": {
    "DataLakeAdmins": [{
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    }
  ],
  "CreateDatabaseDefaultPermissions": [],
  "CreateTableDefaultPermissions": [],
  "TrustedResourceOwners": [],
  "AllowExternalDataFiltering": true,
  "ExternalDataFilteringAllowList": [{
    "DataLakePrincipalIdentifier": "123456789111"
  }],
  "AuthorizedSessionTagValueList": ["Amazon EMR"]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ändern der Standardsicherheitseinstellungen für Ihren Data Lake im AWS Lake Formation Developer Guide](#).

- API-Einheiten finden Sie [PutDataLakeSettings](#) unter AWS CLI Befehlsreferenz.

register-resource

Das folgende Codebeispiel zeigt die Verwendung `register-resource`.

AWS CLI

Beispiel 1: So registrieren Sie Data Lake-Speicher mithilfe von Service Linked Role

Im folgenden `register-resource` Beispiel wird die Ressource mithilfe der mit dem Dienst verknüpften Rolle als von der Lake Formation verwaltet registriert.

```
aws lakeformation register-resource \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
  "UseServiceLinkedRole": true
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen eines Amazon S3 S3-Standorts zu Ihrem Data Lake](#) im AWS Lake Formation Developer Guide.

Beispiel 2: So registrieren Sie Data Lake-Speicher mithilfe einer benutzerdefinierten Rolle

Im folgenden `register-resource` Beispiel wird die Ressource mithilfe einer benutzerdefinierten Rolle als von der Lake Formation verwaltet registriert.

```
aws lakeformation register-resource \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
  "UseServiceLinkedRole": false,
  "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen eines Amazon S3 S3-Standorts zu Ihrem Data Lake](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [RegisterResource](#) in der AWS CLI Befehlsreferenz.

remove-lf-tags-from-resource

Das folgende Codebeispiel zeigt die Verwendung `remove-lf-tags-from-resource`.

AWS CLI

Um das LF-Tag aus einer Ressource zu entfernen

Im folgenden `remove-lf-tags-from-resource` Beispiel wird die LF-Tag-Zuordnung zur Tabellenressource entfernt.

```
aws lakeformation remove-lf-tags-from-resource \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  },  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "usergroup",  
    "TagValues": [  
      "developer"  
    ]  
  }]  
}
```

Ausgabe:

```
{  
  "Failures": []  
}
```

Weitere Informationen finden Sie unter [Zuweisen von LF-Tags zu Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [RemoveLfTagsFromResource](#).AWS CLI

revoke-permissions

Das folgende Codebeispiel zeigt die Verwendung `revoke-permissions`.

AWS CLI

Um dem Prinzipal Berechtigungen für Ressourcen zu entziehen

Im folgenden `revoke-permissions` Beispiel wird dem Prinzipalzugriff auf eine bestimmte Tabelle einer bestimmten Datenbank entzogen.

```
aws lakeformation revoke-permissions \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  },  
  "Permissions": [  
    "ALL"  
  ],  
  "PermissionsWithGrantOption": []  
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [RevokePermissions AWS CLI](#) Befehlsreferenz.

search-databases-by-lf-tags

Das folgende Codebeispiel zeigt die Verwendung `search-databases-by-lf-tags`.

AWS CLI

Um nach Datenbankressourcen zu suchen nach LFTags

Im folgenden `search-databases-by-lf-tags` Beispiel wird nach Datenbankressourcen gesucht, die dem LFTag Ausdruck entsprechen.

```
aws lakeformation search-databases-by-lf-tags \  
--cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "MaxResults": 1,  
  "CatalogId": "123456789111",  
  "Expression": [{  
    "TagKey": "usergroup",  
    "TagValues": [  
      "developer"  
    ]  
  }]  
}
```

Ausgabe:

```
{  
  "DatabaseList": [{  
    "Database": {  
      "CatalogId": "123456789111",  
      "Name": "tpc"  
    },  
    "LFTags": [{  
      "CatalogId": "123456789111",  
      "TagKey": "usergroup",  
      "TagValues": [  
        "developer"  
      ]  
    }]  
  }]  
}
```

Weitere Informationen finden Sie [im AWS Lake Formation Developer Guide unter Anzeigen der Ressourcen, denen ein LF-Tag zugewiesen ist](#).

- API-Einheiten finden Sie [SearchDatabasesByLfTags](#) in der AWS CLI Befehlsreferenz.

search-tables-by-lf-tags

Das folgende Codebeispiel zeigt die Verwendung `search-tables-by-lf-tags`.

AWS CLI

Um nach Tabellenressourcen zu suchen nach LFTags

Im folgenden `search-tables-by-lf-tags` Beispiel wird nach Tabellenressourcen gesucht, die einem LFTag Ausdruck entsprechen.

```
aws lakeformation search-tables-by-lf-tags \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "MaxResults": 2,  
  "CatalogId": "123456789111",  
  "Expression": [{  
    "TagKey": "usergroup",  
    "TagValues": [  
      "developer"  
    ]  
  }]  
}
```

Ausgabe:

```
{  
  "NextToken": "c2VhcmNoQWxsVGFnY0luVGFiYGVzIjpmYWxzZX0=",  
  "TableList": [{  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_item"  
    },  
  },  
}
```

```
"LFTagOnDatabase": [{
  "CatalogId": "123456789111",
  "TagKey": "usergroup",
  "TagValues": [
    "developer"
  ]
}],
"LFTagsOnTable": [{
  "CatalogId": "123456789111",
  "TagKey": "usergroup",
  "TagValues": [
    "developer"
  ]
}],
"LFTagsOnColumns": [{
  "Name": "i_item_desc",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  ]
}],
{
  "Name": "i_container",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  ]
}],
{
  "Name": "i_wholesale_cost",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  ]
}],
},
```

```
{
  "Name": "i_manufact_id",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_brand_id",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_formulation",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_current_price",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_size",
  "LFTags": [{
    "CatalogId": "123456789111",
```

```
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }
},
{
    "Name": "i_rec_start_date",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
},
{
    "Name": "i_manufact",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
},
{
    "Name": "i_item_sk",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
},
{
    "Name": "i_manager_id",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

```
    ]],  
  },  
  {  
    "Name": "i_item_id",  
    "LFTags": [{  
      "CatalogId": "123456789111",  
      "TagKey": "usergroup",  
      "TagValues": [  
        "developer"  
      ]  
    }]  
  },  
  {  
    "Name": "i_class_id",  
    "LFTags": [{  
      "CatalogId": "123456789111",  
      "TagKey": "usergroup",  
      "TagValues": [  
        "developer"  
      ]  
    }]  
  },  
  {  
    "Name": "i_class",  
    "LFTags": [{  
      "CatalogId": "123456789111",  
      "TagKey": "usergroup",  
      "TagValues": [  
        "developer"  
      ]  
    }]  
  },  
  {  
    "Name": "i_category",  
    "LFTags": [{  
      "CatalogId": "123456789111",  
      "TagKey": "usergroup",  
      "TagValues": [  
        "developer"  
      ]  
    }]  
  },  
  {  
    "Name": "i_category_id",
```



```
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_brand",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_units",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_rec_end_date",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_color",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
```

```

        "developer"
      ]
    ]}
  },
  {
    "Name": "i_product_name",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  }
]
}]
}

```

Weitere Informationen finden Sie [im AWS Lake Formation Developer Guide unter Anzeigen der Ressourcen, denen ein LF-Tag zugewiesen ist](#).

- API-Einheiten finden Sie [SearchTablesByLFTags](#) in der AWS CLI Befehlsreferenz.

start-query-planning

Das folgende Codebeispiel zeigt die Verwendung `start-query-planning`.

AWS CLI

Um eine Abfrageanweisung zu verarbeiten

Im folgenden `start-query-planning` Beispiel wird eine Anforderung zur Verarbeitung einer Abfrageanweisung gesendet.

```
aws lakeformation start-query-planning \
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{
  "QueryPlanningContext": {
    "CatalogId": "012345678901",
```

```
    "DatabaseName": "tpc"
  },
  "QueryString": "select * from dl_tpc_household_demographics_gov where
hd_income_band_sk=9"
}
```

Ausgabe:

```
{
  "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"
}
```

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [StartQueryPlanning](#) in der AWS CLI Befehlsreferenz.

start-transaction

Das folgende Codebeispiel zeigt die Verwendung `start-transaction`.

AWS CLI

Um eine neue Transaktion zu starten

Das folgende `start-transaction` Beispiel startet eine neue Transaktion und gibt ihre Transaktions-ID zurück.

```
aws lakeformation start-transaction \
  --transaction-type = 'READ_AND_WRITE'
```

Ausgabe:

```
{
  "TransactionId": "b014d972ca8347b89825e33c5774aec4"
}
```

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [StartTransaction](#) in der AWS CLI Befehlsreferenz.

update-lf-tag

Das folgende Codebeispiel zeigt die Verwendung `update-lf-tag`.

AWS CLI

Um die LF-Tag-Definition zu aktualisieren

Im folgenden `update-lf-tag` Beispiel wird die LF-Tag-Definition aktualisiert.

```
aws lakeformation update-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values-to-add ['admin']
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing LF-Tags for Metadata Access Control](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie unter [UpdateLFTag AWS CLI Befehlsreferenz](#).

update-table-objects

Das folgende Codebeispiel zeigt die Verwendung `update-table-objects`.

AWS CLI

Um Objekte einer verwalteten Tabelle zu ändern

Das folgende `update-table-objects` Beispiel fügt bereitgestellte S3-Objekte zur angegebenen verwalteten Tabelle hinzu.

```
aws lakeformation update-table-objects \  
  --cli-input-json file://input.json
```

Inhalt von `input.json`:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",
```

```
"TableName": "dl_tpc_household_demographics_gov",
"TransactionId": "12347a9f75424b9b915f6ff201d2a190",
"WriteOperations": [{
  "AddObject": {
    "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
    "ETag": "1234ab1fc50a316b149b4e1f21a73800",
    "Size": 42200
  }
}]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Lesen aus dem Data Lake und Schreiben in den Data Lake innerhalb von Transaktionen](#) im AWS Lake Formation Developer Guide.

- API-Einheiten finden Sie [UpdateTableObjects](#) in der AWS CLI Befehlsreferenz.

Lambda-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Lambda Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-layer-version-permission

Das folgende Codebeispiel zeigt die Verwendung `add-layer-version-permission`.

AWS CLI

Um einer Layer-Version Berechtigungen hinzuzufügen

Im folgenden `add-layer-version-permission` Beispiel wird dem angegebenen Konto die Erlaubnis erteilt, Version 1 des Layers zu verwenden `my-layer`.

```
aws lambda add-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --action lambda:GetLayerVersion \  
  --principal 123456789012 \  
  --version-number 1
```

Ausgabe:

```
{  
  "RevisionId": "35d87451-f796-4a3f-a618-95a3671b0a0c",  
  "Statement":  
  {  
    "Sid": "xaccount",  
    "Effect": "Allow",  
    "Principal": {  
      "AWS": "arn:aws:iam::210987654321:root"  
    },  
    "Action": "lambda:GetLayerVersion",  
    "Resource": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1"  
  }  
}
```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [AddLayerVersionPermission](#) in der AWS CLI Befehlsreferenz.

add-permission

Das folgende Codebeispiel zeigt, wie man es benutzt `add-permission`.

AWS CLI

So fügen Sie einer vorhandenen Lambda-Funktion Berechtigungen hinzu

Das folgende `add-permission` Beispiel erteilt dem SNS Amazon-Service die Erlaubnis, eine Funktion mit dem Namen `my-function` aufzurufen.

```
aws lambda add-permission \  
  --function-name my-function \  
  --action lambda:InvokeFunction \  
  --statement-id sns \  
  --principal sns.amazonaws.com
```

Ausgabe:

```
{  
  "Statement":  
  {  
    "Sid": "sns",  
    "Effect": "Allow",  
    "Principal": {  
      "Service": "sns.amazonaws.com"  
    },  
    "Action": "lambda:InvokeFunction",  
    "Resource": "arn:aws:lambda:us-east-2:123456789012:function:my-function"  
  }  
}
```

Weitere Informationen finden Sie unter [Using Resource-based Policies for AWS Lambda im Lambda Developer Guide AWS](#).

- API-Einheiten finden Sie unter [AddPermission](#) Befehlsreferenz. AWS CLI

create-alias

Das folgende Codebeispiel zeigt, wie man es benutzt `create-alias`.

AWS CLI

Um einen Alias für eine Lambda-Funktion zu erstellen

Im folgenden `create-alias` Beispiel wird ein Alias mit dem Namen `erstelltLIVE`, der auf Version 1 der `my-function` Lambda-Funktion verweist.

```
aws lambda create-alias \  
  --function-name my-function \  
  --name erstelltLIVE \  
  --version 1
```

```
--function-name my-function \  
--description "alias for live version of function" \  
--function-version 1 \  
--name LIVE
```

Ausgabe:

```
{  
  "FunctionVersion": "1",  
  "Name": "LIVE",  
  "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",  
  "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",  
  "Description": "alias for live version of function"  
}
```

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliasen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateAlias](#).AWS CLI

create-event-source-mapping

Das folgende Codebeispiel zeigt, wie man es benutzt `create-event-source-mapping`.

AWS CLI

Um eine Zuordnung zwischen einer Ereignisquelle und einer AWS Lambda-Funktion zu erstellen

Das folgende `create-event-source-mapping` Beispiel erstellt eine Zuordnung zwischen einer SQS Warteschlange und der `my-function` Lambda-Funktion.

```
aws lambda create-event-source-mapping \  
  --function-name my-function \  
  --batch-size 5 \  
  --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

Ausgabe:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",
```



```
"LastModified": 1569284520.333,  
"BatchSize": 5,  
"State": "Creating",  
"FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
"EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

Weitere Informationen finden Sie unter [AWS Lambda Event Source Mapping](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [CreateEventSourceMapping](#) in der AWS CLI Befehlsreferenz.

create-function

Das folgende Codebeispiel zeigt, wie man es benutzt `create-function`.

AWS CLI

Eine Lambda-Funktion erstellen

Im folgenden Beispiel für `create-function` wird eine Lambda-Funktion mit dem Namen `my-function` erstellt.

```
aws lambda create-function \  
  --function-name my-function \  
  --runtime nodejs18.x \  
  --zip-file fileb://my-function.zip \  
  --handler my-function.handler \  
  --role arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-tges6bf4
```

Inhalt von `my-function.zip`:

```
This file is a deployment package that contains your function code and any  
dependencies.
```

Ausgabe:

```
{  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
}
```

```
"CodeSha256": "PFn4S+er27qk+UuZSTKEQfNKG/XNn7QJs90mJgq6oH8=",
"FunctionName": "my-function",
"CodeSize": 308,
"RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",
"MemorySize": 128,
"FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
"Version": "$LATEST",
"Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",
"Timeout": 3,
"LastModified": "2023-10-14T22:26:11.234+0000",
"Handler": "my-function.handler",
"Runtime": "nodejs18.x",
"Description": ""
}
```

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateFunction](#) in der AWS CLI Befehlsreferenz.

delete-alias

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-alias`.

AWS CLI

Um einen Alias einer Lambda-Funktion zu löschen

Im folgenden `delete-alias` Beispiel wird der Alias mit dem Namen `LIVE` aus der `my-function` Lambda-Funktion gelöscht.

```
aws lambda delete-alias \  
  --function-name my-function \  
  --name LIVE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliasen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteAlias](#).AWS CLI

delete-event-source-mapping

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-event-source-mapping`.

AWS CLI

Um die Zuordnung zwischen einer Ereignisquelle und einer AWS Lambda-Funktion zu löschen

Im folgenden `delete-event-source-mapping` Beispiel wird die Zuordnung zwischen einer SQS Warteschlange und der `my-function` Lambda-Funktion gelöscht.

```
aws lambda delete-event-source-mapping \  
  --uuid a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569285870.271,  
  "BatchSize": 5,  
  "State": "Deleting",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

Weitere Informationen finden Sie unter [AWS Lambda Event Source Mapping](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [DeleteEventSourceMapping](#) in der AWS CLI Befehlsreferenz.

delete-function-concurrency

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-function-concurrency`.

AWS CLI

Um das reservierte Limit für gleichzeitige Ausführung aus einer Funktion zu entfernen

Im folgenden `delete-function-concurrency` Beispiel wird das reservierte Limit für gleichzeitige Ausführung aus der Funktion gelöscht. `my-function`

```
aws lambda delete-function-concurrency \  
  --function-name my-function
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Parallelität für eine Lambda-Funktion reservieren im Lambda Developer Guide AWS](#).

- API-Einheiten finden Sie unter [DeleteFunctionConcurrency](#) Befehlsreferenz.AWS CLI

delete-function-event-invoke-config

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-function-event-invoke-config`.

AWS CLI

Um eine asynchrone Aufrufkonfiguration zu löschen

Im folgenden `delete-function-event-invoke-config` Beispiel wird die asynchrone Aufrufkonfiguration für den GREEN Alias der angegebenen Funktion gelöscht.

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteFunctionEventInvokeConfig](#).AWS CLI

delete-function

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-function`.

AWS CLI

Beispiel 1: Eine Lambda-Funktion anhand des Funktionsnamens löschen

Im folgenden Beispiel für `delete-function` wird die Lambda-Funktion `my-function` durch Angabe des Funktionsnamens gelöscht.

```
aws lambda delete-function \  
  --function-name my-function
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um eine Lambda-Funktion nach Funktion zu löschen ARN

Im folgenden `delete-function` Beispiel wird die Lambda-Funktion gelöscht, die `my-function` durch Angabe der Funktion benannt wurde. ARN

```
aws lambda delete-function \  
  --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 3: Um eine Lambda-Funktion nach einer Teilfunktion zu löschen ARN

Im folgenden `delete-function` Beispiel wird die Lambda-Funktion gelöscht, die `my-function` durch Angabe des Partialbereichs der Funktion benannt wurde. ARN

```
aws lambda delete-function \  
  --function-name 123456789012:function:my-function
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS - Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteFunction](#) in der AWS CLI Befehlsreferenz.

delete-layer-version

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-layer-version`.

AWS CLI

Um eine Version einer Lambda-Schicht zu löschen

Im folgenden `delete-layer-version` Beispiel wird Version 2 des genannten Layers gelöscht. `my-layer`

```
aws lambda delete-layer-version \  
  --layer-name my-layer \  
  --version-number 2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [DeleteLayerVersion](#) in der AWS CLI Befehlsreferenz.

delete-provisioned-concurrency-config

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-provisioned-concurrency-config`.

AWS CLI

Um eine bereitgestellte Parallelitätskonfiguration zu löschen

Im folgenden `delete-provisioned-concurrency-config` Beispiel wird die bereitgestellte Parallelitätskonfiguration für den GREEN Alias der angegebenen Funktion gelöscht.

```
aws lambda delete-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier GREEN
```

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteProvisionedConcurrencyConfig](#).AWS CLI

get-account-settings

Das folgende Codebeispiel zeigt, wie man es benutzt `get-account-settings`.

AWS CLI

Um Details zu Ihrem Konto in einer AWS Region abzurufen

Im folgenden `get-account-settings` Beispiel werden die Lambda-Grenzwerte und Nutzungsinformationen für Ihr Konto angezeigt.

```
aws lambda get-account-settings
```

Ausgabe:

```
{  
  "AccountLimit": {  
    "CodeSizeUnzipped": 262144000,  
    "UnreservedConcurrentExecutions": 1000,
```

```

    "ConcurrentExecutions": 1000,
    "CodeSizeZipped": 52428800,
    "TotalCodeSize": 80530636800
  },
  "AccountUsage": {
    "FunctionCount": 4,
    "TotalCodeSize": 9426
  }
}

```

Weitere Informationen finden Sie unter [AWS Lambda Limits](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [GetAccountSettings](#) in der AWS CLI Befehlsreferenz.

get-alias

Das folgende Codebeispiel zeigt, wie man es benutzt `get-alias`.

AWS CLI

Um Details zu einem Funktionsalias abzurufen

Im folgenden `get-alias` Beispiel werden Details für den Alias angezeigt, der in LIVE der `my-function` Lambda-Funktion benannt ist.

```

aws lambda get-alias \
  --function-name my-function \
  --name LIVE

```

Ausgabe:

```

{
  "FunctionVersion": "3",
  "Name": "LIVE",
  "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
  "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",
  "Description": "alias for live version of function"
}

```

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliasen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [GetAlias](#).AWS CLI

get-event-source-mapping

Das folgende Codebeispiel zeigt, wie man es benutzt `get-event-source-mapping`.

AWS CLI

Um Details zu einer Ereignisquellenzuordnung abzurufen

Das folgende `get-event-source-mapping` Beispiel zeigt die Details für die Zuordnung zwischen einer SQS Warteschlange und der `my-function` Lambda-Funktion.

```
aws lambda get-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Ausgabe:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569284520.333,  
  "BatchSize": 5,  
  "State": "Enabled",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

Weitere Informationen finden Sie unter [AWS Lambda Event Source Mapping](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [GetEventSourceMapping](#) in der AWS CLI Befehlsreferenz.

get-function-concurrency

Das folgende Codebeispiel zeigt, wie man es benutzt `get-function-concurrency`.

AWS CLI

Um die reservierte Parallelitätseinstellung für eine Funktion anzuzeigen

Im folgenden `get-function-concurrency` Beispiel wird die reservierte Parallelitätseinstellung für die angegebene Funktion abgerufen.


```
aws lambda get-function-concurrency \  
  --function-name my-function
```

Ausgabe:

```
{  
  "ReservedConcurrentExecutions": 250  
}
```

- API-Einheiten finden Sie unter [GetFunctionConcurrency AWS CLI-Befehlsreferenz](#).

get-function-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-function-configuration`.

AWS CLI

Um die versionsspezifischen Einstellungen einer Lambda-Funktion abzurufen

Im folgenden `get-function-configuration` Beispiel werden die Einstellungen für Version 2 der Funktion angezeigt. `my-function`

```
aws lambda get-function-configuration \  
  --function-name my-function:2
```

Ausgabe:

```
{  
  "FunctionName": "my-function",  
  "LastModified": "2019-09-26T20:28:40.438+0000",  
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",  
  "MemorySize": 256,  
  "Version": "2",  
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmLKidWoaCgk=",  
  "Description": ""
```

```
"VpcConfig": {
  "SubnetIds": [],
  "VpcId": "",
  "SecurityGroupIds": []
},
"CodeSize": 304,
"FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",
"Handler": "index.handler"
}
```

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetFunctionConfiguration](#) in der AWS CLI Befehlsreferenz.

get-function-event-invoke-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-function-event-invoke-config`.

AWS CLI

Um eine asynchrone Aufrufkonfiguration anzuzeigen

Im folgenden `get-function-event-invoke-config` Beispiel wird die asynchrone Aufrufkonfiguration für den BLUE Alias der angegebenen Funktion abgerufen.

```
aws lambda get-function-event-invoke-config \
  --function-name my-function:BLUE
```

Ausgabe:

```
{
  "LastModified": 1577824396.653,
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
  "MaximumRetryAttempts": 0,
  "MaximumEventAgeInSeconds": 3600,
  "DestinationConfig": {
    "OnSuccess": {},
    "OnFailure": {
      "Destination": "arn:aws:sqs:us-east-2:123456789012:failed-invocations"
    }
  }
}
```

```
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [GetFunctionEventInvokeConfig](#).AWS CLI

get-function

Das folgende Codebeispiel zeigt, wie man es benutzt `get-function`.

AWS CLI

Informationen zu einer Funktion abrufen

Das folgende Beispiel für `get-function` zeigt Informationen zur Funktion `my-function` an.

```
aws lambda get-function \
  --function-name my-function
```

Ausgabe:

```
{
  "Concurrency": {
    "ReservedConcurrentExecutions": 100
  },
  "Code": {
    "RepositoryType": "S3",
    "Location": "https://awslambda-us-west-2-tasks.s3.us-west-2.amazonaws.com/
snapshots/123456789012/my-function..."
  },
  "Configuration": {
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "Version": "$LATEST",
    "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJm1KidWaaCgk=",
    "FunctionName": "my-function",
    "VpcConfig": {
      "SubnetIds": [],
      "VpcId": "",
      "SecurityGroupIds": []
    },
    "MemorySize": 128,
    "RevisionId": "28f0fb31-5c5c-43d3-8955-03e76c5c1075",
```

```

    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-role-uy319qq",
    "Timeout": 3,
    "LastModified": "2019-09-24T18:20:35.054+0000",
    "Runtime": "nodejs10.x",
    "Description": ""
  }
}

```

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetFunction](#) in der AWS CLI Befehlsreferenz.

get-layer-version-by-arn

Das folgende Codebeispiel zeigt, wie man es benutzt `get-layer-version-by-arn`.

AWS CLI

Um Informationen über eine Lambda-Layer-Version abzurufen

Im folgenden `get-layer-version-by-arn` Beispiel werden Informationen zur Layer-Version mit dem angegebenen Amazon-Ressourcennamen (ARN) angezeigt.

```

aws lambda get-layer-version-by-arn \
  --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2"

```

Ausgabe:

```

{
  "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2",
  "Description": "AWS Lambda SciPy layer for Python 3.11 (scipy-1.1.0, numpy-1.15.4) https://github.com/scipy/scipy/releases/tag/v1.1.0 https://github.com/numpy/numpy/releases/tag/v1.15.4",
  "CreateDate": "2023-10-12T10:09:38.398+0000",
  "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x",
}

```

```
"Content": {
  "CodeSize": 41784542,
  "CodeSha256": "GGmv8ocUw4cly0T8HL0Vx/f5V4RmSCGNjDIslY4VskM=",
  "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/
snapshots/123456789012/..."
},
"Version": 2,
"CompatibleRuntimes": [
  "python3.11"
],
"LicenseInfo": "SciPy: https://github.com/scipy/scipy/blob/main/LICENSE.txt,
NumPy: https://github.com/numpy/numpy/blob/main/LICENSE.txt"
}
```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [GetLayerVersionByArn](#) in der AWS CLI Befehlsreferenz.

get-layer-version-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `get-layer-version-policy`.

AWS CLI

So rufen Sie die Berechtigungsrichtlinie für eine Lambda-Layer-Version ab

Im folgenden `get-layer-version-policy` Beispiel werden Richtlinieninformationen zu Version 1 für den genannten `my-layer` Layer angezeigt.

```
aws lambda get-layer-version-policy \
  --layer-name my-layer \
  --version-number 1
```

Ausgabe:

```
{
  "Policy": {
    "Version": "2012-10-17",
    "Id": "default",
    "Statement":
    [
      {
```

```

        "Sid": "xaccount",
        "Effect": "Allow",
        "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
        "Action": "lambda:GetLayerVersion",
        "Resource": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1"
    }
]
},
"RevisionId": "c68f21d2-cbf0-4026-90f6-1375ee465cd0"
}

```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [GetLayerVersionPolicy](#) in der AWS CLI Befehlsreferenz.

get-layer-version

Das folgende Codebeispiel zeigt, wie man es benutzt `get-layer-version`.

AWS CLI

Um Informationen über eine Lambda-Layer-Version abzurufen

Im folgenden `get-layer-version` Beispiel werden Informationen für Version 1 des genannten `my-layer` Layers angezeigt.

```

aws lambda get-layer-version \
  --layer-name my-layer \
  --version-number 1

```

Ausgabe:

```

{
  "Content": {
    "Location": "https://awslambda-us-east-2-layers.s3.us-east-2.amazonaws.com/snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?versionId=27iWyA73cCAYqyH...",
    "CodeSha256": "tv9jJ0+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",
    "CodeSize": 169
  },
  "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
  "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1",
  "Description": "My Python layer",
}

```

```
"CreateDate": "2018-11-14T23:03:52.894+0000",
"Version": 1,
"LicenseInfo": "MIT",
"CompatibleRuntimes": [
  "python3.10",
  "python3.11"
]
}
```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [GetLayerVersion](#) in der AWS CLI Befehlsreferenz.

get-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `get-policy`.

AWS CLI

Um die ressourcenbasierte IAM Richtlinie für eine Funktion, Version oder einen Alias abzurufen

Im folgenden `get-policy` Beispiel werden Richtlinieninformationen zur `my-function` Lambda-Funktion angezeigt.

```
aws lambda get-policy \
  --function-name my-function
```

Ausgabe:

```
{
  "Policy": {
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
      {
        "Sid": "iot-events",
        "Effect": "Allow",
        "Principal": {"Service": "iotevents.amazonaws.com"},
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-west-2:123456789012:function:my-
function"
      }
    ]
  }
}
```

```
    ]
  },
  "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668"
}
```

Weitere Informationen finden Sie unter [Using Resource-based Policies for AWS Lambda im Lambda Developer Guide AWS](#) .

- API-Einheiten finden Sie unter [GetPolicy](#) Befehlsreferenz.AWS CLI

get-provisioned-concurrency-config

Das folgende Codebeispiel zeigt, wie man es benutzt `get-provisioned-concurrency-config`.

AWS CLI

So zeigen Sie eine bereitgestellte Parallelitätskonfiguration an

Im folgenden `get-provisioned-concurrency-config` Beispiel werden Details zur bereitgestellten Parallelitätskonfiguration für den BLUE Alias der angegebenen Funktion angezeigt.

```
aws lambda get-provisioned-concurrency-config \
  --function-name my-function \
  --qualifier BLUE
```

Ausgabe:

```
{
  "RequestedProvisionedConcurrentExecutions": 100,
  "AvailableProvisionedConcurrentExecutions": 100,
  "AllocatedProvisionedConcurrentExecutions": 100,
  "Status": "READY",
  "LastModified": "2019-12-31T20:28:49+0000"
}
```

- API-Einheiten finden Sie unter [GetProvisionedConcurrencyConfig AWS CLI](#) Befehlsreferenz.

invoke

Das folgende Codebeispiel zeigt, wie man es benutzt `invoke`.

AWS CLI

Beispiel 1: Eine Lambda-Funktion synchron aufrufen

Im folgenden Beispiel für `invoke` wird die Funktion `my-function` synchron aufgerufen. Die `cli-binary-format` Option ist erforderlich, wenn Sie AWS CLI Version 2 verwenden. Weitere Informationen finden Sie unter [AWS CLI Unterstützte globale Befehlszeilenoptionen](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

```
aws lambda invoke \  
  --function-name my-function \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

Ausgabe:

```
{  
  "ExecutedVersion": "$LATEST",  
  "StatusCode": 200  
}
```

Weitere Informationen finden Sie unter [Synchroner Aufruf](#) im AWS -Lambda-Entwicklerhandbuch.

Beispiel 2: Eine Lambda-Funktion asynchron aufrufen

Im folgenden Beispiel für `invoke` wird die Funktion `my-function` asynchron aufgerufen. Die `cli-binary-format` Option ist erforderlich, wenn Sie AWS CLI Version 2 verwenden. Weitere Informationen finden Sie unter [AWS CLI Unterstützte globale Befehlszeilenoptionen](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

```
aws lambda invoke \  
  --function-name my-function \  
  --invocation-type Event \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

Ausgabe:

```
{
```

```
"statusCode": 202
}
```

Weitere Informationen finden Sie unter [Asynchroner Aufruf](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie unter In der AWS CLI Befehlsreferenz [aufrufen](#).

list-aliases

Das folgende Codebeispiel zeigt, wie Sie es verwenden `list-aliases`.

AWS CLI

Um die Liste der Aliase für eine Lambda-Funktion abzurufen

Im folgenden `list-aliases` Beispiel wird eine Liste der Aliase für die `my-function` Lambda-Funktion angezeigt.

```
aws lambda list-aliases \
  --function-name my-function
```

Ausgabe:

```
{
  "Aliases": [
    {
      "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:BETA",
      "RevisionId": "a410117f-ab16-494e-8035-7e204bb7933b",
      "FunctionVersion": "2",
      "Name": "BETA",
      "Description": "alias for beta version of function"
    },
    {
      "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:LIVE",
      "RevisionId": "21d40116-f8b1-40ba-9360-3ea284da1bb5",
      "FunctionVersion": "1",
      "Name": "LIVE",
      "Description": "alias for live version of function"
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliasen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [ListAliases](#).AWS CLI

list-event-source-mappings

Das folgende Codebeispiel zeigt die Verwendung `list-event-source-mappings`.

AWS CLI

Um die Zuordnungen der Ereignisquellen für eine Funktion aufzulisten

Im folgenden `list-event-source-mappings` Beispiel wird eine Liste der Ereignisquellenzuordnungen für die `my-function` Lambda-Funktion angezeigt.

```
aws lambda list-event-source-mappings \
  --function-name my-function
```

Ausgabe:

```
{
  "EventSourceMappings": [
    {
      "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "StateTransitionReason": "USER_INITIATED",
      "LastModified": 1569284520.333,
      "BatchSize": 5,
      "State": "Enabled",
      "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
      "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"
    }
  ]
}
```

Weitere Informationen finden Sie unter [AWS Lambda Event Source Mapping](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [ListEventSourceMappings](#) in der AWS CLI Befehlsreferenz.

list-function-event-invoke-configs

Das folgende Codebeispiel zeigt die Verwendung `list-function-event-invoke-configs`.

AWS CLI

Um eine Liste mit asynchronen Aufrufkonfigurationen anzuzeigen

Das folgende `list-function-event-invoke-configs` Beispiel listet die asynchronen Aufrufkonfigurationen für die angegebene Funktion auf.

```
aws lambda list-function-event-invoke-configs \  
  --function-name my-function
```

Ausgabe:

```
{  
  "FunctionEventInvokeConfigs": [  
    {  
      "LastModified": 1577824406.719,  
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-  
function:GREEN",  
      "MaximumRetryAttempts": 2,  
      "MaximumEventAgeInSeconds": 1800  
    },  
    {  
      "LastModified": 1577824396.653,  
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-  
function:BLUE",  
      "MaximumRetryAttempts": 0,  
      "MaximumEventAgeInSeconds": 3600  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [ListFunctionEventInvokeConfigs AWS CLI](#) Befehlsreferenz.

list-functions

Das folgende Codebeispiel zeigt die Verwendung `list-functions`.

AWS CLI

Eine Liste der Lambda-Funktionen abrufen

Im folgenden Beispiel für `list-functions` wird eine Liste aller Funktionen für den aktuellen Benutzer angezeigt.

```
aws lambda list-functions
```

Ausgabe:

```
{
  "Functions": [
    {
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "Version": "$LATEST",
      "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbl/RM10fT/I=",
      "FunctionName": "helloworld",
      "MemorySize": 128,
      "RevisionId": "1718e831-badf-4253-9518-d0644210af7b",
      "CodeSize": 294,
      "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:helloworld",
      "Handler": "helloworld.handler",
      "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",
      "Timeout": 3,
      "LastModified": "2023-09-23T18:32:33.857+0000",
      "Runtime": "nodejs18.x",
      "Description": ""
    },
    {
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "Version": "$LATEST",
      "CodeSha256": "sU0cJ2/h0ZevwV/1TxCuQqK3gDZP3i8gUoqUUVrMvY6E=",
      "FunctionName": "my-function",
      "VpcConfig": {
        "SubnetIds": [],
        "VpcId": ""
      }
    }
  ]
}
```

```
        "SecurityGroupIds": [],
      },
      "MemorySize": 256,
      "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
      "CodeSize": 266,
      "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
      "Handler": "index.handler",
      "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
      "Timeout": 3,
      "LastModified": "2023-10-01T16:47:28.490+0000",
      "Runtime": "nodejs18.x",
      "Description": ""
    },
    {
      "Layers": [
        {
          "CodeSize": 41784542,
          "Arn": "arn:aws:lambda:us-west-2:420165488524:layer:AWSLambda-
Python37-SciPy1x:2"
        },
        {
          "CodeSize": 4121,
          "Arn": "arn:aws:lambda:us-
west-2:123456789012:layer:pythonLayer:1"
        }
      ],
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "Version": "$LATEST",
      "CodeSha256": "ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08mI=",
      "FunctionName": "my-python-function",
      "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
      },
      "MemorySize": 128,
      "RevisionId": "80b4eabc-acf7-4ea8-919a-e874c213707d",
      "CodeSize": 299,
      "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
python-function",
    }
  ]
}
```

```
        "Handler": "lambda_function.lambda_handler",
        "Role": "arn:aws:iam::123456789012:role/service-role/my-python-function-
role-z5g7dr6n",
        "Timeout": 3,
        "LastModified": "2023-10-01T19:40:41.643+0000",
        "Runtime": "python3.11",
        "Description": ""
    }
]
}
```

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListFunctions](#) in der AWS CLI Befehlsreferenz.

list-layer-versions

Das folgende Codebeispiel zeigt die Verwendung `list-layer-versions`.

AWS CLI

Um die Versionen einer AWS Lambda-Schicht aufzulisten

Im folgenden `list-layers-versions` Beispiel werden Informationen zu den Versionen für den genannten `my-layer` Layer angezeigt.

```
aws lambda list-layer-versions \
  --layer-name my-layer
```

Ausgabe:

```
{
  "Layers": [
    {
      "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
      "Version": 2,
      "Description": "My layer",
      "CreateDate": "2023-11-15T00:37:46.592+0000",
      "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [ListLayerVersions](#) in der AWS CLI Befehlsreferenz.

list-layers

Das folgende Codebeispiel zeigt die Verwendung `list-layers`.

AWS CLI

Um die Layer aufzulisten, die mit der Laufzeit Ihrer Funktion kompatibel sind

Im folgenden `list-layers` Beispiel werden Informationen zu Layern angezeigt, die mit der Python 3.11-Laufzeit kompatibel sind.

```
aws lambda list-layers \
  --compatible-runtime python3.11
```

Ausgabe:

```
{
  "Layers": [
    {
      "LayerName": "my-layer",
      "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
      "LatestMatchingVersion": {
        "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
        "Version": 2,
        "Description": "My layer",
        "CreateDate": "2023-11-15T00:37:46.592+0000",
        "CompatibleRuntimes": [
          "python3.10",
          "python3.11"
        ]
      }
    }
  ]
}
```



```
}
```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [ListLayers](#) in der AWS CLI Befehlsreferenz.

list-provisioned-concurrency-configs

Das folgende Codebeispiel zeigt die Verwendung `list-provisioned-concurrency-configs`.

AWS CLI

Um eine Liste der bereitgestellten Parallelitätskonfigurationen abzurufen

Im folgenden `list-provisioned-concurrency-configs` Beispiel werden die bereitgestellten Parallelitätskonfigurationen für die angegebene Funktion aufgeführt.

```
aws lambda list-provisioned-concurrency-configs \  
  --function-name my-function
```

Ausgabe:

```
{  
  "ProvisionedConcurrencyConfigs": [  
    {  
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-  
function:GREEN",  
      "RequestedProvisionedConcurrentExecutions": 100,  
      "AvailableProvisionedConcurrentExecutions": 100,  
      "AllocatedProvisionedConcurrentExecutions": 100,  
      "Status": "READY",  
      "LastModified": "2019-12-31T20:29:00+0000"  
    },  
    {  
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-  
function:BLUE",  
      "RequestedProvisionedConcurrentExecutions": 100,  
      "AvailableProvisionedConcurrentExecutions": 100,  
      "AllocatedProvisionedConcurrentExecutions": 100,  
      "Status": "READY",  
      "LastModified": "2019-12-31T20:28:49+0000"  
    }  
  ]  
}
```

```
}
```

- API-Einheiten finden Sie unter [ListProvisionedConcurrencyConfigs AWS CLI](#) Befehlsreferenz.

list-tags

Das folgende Codebeispiel zeigt die Verwendung `list-tags`.

AWS CLI

Um die Liste der Tags für eine Lambda-Funktion abzurufen

Im folgenden `list-tags` Beispiel werden die der `my-function` Lambda-Funktion angehängten Tags angezeigt.

```
aws lambda list-tags \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Ausgabe:

```
{  
  "Tags": {  
    "Category": "Web Tools",  
    "Department": "Sales"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging Lambda Functions im AWS Lambda Developer Guide](#).

- API-Einheiten finden Sie [ListTags](#) in AWS CLI der Befehlsreferenz.

list-versions-by-function

Das folgende Codebeispiel zeigt die Verwendung `list-versions-by-function`.

AWS CLI

Um eine Liste von Versionen einer Funktion abzurufen

Im folgenden `list-versions-by-function` Beispiel wird die Liste der Versionen für die `my-function` Lambda-Funktion angezeigt.

```
aws lambda list-versions-by-function \  
--function-name my-function
```

Ausgabe:

```
{  
  "Versions": [  
    {  
      "TracingConfig": {  
        "Mode": "PassThrough"  
      },  
      "Version": "$LATEST",  
      "CodeSha256": "sU0cJ2/h0ZevwV/1TxCuQqK3gDZP3i8gUoqUUVRmY6E=",  
      "FunctionName": "my-function",  
      "VpcConfig": {  
        "SubnetIds": [],  
        "VpcId": "",  
        "SecurityGroupIds": []  
      },  
      "MemorySize": 256,  
      "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",  
      "CodeSize": 266,  
      "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-  
function:$LATEST",  
      "Handler": "index.handler",  
      "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-  
role-uy3l9qqq",  
      "Timeout": 3,  
      "LastModified": "2019-10-01T16:47:28.490+0000",  
      "Runtime": "nodejs10.x",  
      "Description": ""  
    },  
    {  
      "TracingConfig": {  
        "Mode": "PassThrough"  
      },  
      "Version": "1",  
      "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",  
      "FunctionName": "my-function",  
      "VpcConfig": {  
        "SubnetIds": [],  
        "VpcId": "",  
        "SecurityGroupIds": []  
      }  
    }  
  ]  
}
```

```
    },
    "MemorySize": 256,
    "RevisionId": "949c8914-012e-4795-998c-e467121951b1",
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:1",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qq",
    "Timeout": 3,
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "Runtime": "nodejs10.x",
    "Description": "new version"
  },
  {
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "Version": "2",
    "CodeSha256": "sU0cJ2/h0ZevwV/1TxCuQqK3gDZP3i8gUoqUUVRmY6E=",
    "FunctionName": "my-function",
    "VpcConfig": {
      "SubnetIds": [],
      "VpcId": "",
      "SecurityGroupIds": []
    },
    "MemorySize": 256,
    "RevisionId": "cd669f21-0f3d-4e1c-9566-948837f2e2ea",
    "CodeSize": 266,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:2",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qq",
    "Timeout": 3,
    "LastModified": "2019-10-01T16:47:28.490+0000",
    "Runtime": "nodejs10.x",
    "Description": "newer version"
  }
]
}
```

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliasen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [ListVersionsByFunction](#).AWS CLI

publish-layer-version

Das folgende Codebeispiel zeigt die Verwendung `publish-layer-version`.

AWS CLI

So erstellen Sie eine Lambda-Layer-Version

Im folgenden `publish-layer-version` Beispiel wird eine neue Layer-Version der Python-Bibliothek erstellt. Der Befehl ruft den Layer-Inhalt einer Datei ab, die `layer.zip` im angegebenen S3-Bucket benannt ist.

```
aws lambda publish-layer-version \  
  --layer-name my-layer \  
  --description "My Python layer" \  
  --license-info "MIT" \  
  --content S3Bucket=lambda-layers-us-west-2-123456789012,S3Key=layer.zip \  
  --compatible-runtimes python3.10 python3.11
```

Ausgabe:

```
{  
  "Content": {  
    "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/  
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?  
versionId=27iWyA73cCAYqyH...",  
    "CodeSha256": "tv9jJ0+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",  
    "CodeSize": 169  
  },  
  "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer",  
  "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1",  
  "Description": "My Python layer",  
  "CreateDate": "2023-11-14T23:03:52.894+0000",  
  "Version": 1,  
  "LicenseInfo": "MIT",  
  "CompatibleRuntimes": [  
    "python3.10",  
    "python3.11"  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [PublishLayerVersion](#) in der AWS CLI Befehlsreferenz.

publish-version

Das folgende Codebeispiel zeigt die Verwendung `publish-version`.

AWS CLI

Um eine neue Version einer Funktion zu veröffentlichen

Das folgende `publish-version` Beispiel veröffentlicht eine neue Version der `my-function` Lambda-Funktion.

```
aws lambda publish-version \
  --function-name my-function
```

Ausgabe:

```
{
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "CodeSha256": "dBG9m8SGdm1Ejw/JYX1hhvCrAv5TxvXsbl/RM1r0fT/I=",
  "FunctionName": "my-function",
  "CodeSize": 294,
  "RevisionId": "f31d3d39-cc63-4520-97d4-43cd44c94c20",
  "MemorySize": 128,
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:3",
  "Version": "2",
  "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",
  "Timeout": 3,
  "LastModified": "2019-09-23T18:32:33.857+0000",
  "Handler": "my-function.handler",
  "Runtime": "nodejs10.x",
  "Description": ""
}
```

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliesen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [PublishVersion](#).AWS CLI

put-function-concurrency

Das folgende Codebeispiel zeigt die Verwendung `put-function-concurrency`.

AWS CLI

Um ein reserviertes Parallelitätslimit für eine Funktion zu konfigurieren

Im folgenden `put-function-concurrency` Beispiel werden 100 reservierte gleichzeitige Ausführungen für die Funktion konfiguriert. `my-function`

```
aws lambda put-function-concurrency \  
  --function-name my-function \  
  --reserved-concurrent-executions 100
```

Ausgabe:

```
{  
  "ReservedConcurrentExecutions": 100  
}
```

Weitere Informationen finden Sie unter [Parallelität für eine Lambda-Funktion reservieren im Lambda](#) Developer Guide AWS .

- API-Einheiten finden Sie unter [PutFunctionConcurrency](#) Befehlsreferenz.AWS CLI

put-function-event-invoke-config

Das folgende Codebeispiel zeigt die Verwendung `put-function-event-invoke-config`.

AWS CLI

Um die Fehlerbehandlung für asynchrone Aufrufe zu konfigurieren

Im folgenden `put-function-event-invoke-config` Beispiel wird ein maximales Ereignisalter von einer Stunde festgelegt und Wiederholungsversuche für die angegebene Funktion deaktiviert.

```
aws lambda put-function-event-invoke-config \  
  --function-name my-function \  
  --maximum-event-age-in-seconds 3600 \  
  --maximum-retry-attempts 0
```

Ausgabe:

```
{  
  "LastModified": 1573686021.479,  
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:  
$LATEST",  
  "MaximumRetryAttempts": 0,  
  "MaximumEventAgeInSeconds": 3600,  
  "DestinationConfig": {  
    "OnSuccess": {},  
    "OnFailure": {}  
  }  
}
```

- API-Einheiten finden Sie unter [PutFunctionEventInvokeConfig AWS CLI Befehlsreferenz](#).

put-provisioned-concurrency-config

Das folgende Codebeispiel zeigt die Verwendung `put-provisioned-concurrency-config`.

AWS CLI

Um bereitgestellte Parallelität zuzuweisen

Im folgenden `put-provisioned-concurrency-config` Beispiel werden 100 bereitgestellte Parallelität für den Alias der angegebenen Funktion zugewiesen. BLUE

```
aws lambda put-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier BLUE \  
  --provisioned-concurrent-executions 100
```

Ausgabe:

```
{  
  "Requested ProvisionedConcurrentExecutions": 100,
```



```
"Allocated ProvisionedConcurrentExecutions": 0,  
"Status": "IN_PROGRESS",  
"LastModified": "2019-11-21T19:32:12+0000"  
}
```

- API-Einheiten finden Sie unter [PutProvisionedConcurrencyConfig](#) Befehlsreferenz. AWS CLI

remove-layer-version-permission

Das folgende Codebeispiel zeigt die Verwendung `remove-layer-version-permission`.

AWS CLI

Um Layer-Versionsberechtigungen zu löschen

Im folgenden `remove-layer-version-permission` Beispiel wird die Berechtigung für ein Konto gelöscht, eine Layer-Version zu konfigurieren.

```
aws lambda remove-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --version-number 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS Lambda Layers](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [RemoveLayerVersionPermission](#) in der AWS CLI Befehlsreferenz.

remove-permission

Das folgende Codebeispiel zeigt die Verwendung `remove-permission`.

AWS CLI

So entfernen Sie Berechtigungen aus einer vorhandenen Lambda-Funktion

Im folgenden `remove-permission` Beispiel wird die Berechtigung zum Aufrufen einer Funktion mit dem Namen entfernt. `my-function`

```
aws lambda remove-permission \  
  --function-name my-function \  
  --statement-id xaccount
```

```
--function-name my-function \  
--statement-id sns
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using Resource-based Policies for AWS Lambda im Lambda Developer Guide AWS](#) .

- APIEinzelheiten finden Sie unter [RemovePermission](#) Befehlsreferenz.AWS CLI

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer vorhandenen Lambda-Funktion Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel wird der angegebenen Lambda-Funktion ein Tag mit dem Schlüsselnamen `DEPARTMENT` und `Department A` dem Wert von hinzugefügt.

```
aws lambda tag-resource \  
--resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
--tags "DEPARTMENT=Department A"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Lambda Functions im AWS Lambda Developer Guide](#).

- APIEinzelheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer vorhandenen Lambda-Funktion zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem `DEPARTMENT` Schlüsselnamen-Tag aus der `my-function` Lambda-Funktion entfernt.

```
aws lambda untag-resource \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --tag-keys DEPARTMENT
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Lambda Functions im AWS Lambda Developer Guide](#).

- API-Einheiten finden Sie [UntagResource](#) in AWS CLI der Befehlsreferenz.

update-alias

Das folgende Codebeispiel zeigt die Verwendung `update-alias`.

AWS CLI

Um einen Funktionsalias zu aktualisieren

Im folgenden `update-alias` Beispiel wird der Aliasname so aktualisiert `LIVE`, dass er auf Version 3 der `my-function` Lambda-Funktion verweist.

```
aws lambda update-alias \  
  --function-name my-function \  
  --function-version 3 \  
  --name LIVE
```

Ausgabe:

```
{  
  "FunctionVersion": "3",  
  "Name": "LIVE",  
  "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",  
  "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",  
  "Description": "alias for live version of function"  
}
```

Weitere Informationen finden Sie unter [Konfiguration von AWS Lambda-Funktionsaliasen](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [UpdateAlias](#).AWS CLI

update-event-source-mapping

Das folgende Codebeispiel zeigt die Verwendung `update-event-source-mapping`.

AWS CLI

Um die Zuordnung zwischen einer Ereignisquelle und einer AWS Lambda-Funktion zu aktualisieren

Im folgenden `update-event-source-mapping` Beispiel wird die Batchgröße in der angegebenen Zuordnung auf 8 aktualisiert.

```
aws lambda update-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --batch-size 8
```

Ausgabe:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569284520.333,  
  "BatchSize": 8,  
  "State": "Updating",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

Weitere Informationen finden Sie unter [AWS Lambda Event Source Mapping](#) im AWS Lambda Developer Guide.

- API-Einheiten finden Sie [UpdateEventSourceMapping](#) in der AWS CLI Befehlsreferenz.

update-function-code

Das folgende Codebeispiel zeigt die Verwendung `update-function-code`.

AWS CLI

Den Code einer Lambda-Funktion aktualisieren

Im folgenden `update-function-code` Beispiel wird der Code der unveröffentlichten Version (\$LATEST) der `my-function` Funktion durch den Inhalt der angegebenen ZIP-Datei ersetzt.

```
aws lambda update-function-code \  
  --function-name my-function \  
  --zip-file fileb://my-function.zip
```

Ausgabe:

```
{  
  "FunctionName": "my-function",  
  "LastModified": "2019-09-26T20:28:40.438+0000",  
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",  
  "MemorySize": 256,  
  "Version": "$LATEST",  
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy319qq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqWR716pZ2dpkn/0J1FrzJmLKidWoaCgk=",  
  "Description": "",  
  "VpcConfig": {  
    "SubnetIds": [],  
    "VpcId": "",  
    "SecurityGroupIds": []  
  },  
  "CodeSize": 304,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "Handler": "index.handler"  
}
```

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateFunctionCode](#) in der AWS CLI Befehlsreferenz.

update-function-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-function-configuration`.

AWS CLI

Die Konfiguration einer Funktion ändern

Im folgenden `update-function-configuration` Beispiel wird die Speichergröße für die unveröffentlichte Version (`$LATEST`) der Funktion auf 256 MB geändert. `my-function`

```
aws lambda update-function-configuration \  
  --function-name my-function \  
  --memory-size 256
```

Ausgabe:

```
{  
  "FunctionName": "my-function",  
  "LastModified": "2019-09-26T20:28:40.438+0000",  
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",  
  "MemorySize": 256,  
  "Version": "$LATEST",  
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy319qq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJm1KidWoaCgk=",  
  "Description": "",  
  "VpcConfig": {  
    "SubnetIds": [],  
    "VpcId": "",  
    "SecurityGroupIds": []  
  },  
  "CodeSize": 304,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "Handler": "index.handler"  
}
```

Weitere Informationen finden Sie unter [Konfigurieren von AWS -Lambda-Funktionen](#) im AWS -Lambda-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateFunctionConfiguration](#) in der AWS CLI Befehlsreferenz.

update-function-event-invoke-config

Das folgende Codebeispiel zeigt die Verwendung `update-function-event-invoke-config`.

AWS CLI

Um eine asynchrone Aufrufkonfiguration zu aktualisieren

Im folgenden `update-function-event-invoke-config` Beispiel wird der vorhandenen asynchronen Aufrufkonfiguration für die angegebene Funktion ein Ziel für den Fall eines Fehlers hinzugefügt.

```
aws lambda update-function-event-invoke-config \  
  --function-name my-function \  
  --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-  
east-2:123456789012:destination"}}'
```

Ausgabe:

```
{  
  "LastModified": 1573687896.493,  
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:  
$LATEST",  
  "MaximumRetryAttempts": 0,  
  "MaximumEventAgeInSeconds": 3600,  
  "DestinationConfig": {  
    "OnSuccess": {},  
    "OnFailure": {  
      "Destination": "arn:aws:sqs:us-east-2:123456789012:destination"  
    }  
  }  
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [UpdateFunctionEventInvokeConfig](#).AWS CLI

License Manager Manager-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit License Manager Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-license-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-license-configuration`.

AWS CLI

Beispiel 1: Um eine Lizenzkonfiguration zu erstellen

Im folgenden `create-license-configuration` Beispiel wird eine Lizenzkonfiguration mit einem festen Limit von 10 Kernen erstellt.

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type Core \  
  --license-count 10 \  
  --license-count-hard-limit
```

Ausgabe:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"  
}
```

Beispiel 2: Um eine Lizenzkonfiguration zu erstellen

Im folgenden `create-license-configuration` Beispiel wird eine Lizenzkonfiguration mit einem Soft-Limit von 100 erstellt vCPUs. Es verwendet eine Regel, um die CPU V-Optimierung zu aktivieren.

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type vCPU \  
  --license-count 100 \  
  --license-rules "#honorVcpuOptimization=true"
```


Ausgabe:

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"
}
```

- API-Einheiten finden Sie [CreateLicenseConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-license-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-license-configuration`.

AWS CLI

Um eine Lizenzkonfiguration zu löschen

Im folgenden `delete-license-configuration` Beispiel wird die angegebene Lizenzkonfiguration gelöscht.

```
aws license-manager delete-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteLicenseConfiguration AWS CLI](#) Befehlsreferenz.

get-license-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-license-configuration`.

AWS CLI

Um Informationen zur Lizenzkonfiguration abzurufen

Im folgenden `get-license-configuration` Beispiel werden Details für die angegebene Lizenzkonfiguration angezeigt.

```
aws license-manager get-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Ausgabe:

```
{
  "LicenseConfigurationId": "lic-38b658717b87478aaa7c00883EXAMPLE",
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE",
  "Name": "my-license-configuration",
  "LicenseCountingType": "vCPU",
  "LicenseRules": [],
  "LicenseCountHardLimit": false,
  "ConsumedLicenses": 0,
  "Status": "AVAILABLE",
  "OwnerAccountId": "123456789012",
  "ConsumedLicenseSummaryList": [
    {
      "ResourceType": "EC2_INSTANCE",
      "ConsumedLicenses": 0
    },
    {
      "ResourceType": "EC2_HOST",
      "ConsumedLicenses": 0
    },
    {
      "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
      "ConsumedLicenses": 0
    }
  ],
  "ManagedResourceSummaryList": [
    {
      "ResourceType": "EC2_INSTANCE",
      "AssociationCount": 0
    },
    {
      "ResourceType": "EC2_HOST",
      "AssociationCount": 0
    },
    {
      "ResourceType": "EC2_AMI",
      "AssociationCount": 2
    },
    {
      "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
      "AssociationCount": 0
    }
  ]
}
```

```
]
}
```

- API-Einheiten finden Sie [GetLicenseConfiguration](#) unter AWS CLI Befehlsreferenz.

get-service-settings

Das folgende Codebeispiel zeigt die Verwendung `get-service-settings`.

AWS CLI

Um die License Manager Manager-Einstellungen abzurufen

Im folgenden `get-service-settings` Beispiel werden die Dienstinstellungen für License Manager in der aktuellen Region angezeigt.

```
aws license-manager get-service-settings
```

Das folgende Beispiel zeigt eine Ausgabe für den Fall, dass die kontenübergreifende Ressourcenerkennung deaktiviert ist.

```
{
  "OrganizationConfiguration": {
    "EnableIntegration": false
  },
  "EnableCrossAccountsDiscovery": false
}
```

Im Folgenden wird eine Beispielausgabe gezeigt, wenn die kontenübergreifende Ressourcensuche aktiviert ist.

```
{
  "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
  "OrganizationConfiguration": {
    "EnableIntegration": true
  },
  "EnableCrossAccountsDiscovery": true
}
```

- API-Einheiten finden Sie [GetServiceSettings](#) in der AWS CLI Befehlsreferenz.

list-associations-for-license-configuration

Das folgende Codebeispiel zeigt die Verwendung `list-associations-for-license-configuration`.

AWS CLI

Um Verknüpfungen für eine Lizenzkonfiguration abzurufen

Im folgenden `list-associations-for-license-configuration` Beispiel werden detaillierte Informationen zu den Zuordnungen der angegebenen Lizenzkonfiguration angezeigt.

```
aws license-manager list-associations-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Ausgabe:

```
{
  "LicenseConfigurationAssociations": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0",
      "ResourceType": "EC2_AMI",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1568825118.617
    },
    {
      "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-0abcdef1234567890",
      "ResourceType": "EC2_AMI",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1568825118.946
    }
  ]
}
```

- API-Einheiten finden Sie [ListAssociationsForLicenseConfiguration](#) unter AWS CLI Befehlsreferenz.

list-license-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-license-configurations`.

AWS CLI

Beispiel 1: Um alle Ihre Lizenzkonfigurationen aufzulisten

Das folgende `list-license-configurations` Beispiel listet alle Ihre Lizenzkonfigurationen auf.

```
aws license-manager list-license-configurations
```

Ausgabe:

```
{
  "LicenseConfigurations": [
    {
      "LicenseConfigurationId": "lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
      "LicenseConfigurationArn": "arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
      "Name": "my-license-configuration",
      "LicenseCountingType": "Core",
      "LicenseRules": [],
      "LicenseCount": 10,
      "LicenseCountHardLimit": true,
      "ConsumedLicenses": 0,
      "Status": "AVAILABLE",
      "OwnerAccountId": "123456789012",
      "ConsumedLicenseSummaryList": [
        {
          "ResourceType": "EC2_INSTANCE",
          "ConsumedLicenses": 0
        },
        {
          "ResourceType": "EC2_HOST",
          "ConsumedLicenses": 0
        },
        {
          "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
          "ConsumedLicenses": 0
        }
      ],
      "ManagedResourceSummaryList": [
        {
          "ResourceType": "EC2_INSTANCE",
          "AssociationCount": 0
        }
      ]
    }
  ]
}
```

```

    },
    {
      "ResourceType": "EC2_HOST",
      "AssociationCount": 0
    },
    {
      "ResourceType": "EC2_AMI",
      "AssociationCount": 0
    },
    {
      "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
      "AssociationCount": 0
    }
  ]
},
{
  ...
}
]
}

```

Beispiel 2: Um eine bestimmte Lizenzkonfiguration aufzulisten

Im folgenden `list-license-configurations` Beispiel wird nur die angegebene Lizenzkonfiguration aufgeführt.

```

aws license-manager list-license-configurations \
  --license-configuration-arns arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE

```

- API-Einheiten finden Sie [ListLicenseConfigurations](#) in der AWS CLI Befehlsreferenz.

list-license-specifications-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-license-specifications-for-resource`.

AWS CLI

Um die Lizenzkonfigurationen für eine Ressource aufzulisten

Das folgende `list-license-specifications-for-resource` Beispiel listet die Lizenzkonfigurationen auf, die dem angegebenen Amazon Machine Image (AMI) zugeordnet sind.

```
aws license-manager list-license-specifications-for-resource \  
  --resource-arn arn:aws:ec2:us-west-2:image/ami-1234567890abcdef0
```

Ausgabe:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"  
}
```

- API-Einheiten finden Sie [ListLicenseSpecificationsForResource](#) unter AWS CLI Befehlsreferenz.

list-resource-inventory

Das folgende Codebeispiel zeigt die Verwendung `list-resource-inventory`.

AWS CLI

Um Ressourcen im Ressourceninventar aufzulisten

Das folgende `list-resource-inventory` Beispiel listet die Ressourcen auf, die mithilfe des Systems Manager Manager-Inventars verwaltet werden.

```
aws license-manager list-resource-inventory
```

Ausgabe:

```
{  
  "ResourceInventoryList": [  
    {  
      "Platform": "Red Hat Enterprise Linux Server",  
      "ResourceType": "EC2Instance",  
      "PlatformVersion": "7.4",  
      "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/  
i-05d3cdfb05bd36376",  
      "ResourceId": "i-05d3cdfb05bd36376",  
      "ResourceOwningAccountId": "1234567890129"  
    }  
  ]  
}
```

```

    },
    {
      "Platform": "Amazon Linux",
      "ResourceType": "EC2Instance",
      "PlatformVersion": "2",
      "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0b1d036cfd4594808",
      "ResourceId": "i-0b1d036cfd4594808",
      "ResourceOwningAccountId": "1234567890129"
    },
    {
      "Platform": "Microsoft Windows Server 2019 Datacenter",
      "ResourceType": "EC2Instance",
      "PlatformVersion": "10.0.17763",
      "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0cdb3b54a2a8246ad",
      "ResourceId": "i-0cdb3b54a2a8246ad",
      "ResourceOwningAccountId": "1234567890129"
    }
  ]
}

```

- API-Einheiten finden Sie [ListResourceInventory](#) unter AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Lizenzkonfiguration aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für die angegebene Lizenzkonfiguration auf.

```

aws license-manager list-tags-for-resource \
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE

```

Ausgabe:

```
{
```



```
    "Tags": [
      {
        "Key": "project",
        "Value": "lima"
      }
    ]
  }
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

list-usage-for-license-configuration

Das folgende Codebeispiel zeigt die Verwendung `list-usage-for-license-configuration`.

AWS CLI

Um die Lizenzen aufzulisten, die für eine Lizenzkonfiguration verwendet werden

Das folgende `list-usage-for-license-configuration` Beispiel listet Informationen über die Ressourcen auf, die Lizenzen für die angegebene Lizenzkonfiguration verwenden. Wenn der Lizenztyp beispielsweise `v` ist CPU, verbrauchen alle Instanzen eine Lizenz pro CPU `v`.

```
aws license-manager list-usage-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Ausgabe:

```
{
  "LicenseConfigurationUsageList": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-04a636d18e83cfac",
      "ResourceType": "EC2_INSTANCE",
      "ResourceStatus": "running",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1570892850.519,
      "ConsumedLicenses": 2
    }
  ]
}
```

- API-Einheiten finden Sie [ListUsageForLicenseConfiguration](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Tag hinzuzufügen: eine Lizenzkonfiguration

Im folgenden `tag-resource` Beispiel wird das angegebene Tag (Schlüsselname und Wert) zur angegebenen Lizenzkonfiguration hinzugefügt.

```
aws license-manager tag-resource \  
  --tags Key=project,Value=lima \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-  
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Lizenzkonfiguration zu entfernen

Im folgenden `untag-resource` Beispiel wird das angegebene Tag (Schlüsselname und Ressource) aus der angegebenen Lizenzkonfiguration entfernt.

```
aws license-manager untag-resource \  
  --tag-keys project \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-  
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-license-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-license-configuration`.

AWS CLI

Um eine Lizenzkonfiguration zu aktualisieren

Im folgenden `update-license-configuration` Beispiel wird die angegebene Lizenzkonfiguration aktualisiert, um das feste Limit aufzuheben.

```
aws license-manager update-license-configuration \  
  --no-license-count-hard-limit \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Im folgenden `update-license-configuration` Beispiel wird die angegebene Lizenzkonfiguration aktualisiert, sodass ihr Status auf `DISABLED` geändert wird.

```
aws license-manager update-license-configuration \  
  --license-configuration-status DISABLED \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UpdateLicenseConfiguration](#) in der AWS CLI Befehlsreferenz.

update-license-specifications-for-resource

Das folgende Codebeispiel zeigt die Verwendung `update-license-specifications-for-resource`.

AWS CLI

Um die Lizenzkonfigurationen für eine Ressource zu aktualisieren

Das folgende `update-license-specifications-for-resource` Beispiel ersetzt die Lizenzkonfiguration, die dem angegebenen Amazon Machine Image (AMI) zugeordnet ist, indem eine Lizenzkonfiguration entfernt und eine weitere hinzugefügt wird.

```
aws license-manager update-license-specifications-for-resource \  
  --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0 \  
  --remove-license-specifications LicenseConfigurationArn=arn:aws:license-  
manager:us-west-2:123456789012:license-  
configuration:lic-38b658717b87478aaa7c00883EXAMPLE \  
  --add-license-specifications LicenseConfigurationArn=arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-42b6deb06e5399a980d555927EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UpdateLicenseSpecificationsForResource](#) in der AWS CLI Befehlsreferenz.

update-service-settings

Das folgende Codebeispiel zeigt die Verwendung `update-service-settings`.

AWS CLI

Um die License Manager Manager-Einstellungen zu aktualisieren

Das folgende `update-service-settings` Beispiel ermöglicht die kontenübergreifende Ressourcensuche für License Manager in der aktuellen AWS Region. Der Amazon S3 S3-Bucket ist der Resource Data Sync, der für das Systems Manager Manager-Inventar erforderlich ist.

```
aws license-manager update-service-settings \  
  --organization-configuration EnableIntegration=true \  
  --enable-cross-accounts-discovery \  
  --s3-bucket-arn arn:aws:s3:::aws-license-manager-service-abcd1234EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UpdateServiceSettings](#) in der AWS CLI Befehlsreferenz.

Lightsail-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Lightsail Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

allocate-static-ip

Das folgende Codebeispiel zeigt die Verwendung `allocate-static-ip`.

AWS CLI

Um eine statische IP zu erstellen

Im folgenden `allocate-static-ip` Beispiel wird die angegebene statische IP erstellt, die an eine Instanz angehängt werden kann.

```
aws lightsail allocate-static-ip \  
  --static-ip-name StaticIp-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "b5d06d13-2f19-4683-889f-dEXAMPLEed79",  
      "resourceName": "StaticIp-1",  
      "resourceType": "StaticIp",  
      "createdAt": 1571071325.076,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationType": "AllocateStaticIp",  
      "status": "Succeeded",  
    }  
  ]  
}
```

```

        "statusChangedAt": 1571071325.274
      }
    ]
  }

```

- API-Einheiten finden Sie [AllocateStaticIp](#) unter AWS CLI Befehlsreferenz.

attach-disk

Das folgende Codebeispiel zeigt die Verwendung `attach-disk`.

AWS CLI

Um eine Blockspeicherfestplatte an eine Instanz anzuhängen

Im folgenden `attach-disk` Beispiel wird eine Festplatte `WordPress_Multisite-1` mit dem Festplattenpfad von `Disk-1` an eine Instanz angehängt `/dev/xvdf`

```

aws lightsail attach-disk \
  --disk-name Disk-1 \
  --disk-path /dev/xvdf \
  --instance-name WordPress_Multisite-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "10a08267-19ce-43be-b913-6EXAMPLE7e80",
      "resourceName": "Disk-1",
      "resourceType": "Disk",
      "createdAt": 1571071465.472,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "WordPress_Multisite-1",
      "operationType": "AttachDisk",
      "status": "Started",
      "statusChangedAt": 1571071465.472
    },
  ],
}

```

```

    {
      "id": "2912c477-5295-4539-88c9-bEXAMPLEd1f0",
      "resourceName": "WordPress_Multisite-1",
      "resourceType": "Instance",
      "createdAt": 1571071465.474,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "Disk-1",
      "operationType": "AttachDisk",
      "status": "Started",
      "statusChangedAt": 1571071465.474
    }
  ]
}

```

- API-Einheiten finden Sie unter [AttachDisk AWS CLI](#) Befehlsreferenz.

attach-instances-to-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `attach-instances-to-load-balancer`.

AWS CLI

Um Instances an einen Load Balancer anzuhängen

Im folgenden `attach-instances-to-load-balancer` Beispiel werden Instances MEAN-1, MEAN-2, und an den Load MEAN-3 Balancer angehängt. LoadBalancer-1

```

aws lightsail attach-instances-to-load-balancer \
  --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \
  --load-balancer-name LoadBalancer-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "8055d19d-abb2-40b9-b527-1EXAMPLE3c7b",
      "resourceName": "LoadBalancer-1",

```

```
    "resourceType": "LoadBalancer",
    "createdAt": 1571071699.892,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "MEAN-2",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.892
  },
  {
    "id": "c35048eb-8538-456a-a118-0EXAMPLEfb73",
    "resourceName": "MEAN-2",
    "resourceType": "Instance",
    "createdAt": 1571071699.887,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "LoadBalancer-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.887
  },
  {
    "id": "910d09e0-adc5-4372-bc2e-0EXAMPLEd891",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571071699.882,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "MEAN-3",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.882
  },
  {
    "id": "178b18ac-43e8-478c-9bed-1EXAMPLE4755",
```



```
    "resourceName": "MEAN-3",
    "resourceType": "Instance",
    "createdAt": 1571071699.901,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "LoadBalancer-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.901
  },
  {
    "id": "fb62536d-2a98-4190-a6fc-4EXAMPLE7470",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571071699.885,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "MEAN-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.885
  },
  {
    "id": "787dac0d-f98d-46c3-8571-3EXAMPLE5a85",
    "resourceName": "MEAN-1",
    "resourceType": "Instance",
    "createdAt": 1571071699.901,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "LoadBalancer-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.901
  }
]
```

```
}
```

- API-Einheiten finden Sie unter [AttachInstancesToLoadBalancer AWS CLI](#) Befehlsreferenz.

attach-load-balancer-tls-certificate

Das folgende Codebeispiel zeigt die Verwendung `attach-load-balancer-tls-certificate`.

AWS CLI

Um ein TLS-Zertifikat an einen Load Balancer anzuhängen

Im folgenden `attach-load-balancer-tls-certificate` Beispiel wird das Load TLS Balancer-Zertifikat an den Load Balancer `Certificate2` angehängt. `LoadBalancer-1`

```
aws lightsail attach-load-balancer-tls-certificate \  
  --certificate-name Certificate2 \  
  --load-balancer-name LoadBalancer-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "cf1ad6e3-3cbb-4b8a-a7f2-3EXAMPLEa118",  
      "resourceName": "LoadBalancer-1",  
      "resourceType": "LoadBalancer",  
      "createdAt": 1571072255.416,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationDetails": "Certificate2",  
      "operationType": "AttachLoadBalancerTlsCertificate",  
      "status": "Succeeded",  
      "statusChangedAt": 1571072255.416  
    },  
    {  
      "id": "dae1bcfb-d531-4c06-b4ea-bEXAMPLEc04e",  
      "resourceName": "Certificate2",  
      "resourceType": "LoadBalancerTlsCertificate",
```

```

    "createdAt": 1571072255.416,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "LoadBalancer-1",
    "operationType": "AttachLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1571072255.416
  }
]
}

```

- API-Einheiten finden Sie unter Befehlsreferenz [AttachLoadBalancerTlsCertificate](#).AWS CLI

attach-static-ip

Das folgende Codebeispiel zeigt die Verwendung `attach-static-ip`.

AWS CLI

Um eine statische IP an eine Instanz anzuhängen

Im folgenden `attach-static-ip` Beispiel wird eine statische IP an die Instanz `StaticIp-1` `MEAN-1` angehängt.

```

aws lightsail attach-static-ip \
  --static-ip-name StaticIp-1 \
  --instance-name MEAN-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "45e6fa13-4808-4b8d-9292-bEXAMPLE20b2",
      "resourceName": "StaticIp-1",
      "resourceType": "StaticIp",
      "createdAt": 1571072569.375,
      "location": {
        "availabilityZone": "all",

```

```

        "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-1",
    "operationType": "AttachStaticIp",
    "status": "Succeeded",
    "statusChangedAt": 1571072569.375
},
{
    "id": "9ee09a17-863c-4e51-8a6d-3EXAMPLE5475",
    "resourceName": "MEAN-1",
    "resourceType": "Instance",
    "createdAt": 1571072569.376,
    "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "StaticIp-1",
    "operationType": "AttachStaticIp",
    "status": "Succeeded",
    "statusChangedAt": 1571072569.376
}
]
}

```

- API-Einheiten finden Sie [AttachStaticIp](#) in der AWS CLI Befehlsreferenz.

close-instance-public-ports

Das folgende Codebeispiel zeigt die Verwendung `close-instance-public-ports`.

AWS CLI

Um Firewall-Ports für eine Instanz zu schließen

Im folgenden `close-instance-public-ports` Beispiel wird der TCP Port 22 einer Instanz geschlossen `MEAN-2`.

```

aws lightsail close-instance-public-ports \
  --instance-name MEAN-2 \
  --port-info fromPort=22,protocol=TCP,toPort=22

```

Ausgabe:

```
{
  "operation": {
    "id": "4f328636-1c96-4649-ae6d-1EXAMPLEf446",
    "resourceName": "MEAN-2",
    "resourceType": "Instance",
    "createdAt": 1571072845.737,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "22/tcp",
    "operationType": "CloseInstancePublicPorts",
    "status": "Succeeded",
    "statusChangedAt": 1571072845.737
  }
}
```

- API-Einheiten finden Sie [CloseInstancePublicPorts](#) in der AWS CLI Befehlsreferenz.

copy-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-snapshot`.

AWS CLI

Beispiel 1: Um einen Snapshot innerhalb derselben AWS Region zu kopieren

Im folgenden `copy-snapshot` Beispiel wird ein Instanz-Snapshot `MEAN-1-1571075291` als Instanz-Snapshot `MEAN-1-Copy` innerhalb derselben AWS Region kopiert `us-west-2`.

```
aws lightsail copy-snapshot \
  --source-snapshot-name MEAN-1-1571075291 \
  --target-snapshot-name MEAN-1-Copy \
  --source-region us-west-2
```

Ausgabe:

```
{
  "operations": [
```

```

    {
      "id": "ced16fc1-f401-4556-8d82-1EXAMPLEb982",
      "resourceName": "MEAN-1-Copy",
      "resourceType": "InstanceSnapshot",
      "createdAt": 1571075581.498,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "us-west-2:MEAN-1-1571075291",
      "operationType": "CopySnapshot",
      "status": "Started",
      "statusChangedAt": 1571075581.498
    }
  ]
}

```

Weitere Informationen finden Sie unter [Kopieren von Schnapshots von einer AWS Region in eine andere in Amazon Lightsail im Lightsail Dev Guide](#).

Beispiel 2: Um einen Snapshot von einer Region in eine andere zu kopieren AWS

Im folgenden copy-snapshot Beispiel wird der Instanz-Snapshot MEAN-1-1571075291 als Instanz-Snapshot MEAN-1-1571075291-Copy von AWS der Region us-west-2 nach kopiertus-east-1.

```

aws lightsail copy-snapshot \
  --source-snapshot-name MEAN-1-1571075291 \
  --target-snapshot-name MEAN-1-1571075291-Copy \
  --source-region us-west-2 \
  --region us-east-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "91116b79-119c-4451-b44a-dEXAMPLEd97b",
      "resourceName": "MEAN-1-1571075291-Copy",
      "resourceType": "InstanceSnapshot",
      "createdAt": 1571075695.069,

```

```

        "location": {
            "availabilityZone": "all",
            "regionName": "us-east-1"
        },
        "isTerminal": false,
        "operationDetails": "us-west-2:MEAN-1-1571075291",
        "operationType": "CopySnapshot",
        "status": "Started",
        "statusChangedAt": 1571075695.069
    }
]
}

```

Weitere Informationen finden Sie unter [Kopieren von Schnappschüssen von einer AWS Region in eine andere in Amazon Lightsail im Lightsail Dev Guide](#).

Beispiel 3: Um einen automatischen Snapshot innerhalb derselben Region zu kopieren AWS

Im folgenden copy-snapshot Beispiel wird der automatische Snapshot 2019-10-14 der Instanz WordPress-1 als manueller Snapshot WordPress-1-10142019 in der AWS Region kopiert us-west-2.

```

aws lightsail copy-snapshot \
  --source-resource-name WordPress-1 \
  --restore-date 2019-10-14 \
  --target-snapshot-name WordPress-1-10142019 \
  --source-region us-west-2

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "be3e6754-cd1d-48e6-ad9f-2EXAMPLE1805",
      "resourceName": "WordPress-1-10142019",
      "resourceType": "InstanceSnapshot",
      "createdAt": 1571082412.311,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,

```

```

        "operationDetails": "us-west-2:WordPress-1",
        "operationType": "CopySnapshot",
        "status": "Started",
        "statusChangedAt": 1571082412.311
    }
]
}

```

Weitere Informationen finden Sie unter [Automatisches Erstellen von Snapshots von Instances oder Festplatten in Amazon Lightsail im Lightsail Dev Guide](#).

Beispiel 4: Um einen automatischen Snapshot von einer Region in eine andere zu kopieren AWS

Im folgenden copy-snapshot Beispiel wird ein automatischer Snapshot 2019-10-14 der Instanz WordPress-1 als manueller Snapshot WordPress-1-10142019 aus der AWS Region in us-west-2 kopiertus-east-1.

```

aws lightsail copy-snapshot \
  --source-resource-name WordPress-1 \
  --restore-date 2019-10-14 \
  --target-snapshot-name WordPress-1-10142019 \
  --source-region us-west-2 \
  --region us-east-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "dffa128b-0b07-476e-b390-bEXAMPLE3775",
      "resourceName": "WordPress-1-10142019",
      "resourceType": "InstanceSnapshot",
      "createdAt": 1571082493.422,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-east-1"
      },
      "isTerminal": false,
      "operationDetails": "us-west-2:WordPress-1",
      "operationType": "CopySnapshot",
      "status": "Started",
      "statusChangedAt": 1571082493.422
    }
  ]
}

```



```
]
}
```

Weitere Informationen finden Sie unter [Automatisches Erstellen von Snapshots von Instances oder Festplatten in Amazon Lightsail im Lightsail Dev Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [CopySnapshot](#) AWS CLI

create-disk-from-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-disk-from-snapshot`.

AWS CLI

Um ein Laufwerk aus einem Festplatten-Snapshot zu erstellen

Im folgenden `create-disk-from-snapshot` Beispiel wird eine Blockspeicherfestplatte erstellt, die nach `Disk-2` dem angegebenen Blockspeicher-Festplatten-Snapshot benannt ist. Die Festplatte wird in der angegebenen AWS Region und Availability Zone mit 32 GB Speicherplatz erstellt.

```
aws lightsail create-disk-from-snapshot \
  --disk-name Disk-2 \
  --disk-snapshot-name Disk-1-1566839161 \
  --availability-zone us-west-2a \
  --size-in-gb 32
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "d42b605d-5ef1-4b4a-8791-7a3e8b66b5e7",
      "resourceName": "Disk-2",
      "resourceType": "Disk",
      "createdAt": 1569624941.471,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateDiskFromSnapshot",
    }
  ]
}
```

```
        "status": "Started",
        "statusChangedAt": 1569624941.791
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Erstellen einer Blockspeicherfestplatte aus einem Snapshot in Amazon Lightsail im Lightsail Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz [CreateDiskFromSnapshot](#).AWS CLI

create-disk-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-disk-snapshot`.

AWS CLI

Beispiel 1: Um einen Snapshot einer Festplatte zu erstellen

Im folgenden `create-disk-snapshot` Beispiel wird ein Snapshot mit dem Namen `DiskSnapshot-1` des angegebenen Blockspeicherdatenträgers erstellt.

```
aws lightsail create-disk-snapshot \
  --disk-name Disk-1 \
  --disk-snapshot-name DiskSnapshot-1
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "fa74c6d2-03a3-4f42-a7c7-792f124d534b",
      "resourceName": "DiskSnapshot-1",
      "resourceType": "DiskSnapshot",
      "createdAt": 1569625129.739,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "Disk-1",
      "operationType": "CreateDiskSnapshot",
      "status": "Started",
```

```

        "statusChangedAt": 1569625129.739
    },
    {
        "id": "920a25df-185c-4528-87cd-7b85f5488c06",
        "resourceName": "Disk-1",
        "resourceType": "Disk",
        "createdAt": 1569625129.739,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": false,
        "operationDetails": "DiskSnapshot-1",
        "operationType": "CreateDiskSnapshot",
        "status": "Started",
        "statusChangedAt": 1569625129.739
    }
]
}

```

Beispiel 2: Um einen Snapshot der Systemfestplatte einer Instanz zu erstellen

Im folgenden `create-disk-snapshot` Beispiel wird ein Snapshot der Systemfestplatte der angegebenen Instanz erstellt.

```

aws lightsail create-disk-snapshot \
  --instance-name WordPress-1 \
  --disk-snapshot-name SystemDiskSnapshot-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "f508cf1c-6597-42a6-a4c3-4aebd75af0d9",
      "resourceName": "SystemDiskSnapshot-1",
      "resourceType": "DiskSnapshot",
      "createdAt": 1569625294.685,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,

```

```

    "operationDetails": "WordPress-1",
    "operationType": "CreateDiskSnapshot",
    "status": "Started",
    "statusChangedAt": 1569625294.685
  },
  {
    "id": "0bb9f712-da3b-4d99-b508-3bf871d989e5",
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "createdAt": 1569625294.685,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "SystemDiskSnapshot-1",
    "operationType": "CreateDiskSnapshot",
    "status": "Started",
    "statusChangedAt": 1569625294.685
  }
]
}

```

Weitere Informationen finden Sie unter [Snapshots in Amazon Lightsail](#) und [Erstellen eines Snapshots eines Instance-Root-Volumes in Amazon Lightsail im Lightsail Developer Guide](#).

- Einzelheiten API finden Sie in der Befehlsreferenz. [CreateDiskSnapshot](#) AWS CLI

create-disk

Das folgende Codebeispiel zeigt die Verwendung `create-disk`.

AWS CLI

Um eine Blockspeicherfestplatte zu erstellen

Im folgenden `create-disk` Beispiel wird eine Blockspeicherfestplatte `Disk-1` in der angegebenen AWS Region und Availability Zone mit 32 GB Speicherplatz erstellt.

```

aws lightsail create-disk \
  --disk-name Disk-1 \
  --availability-zone us-west-2a \
  --size-in-gb 32

```

Ausgabe:

```
{
  "operations": [
    {
      "id": "1c85e2ec-86ba-4697-b936-77f4d3dc013a",
      "resourceName": "Disk-1",
      "resourceType": "Disk",
      "createdAt": 1569449220.36,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateDisk",
      "status": "Started",
      "statusChangedAt": 1569449220.588
    }
  ]
}
```

- API-Einheiten finden Sie [CreateDisk](#) in der AWS CLI Befehlsreferenz.

create-domain-entry

Das folgende Codebeispiel zeigt die Verwendung `create-domain-entry`.

AWS CLI

Um einen Domäneintrag (DNSRecord) zu erstellen

Im folgenden `create-domain-entry` Beispiel wird ein DNS Datensatz (A) für den Apex der angegebenen Domain erstellt, der auf die IP-Adresse einer Instanz verweist.

Hinweis: Die domänenbezogenen API Operationen von Lightsail sind nur in der Region verfügbar. `us-east-1` Wenn Ihr CLI Profil für die Verwendung einer anderen Region konfiguriert ist, müssen Sie den `--region us-east-1` Parameter angeben, sonst schlägt der Befehl fehl.

```
aws lightsail create-domain-entry \
  --region us-east-1 \
  --domain-name example.com \
  --domain-entry name=example.com,type=A,target=192.0.2.0
```

Ausgabe:

```
{
  "operation": {
    "id": "5be4494d-56f4-41fc-8730-693dcd0ef9e2",
    "resourceName": "example.com",
    "resourceType": "Domain",
    "createdAt": 1569865296.519,
    "location": {
      "availabilityZone": "all",
      "regionName": "global"
    },
    "isTerminal": true,
    "operationType": "CreateDomainEntry",
    "status": "Succeeded",
    "statusChangedAt": 1569865296.519
  }
}
```

Weitere Informationen finden Sie unter [DNSAmazon Lightsail](#) und [Creating a DNS zone to manage your domain DNS records in Amazon Lightsail](#) im [Lightsail Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [CreateDomainEntry](#) AWS CLI

create-domain

Das folgende Codebeispiel zeigt die Verwendung `create-domain`.

AWS CLI

Um eine Domain (DNSZone) zu erstellen

Im folgenden `create-domain` Beispiel wird eine DNS Zone für die angegebene Domäne erstellt.

Hinweis: Die domänenbezogenen API Operationen von Lightsail sind nur in der Region verfügbar. `us-east-1` Wenn Ihr CLI Profil für die Verwendung einer anderen Region konfiguriert ist, müssen Sie den `--region us-east-1` Parameter angeben, sonst schlägt der Befehl fehl.

```
aws lightsail create-domain \
  --region us-east-1 \
  --domain-name example.com
```

Ausgabe:

```
{
  "operation": {
    "id": "64e522c8-9ae1-4c05-9b65-3f237324dc34",
    "resourceName": "example.com",
    "resourceType": "Domain",
    "createdAt": 1569864291.92,
    "location": {
      "availabilityZone": "all",
      "regionName": "global"
    },
    "isTerminal": true,
    "operationType": "CreateDomain",
    "status": "Succeeded",
    "statusChangedAt": 1569864292.109
  }
}
```

Weitere Informationen finden Sie unter [DNSAmazon Lightsail](#) und [Creating a DNS zone to manage your domain DNS records in Amazon Lightsail](#) im [Lightsail Developer Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [CreateDomain](#) AWS CLI

create-instance-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-instance-snapshot`.

AWS CLI

Um einen Snapshot einer Instanz zu erstellen

Im folgenden `create-instance-snapshot` Beispiel wird ein Snapshot von der angegebenen Instanz erstellt.

```
aws lightsail create-instance-snapshot \
  --instance-name WordPress-1 \
  --instance-snapshot-name WordPress-Snapshot-1
```

Ausgabe:

```
{
```

```
"operations": [  
  {  
    "id": "4c3db559-9dd0-41e7-89c0-2cb88c19786f",  
    "resourceName": "WordPress-Snapshot-1",  
    "resourceType": "InstanceSnapshot",  
    "createdAt": 1569866438.48,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": false,  
    "operationDetails": "WordPress-1",  
    "operationType": "CreateInstanceSnapshot",  
    "status": "Started",  
    "statusChangedAt": 1569866438.48  
  },  
  {  
    "id": "c04fdc45-2981-488c-88b5-d6d2fd759a6a",  
    "resourceName": "WordPress-1",  
    "resourceType": "Instance",  
    "createdAt": 1569866438.48,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": false,  
    "operationDetails": "WordPress-Snapshot-1",  
    "operationType": "CreateInstanceSnapshot",  
    "status": "Started",  
    "statusChangedAt": 1569866438.48  
  }  
]  
}
```

- API-Einheiten finden Sie [CreateInstanceSnapshot](#) unter AWS CLI Befehlsreferenz.

create-instances-from-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-instances-from-snapshot`.

AWS CLI

Um eine Instanz aus einem Snapshot zu erstellen

Im folgenden `create-instances-from-snapshot` Beispiel wird mithilfe des USD \$12-Bundles eine Instanz aus dem angegebenen Instanz-Snapshot in der angegebenen AWS Region und Availability Zone erstellt.

Hinweis: Das von Ihnen angegebene Paket muss mindestens den Spezifikationen des Bundles der ursprünglichen Quell-Instance entsprechen, die zur Erstellung des Snapshots verwendet wurde.

```
aws lightsail create-instances-from-snapshot \  
  --instance-snapshot-name WordPress-1-1569866208 \  
  --instance-names WordPress-2 \  
  --availability-zone us-west-2a \  
  --bundle-id small_3_0
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "003f8271-b711-464d-b9b8-7f3806cb496e",  
      "resourceName": "WordPress-2",  
      "resourceType": "Instance",  
      "createdAt": 1569865914.908,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateInstancesFromSnapshot",  
      "status": "Started",  
      "statusChangedAt": 1569865914.908  
    }  
  ]  
}
```

- API-Einheiten finden Sie [CreateInstancesFromSnapshot](#) in der AWS CLI Befehlsreferenz.

create-instances

Das folgende Codebeispiel zeigt die Verwendung `create-instances`.

AWS CLI

Beispiel 1: Um eine einzelne Instanz zu erstellen

Im folgenden `create-instances` Beispiel wird eine Instanz in der angegebenen AWS Region und Availability Zone mithilfe des WordPress Blueprints und des Pakets im Wert von 5,00\$ USD erstellt.

```
aws lightsail create-instances \  
  --instance-names Instance-1 \  
  --availability-zone us-west-2a \  
  --blueprint-id wordpress \  
  --bundle-id nano_3_0
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "9a77158f-7be3-4d6d-8054-cf5ae2b720cc",  
      "resourceName": "Instance-1",  
      "resourceType": "Instance",  
      "createdAt": 1569447986.061,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateInstance",  
      "status": "Started",  
      "statusChangedAt": 1569447986.061  
    }  
  ]  
}
```

Beispiel 2: Um mehrere Instanzen gleichzeitig zu erstellen

Im folgenden `create-instances` Beispiel werden drei Instanzen in der angegebenen AWS Region und Availability Zone mithilfe des WordPress Blueprints und des Pakets im Wert von 5,00 USD USD erstellt.

```
aws lightsail create-instances \  
  --instance-names Instance-1 Instance-2 Instance-3 \  
  --availability-zone us-west-2a \  
  --blueprint-id wordpress \  
  --bundle-id nano_3_0
```

```
--instance-names {"Instance1","Instance2","Instance3"} \  
--availability-zone us-west-2a \  
--blueprint-id wordpress \  
--bundle-id nano_3_0
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "5492f015-9d2e-48c6-8eea-b516840e6903",  
      "resourceName": "Instance1",  
      "resourceType": "Instance",  
      "createdAt": 1569448780.054,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateInstance",  
      "status": "Started",  
      "statusChangedAt": 1569448780.054  
    },  
    {  
      "id": "c58b5f46-2676-44c8-b95c-3ad375898515",  
      "resourceName": "Instance2",  
      "resourceType": "Instance",  
      "createdAt": 1569448780.054,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateInstance",  
      "status": "Started",  
      "statusChangedAt": 1569448780.054  
    },  
    {  
      "id": "a5ad8006-9bee-4499-9eb7-75e42e6f5882",  
      "resourceName": "Instance3",  
      "resourceType": "Instance",  
      "createdAt": 1569448780.054,  
      "location": {
```

```

        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "CreateInstance",
    "status": "Started",
    "statusChangedAt": 1569448780.054
}
]
}

```

- API-Einheiten finden Sie unter [CreateInstances AWS CLI Befehlsreferenz](#).

create-key-pair

Das folgende Codebeispiel zeigt die Verwendung `create-key-pair`.

AWS CLI

So erstellen Sie ein Schlüsselpaar

Im folgenden `create-key-pair` Beispiel wird ein key pair erstellt, mit dem Sie sich authentifizieren und eine Verbindung zu einer Instance herstellen können.

```

aws lightsail create-key-pair \
  --key-pair-name MyPersonalKeyPair

```

Die Ausgabe stellt den Base64-Wert des privaten Schlüssels bereit, mit dem Sie sich bei Instances authentifizieren können, die das erstellte key pair verwenden. Hinweis: Kopieren Sie den Base64-Wert des privaten Schlüssels und fügen Sie ihn an einem sicheren Ort ein, da Sie ihn später nicht mehr abrufen können.

```

{
  "keyPair": {
    "name": "MyPersonalKeyPair",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:KeyPair/55025c71-198f-403b-b42f-a69433e724fb",
    "supportCode": "621291663362/MyPersonalKeyPair",
    "createdAt": 1569866556.567,
    "location": {
      "availabilityZone": "all",

```

```

    "regionName": "us-west-2"
  },
  "resourceType": "KeyPair"
},
"publicKeyBase64": "ssh-rsa ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCV0xUEwx96amPERH7K1bVT1tTF190mNk6o7m5YVHk9x10dMbDRbFvhtXvw4jz
+BHUgedGUXno6uF7agqxZN01kPLJBIVTW26SSYBJ0tE
+y804UyVsjrBUqCaMXDhmfXpWuLMPwuXhwcKh7e8hwoTfkiX0E6Q1
+KqF/MiA3w6DCjEqvvdI07SiEZJFsuGNfYDDN3w60Re15MUhmn30Jdn4y/
A7NWb3IxL4pPfVE4rgFRKU8n1jp9kwRnLVMVB0WuGXk6n+H6M2f1 ",
"privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCaFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAGTAldBMRAwDgYDVoQHEwdTZWF0dGx1MQ8wDQYDVoQKEwZBbWF6\nnb24xFDASBgNVBASTC01BTSBD
\nBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
\nVQOHEwdTZWF0dGx1MQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBASTC01BTSBDB25z
\nnb2x1MRIwEAYDVoQDEwLUZXN0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
\nYXpvbi5jb20wZGZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
\nIbb30hjZnzcVQAaREXAMPLEMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4\nnnUHVvXyUntneD9+h8Mg9q6q
+auNkyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780EXAMPLELvJx79LjStb
\nNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=\n-----END RSA PRIVATE KEY-----",
"operation": {
  "id": "67f984db-9994-45fe-ad38-59bafcaf82ef",
  "resourceName": "MyPersonalKeyPair",
  "resourceType": "KeyPair",
  "createdAt": 1569866556.567,
  "location": {
    "availabilityZone": "all",
    "regionName": "us-west-2"
  },
  "isTerminal": true,
  "operationType": "CreateKeyPair",
  "status": "Succeeded",
  "statusChangedAt": 1569866556.704
}
}

```

- API-Einheiten finden Sie [CreateKeyPair](#) in der AWS CLI Befehlsreferenz.

create-load-balancer-tls-certificate

Das folgende Codebeispiel zeigt die Verwendung `create-load-balancer-tls-certificate`.

AWS CLI

Um ein TLS Zertifikat für einen Load Balancer zu erstellen

Im folgenden `create-load-balancer-tls-certificate` Beispiel wird ein TLS Zertifikat erstellt, das an den angegebenen Load Balancer angehängt ist. Das erstellte Zertifikat gilt für die angegebenen Domänen. Hinweis: Für einen Load Balancer können nur zwei Zertifikate erstellt werden.

```
aws lightsail create-load-balancer-tls-certificate \  
  --certificate-alternative-names abc.example.com \  
  --certificate-domain-name example.com \  
  --certificate-name MySecondCertificate \  
  --load-balancer-name MyFirstLoadBalancer
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "be663aed-cb46-41e2-9b23-e2f747245bd4",  
      "resourceName": "MySecondCertificate",  
      "resourceType": "LoadBalancerTlsCertificate",  
      "createdAt": 1569867364.971,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationDetails": "MyFirstLoadBalancer",  
      "operationType": "CreateLoadBalancerTlsCertificate",  
      "status": "Succeeded",  
      "statusChangedAt": 1569867365.219  
    },  
    {  
      "id": "f3dfa930-969e-41cc-ac7d-337178716f6d",  
      "resourceName": "MyFirstLoadBalancer",  
      "resourceType": "LoadBalancer",  
      "createdAt": 1569867364.971,  
    }  
  ]  
}
```

```

        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "MySecondCertificate",
        "operationType": "CreateLoadBalancerTlsCertificate",
        "status": "Succeeded",
        "statusChangedAt": 1569867365.219
    }
]
}

```

- API-Einheiten finden Sie [CreateLoadBalancerTlsCertificate](#) in der AWS CLI Befehlsreferenz.

create-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `create-load-balancer`.

AWS CLI

Um einen Load Balancer zu erstellen

Im folgenden `create-load-balancer` Beispiel wird ein Load Balancer mit einem TLS Zertifikat erstellt. Das TLS Zertifikat gilt für die angegebenen Domänen und leitet den Datenverkehr an Instanzen an Port 80 weiter.

```

aws lightsail create-load-balancer \
  --certificate-alternative-names www.example.com test.example.com \
  --certificate-domain-name example.com \
  --certificate-name Certificate-1 \
  --instance-port 80 \
  --load-balancer-name LoadBalancer-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "cc7b920a-83d8-4762-a74e-9174fe1540be",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",

```

```
    "createdAt": 1569867169.406,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "CreateLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1569867169.406
  },
  {
    "id": "658ed43b-f729-42f3-a8e4-3f8024d3c98d",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancerTlsCertificate",
    "createdAt": 1569867170.193,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "LoadBalancer-1",
    "operationType": "CreateLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1569867170.54
  },
  {
    "id": "4757a342-5181-4870-b1e0-227eebc35ab5",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1569867170.193,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "Certificate-1",
    "operationType": "CreateLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1569867170.54
  }
]
}
```


Weitere Informationen finden Sie unter [Lightsail Load Balancers](#) im Lightsail Developer Guide.

- API-Einheiten finden Sie in der Befehlsreferenz. [CreateLoadBalancer](#) AWS CLI

create-relational-database-from-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-relational-database-from-snapshot`.

AWS CLI

Um eine verwaltete Datenbank aus einem Snapshot zu erstellen

Im folgenden `create-relational-database-from-snapshot` Beispiel wird eine verwaltete Datenbank aus dem angegebenen Snapshot in der angegebenen AWS Region und Availability Zone erstellt, wobei das USD Standard-Datenbankpaket für 15\$ verwendet wird. Hinweis: Das Paket, das Sie angeben, muss den Spezifikationen des Pakets der ursprünglichen Quelldatenbank entsprechen, die zur Erstellung des Snapshots verwendet wurde, oder höher sein.

```
aws lightsail create-relational-database-from-snapshot \
  --relational-database-snapshot-name Database-Oregon-1-1566839359 \
  --relational-database-name Database-1 \
  --availability-zone us-west-2a \
  --relational-database-bundle-id micro_1_0 \
  --no-publicly-accessible
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "ad6d9193-9d5c-4ea1-97ae-8fe6de600b4c",
      "resourceName": "Database-1",
      "resourceType": "RelationalDatabase",
      "createdAt": 1569867916.938,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateRelationalDatabaseFromSnapshot",
```

```

        "status": "Started",
        "statusChangedAt": 1569867918.643
      }
    ]
  }

```

- API-Einheiten finden Sie [CreateRelationalDatabaseFromSnapshot](#) in der AWS CLI Befehlsreferenz.

create-relational-database-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-relational-database-snapshot`.

AWS CLI

Um einen Snapshot einer verwalteten Datenbank zu erstellen

Im folgenden `create-relational-database-snapshot` Beispiel wird ein Snapshot der angegebenen verwalteten Datenbank erstellt.

```

aws lightsail create-relational-database-snapshot \
  --relational-database-name Database1 \
  --relational-database-snapshot-name RelationalDatabaseSnapshot1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "853667fb-ea91-4c02-8d20-8fc5fd43b9eb",
      "resourceName": "RelationalDatabaseSnapshot1",
      "resourceType": "RelationalDatabaseSnapshot",
      "createdAt": 1569868074.645,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "Database1",
      "operationType": "CreateRelationalDatabaseSnapshot",
      "status": "Started",
      "statusChangedAt": 1569868074.645
    }
  ]
}

```

```

    },
    {
      "id": "fbafa521-3cac-4be8-9773-1c143780b239",
      "resourceName": "Database1",
      "resourceType": "RelationalDatabase",
      "createdAt": 1569868074.645,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "RelationalDatabaseSnapshot1",
      "operationType": "CreateRelationalDatabaseSnapshot",
      "status": "Started",
      "statusChangedAt": 1569868074.645
    }
  ]
}

```

- API-Einheiten finden Sie [CreateRelationalDatabaseSnapshot](#) unter AWS CLI Befehlsreferenz.

create-relational-database

Das folgende Codebeispiel zeigt die Verwendung `create-relational-database`.

AWS CLI

Um eine verwaltete Datenbank zu erstellen

Im folgenden `create-relational-database` Beispiel wird eine verwaltete Datenbank in der angegebenen AWS Region und Availability Zone mithilfe der My SQL 5.6-Datenbank-Engine (`mysql_5_6`) und des USD Standard-Datenbankpakets für 15\$ (`micro_1_0`) erstellt. Die verwaltete Datenbank ist mit einem Masterbenutzernamen vorbelegt und sie ist nicht öffentlich zugänglich.

```

aws lightsail create-relational-database \
  --relational-database-name Database-1 \
  --availability-zone us-west-2a \
  --relational-database-blueprint-id mysql_5_6 \
  --relational-database-bundle-id micro_1_0 \
  --master-database-name dbmaster \
  --master-username user \
  --no-publicly-accessible

```

Ausgabe:

```
{
  "operations": [
    {
      "id": "b52bedee-73ed-4798-8d2a-9c12df89adcd",
      "resourceName": "Database-1",
      "resourceType": "RelationalDatabase",
      "createdAt": 1569450017.244,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateRelationalDatabase",
      "status": "Started",
      "statusChangedAt": 1569450018.637
    }
  ]
}
```

- API-Einheiten finden Sie [CreateRelationalDatabase](#) in der AWS CLI Befehlsreferenz.

delete-auto-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-auto-snapshot`.

AWS CLI

Um einen automatischen Snapshot zu löschen

Im folgenden `delete-auto-snapshot` Beispiel wird der automatische Snapshot `2019-10-10` der Instanz `WordPress-1` gelöscht.

```
aws lightsail delete-auto-snapshot \
  --resource-name WordPress-1 \
  --date 2019-10-10
```

Ausgabe:

```
{
```

```

    "operations": [
      {
        "id": "31c36e09-3d52-46d5-b6d8-7EXAMPLE534a",
        "resourceName": "WordPress-1",
        "resourceType": "Instance",
        "createdAt": 1571088141.501,
        "location": {
          "availabilityZone": "us-west-2",
          "regionName": "us-west-2"
        },
        "isTerminal": true,
        "operationDetails": "DeleteAutoSnapshot-2019-10-10",
        "operationType": "DeleteAutoSnapshot",
        "status": "Succeeded"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Löschen von automatischen Snapshots von Instances oder Festplatten in Amazon Lightsail im Lightsail Dev Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteAutoSnapshot](#) AWS CLI

delete-disk-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-disk-snapshot`.

AWS CLI

Um einen Snapshot einer Blockspeicherfestplatte zu löschen

Im folgenden `delete-disk-snapshot` Beispiel wird der angegebene Snapshot einer Blockspeicherfestplatte gelöscht

```

aws lightsail delete-disk-snapshot \
  --disk-snapshot-name DiskSnapshot-1

```

Ausgabe:

```

{
  "operations": [
    {

```

```

    "id": "d1e5766d-b81e-4595-ad5d-02afbcccfd5d",
    "resourceName": "DiskSnapshot-1",
    "resourceType": "DiskSnapshot",
    "createdAt": 1569873552.79,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteDiskSnapshot",
    "status": "Succeeded",
    "statusChangedAt": 1569873552.79
  }
]
}

```

- API-Einheiten finden Sie [DeleteDiskSnapshot](#) in der AWS CLI Befehlsreferenz.

delete-disk

Das folgende Codebeispiel zeigt die Verwendung `delete-disk`.

AWS CLI

Um eine Blockspeicherfestplatte zu löschen

Im folgenden `delete-disk` Beispiel wird die angegebene Blockspeicherfestplatte gelöscht.

```

aws lightsail delete-disk \
  --disk-name Disk-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "6378c70f-4d75-4f7a-ab66-730fca0bb2fc",
      "resourceName": "Disk-1",
      "resourceType": "Disk",
      "createdAt": 1569872887.864,
      "location": {
        "availabilityZone": "us-west-2a",

```

```

        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "DeleteDisk",
      "status": "Succeeded",
      "statusChangedAt": 1569872887.864
    }
  ]
}

```

- API-Einheiten finden Sie [DeleteDisk](#) in der AWS CLI Befehlsreferenz.

delete-domain-entry

Das folgende Codebeispiel zeigt die Verwendung `delete-domain-entry`.

AWS CLI

Um einen Domäneintrag (DNSRecord) zu löschen

Im folgenden `delete-domain-entry` Beispiel wird der angegebene Domäneintrag aus einer vorhandenen Domäne gelöscht.

Hinweis: Die domänenbezogenen API Operationen von Lightsail sind nur in der Region verfügbar. `us-east-1` Wenn Ihr CLI Profil für die Verwendung einer anderen Region konfiguriert ist, müssen Sie den `--region us-east-1` Parameter angeben, sonst schlägt der Befehl fehl.

```

aws lightsail delete-domain-entry \
  --region us-east-1 \
  --domain-name example.com \
  --domain-entry name=123.example.com,target=192.0.2.0,type=A

```

Ausgabe:

```

{
  "operation": {
    "id": "06eacd01-d785-420e-8daa-823150c7dca1",
    "resourceName": "example.com ",
    "resourceType": "Domain",
    "createdAt": 1569874157.005,
    "location": {

```

```
        "availabilityZone": "all",
        "regionName": "global"
    },
    "isTerminal": true,
    "operationType": "DeleteDomainEntry",
    "status": "Succeeded",
    "statusChangedAt": 1569874157.005
}
}
```

- API-Einheiten finden Sie [DeleteDomainEntry](#) unter AWS CLI Befehlsreferenz.

delete-domain

Das folgende Codebeispiel zeigt die Verwendung `delete-domain`.

AWS CLI

Um eine Domain (DNSZone) zu löschen

Im folgenden `delete-domain` Beispiel werden die angegebene Domäne und alle Einträge in der Domäne (DNS Datensätze) gelöscht.

Hinweis: Die domänenbezogenen API Operationen von Lightsail sind nur in der Region verfügbar. `us-east-1` Wenn Ihr CLI Profil für die Verwendung einer anderen Region konfiguriert ist, müssen Sie den `--region us-east-1` Parameter angeben, sonst schlägt der Befehl fehl.

```
aws lightsail delete-domain \
  --region us-east-1 \
  --domain-name example.com
```

Ausgabe:

```
{
  "operation": {
    "id": "fcef5265-5af1-4a46-a3d7-90b5e18b9b32",
    "resourceName": "example.com",
    "resourceType": "Domain",
    "createdAt": 1569873788.13,
    "location": {
      "availabilityZone": "all",
```



```
    "regionName": "global"
  },
  "isTerminal": true,
  "operationType": "DeleteDomain",
  "status": "Succeeded",
  "statusChangedAt": 1569873788.13
}
```

- API-Einheiten finden Sie [DeleteDomain](#) unter AWS CLI Befehlsreferenz.

delete-instance-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-instance-snapshot`.

AWS CLI

Titel

Im folgenden `delete-instance-snapshot` Beispiel wird der angegebene Snapshot einer Instanz gelöscht.

```
aws lightsail delete-instance-snapshot \
  --instance-snapshot-name WordPress-1-Snapshot-1
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "14dad182-976a-46c6-bfd4-9480482bf0ea",
      "resourceName": "WordPress-1-Snapshot-1",
      "resourceType": "InstanceSnapshot",
      "createdAt": 1569874524.562,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "DeleteInstanceSnapshot",
      "status": "Succeeded",
      "statusChangedAt": 1569874524.562
    }
  ]
}
```

```
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [DeleteInstanceSnapshot AWS CLI Befehlsreferenz](#).

delete-instance

Das folgende Codebeispiel zeigt die Verwendung `delete-instance`.

AWS CLI

Um eine Instanz zu löschen

Im folgenden `delete-instance` Beispiel wird die angegebene Instanz gelöscht.

```
aws lightsail delete-instance \  
  --instance-name WordPress-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "d77345a3-8f80-4d2e-b47d-aaa622718df2",  
      "resourceName": "Disk-1",  
      "resourceType": "Disk",  
      "createdAt": 1569874357.469,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "WordPress-1",  
      "operationType": "DetachDisk",  
      "status": "Started",  
      "statusChangedAt": 1569874357.469  
    },  
    {  
      "id": "708fa606-2bfd-4e48-a2c1-0b856585b5b1",  
      "resourceName": "WordPress-1",  
      "resourceType": "Instance",
```

```

    "createdAt": 1569874357.465,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "Disk-1",
    "operationType": "DetachDisk",
    "status": "Started",
    "statusChangedAt": 1569874357.465
  },
  {
    "id": "3187e823-8acb-405d-b098-fad5ceb17bec",
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "createdAt": 1569874357.829,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteInstance",
    "status": "Succeeded",
    "statusChangedAt": 1569874357.829
  }
]
}

```

- API-Einheiten finden Sie [DeleteInstance](#) in der AWS CLI Befehlsreferenz.

delete-key-pair

Das folgende Codebeispiel zeigt die Verwendung `delete-key-pair`.

AWS CLI

So löschen Sie ein Schlüsselpaar

Im folgenden `delete-key-pair` Beispiel wird das angegebene key pair gelöscht.

```

aws lightsail delete-key-pair \
  --key-pair-name MyPersonalKeyPair

```

Ausgabe:

```
{
  "operation": {
    "id": "81621463-df38-4810-b866-6e801a15abbf",
    "resourceName": "MyPersonalKeyPair",
    "resourceType": "KeyPair",
    "createdAt": 1569874626.466,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteKeyPair",
    "status": "Succeeded",
    "statusChangedAt": 1569874626.685
  }
}
```

- API-Einheiten finden Sie unter [DeleteKeyPair AWS CLI-Befehlsreferenz](#).

delete-known-host-keys

Das folgende Codebeispiel zeigt die Verwendung `delete-known-host-keys`.

AWS CLI

Um bekannte Hostschlüssel aus einer Instanz zu löschen

Im folgenden `delete-known-host-keys` Beispiel wird der bekannte Hostschlüssel aus der angegebenen Instanz gelöscht.

```
aws lightsail delete-known-host-keys \
  --instance-name Instance-1
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "c61afe9c-45a4-41e6-a97e-d212364da3f5",
      "resourceName": "Instance-1",

```

```

    "resourceType": "Instance",
    "createdAt": 1569874760.201,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteKnownHostKeys",
    "status": "Succeeded",
    "statusChangedAt": 1569874760.201
  }
]
}

```

Weitere Informationen finden Sie unter [Behebung von Verbindungsproblemen mit dem Amazon Lightsail-Browser SSH oder RDP -Client im Lightsail Dev Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteKnownHostKeys](#) AWS CLI

delete-load-balancer-tls-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-load-balancer-tls-certificate`.

AWS CLI

Um ein TLS Zertifikat für einen Load Balancer zu löschen

Im folgenden `delete-load-balancer-tls-certificate` Beispiel wird das angegebene TLS Zertifikat aus dem angegebenen Load Balancer gelöscht.

```

aws lightsail delete-load-balancer-tls-certificate \
  --load-balancer-name MyFirstLoadBalancer \
  --certificate-name MyFirstCertificate

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "50bec274-e45e-4caa-8a69-b763ef636583",
      "resourceName": "MyFirstCertificate",
      "resourceType": "LoadBalancerTlsCertificate",

```

```

    "createdAt": 1569874989.48,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "DeleteLoadBalancerTlsCertificate",
    "status": "Started",
    "statusChangedAt": 1569874989.48
  },
  {
    "id": "78c58cdc-a59a-4b27-8213-500638634a8f",
    "resourceName": "MyFirstLoadBalancer",
    "resourceType": "LoadBalancer",
    "createdAt": 1569874989.48,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "DeleteLoadBalancerTlsCertificate",
    "status": "Started",
    "statusChangedAt": 1569874989.48
  }
]
}

```

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteLoadBalancerTlsCertificate](#).AWS CLI

delete-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `delete-load-balancer`.

AWS CLI

Um einen Load Balancer zu löschen

Im folgenden `delete-load-balancer` Beispiel werden der angegebene Load Balancer und alle zugehörigen Zertifikate gelöscht. TLS

```

aws lightsail delete-load-balancer \
  --load-balancer-name MyFirstLoadBalancer

```

Ausgabe:

```
{
  "operations": [
    {
      "id": "a8c968c7-72a3-4680-a714-af8f03eea535",
      "resourceName": "MyFirstLoadBalancer",
      "resourceType": "LoadBalancer",
      "createdAt": 1569875092.125,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "DeleteLoadBalancer",
      "status": "Succeeded",
      "statusChangedAt": 1569875092.125
    },
    {
      "id": "f91a29fc-8ce3-4e69-a227-ea70ca890bf5",
      "resourceName": "MySecondCertificate",
      "resourceType": "LoadBalancerTlsCertificate",
      "createdAt": 1569875091.938,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "DeleteLoadBalancerTlsCertificate",
      "status": "Started",
      "statusChangedAt": 1569875091.938
    },
    {
      "id": "cf64c060-154b-4eb4-ba57-84e2e41563d6",
      "resourceName": "MyFirstLoadBalancer",
      "resourceType": "LoadBalancer",
      "createdAt": 1569875091.94,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "DeleteLoadBalancerTlsCertificate",
      "status": "Started",
    }
  ]
}
```

```
        "statusChangedAt": 1569875091.94
      }
    ]
  }
```

Weitere Informationen finden Sie im Titel des Handbuchs.

- API-Einheiten finden Sie [DeleteLoadBalancer](#) in der AWS CLI Befehlsreferenz.

delete-relational-database-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-relational-database-snapshot`.

AWS CLI

Um einen Snapshot einer verwalteten Datenbank zu löschen

Im folgenden `delete-relational-database-snapshot` Beispiel wird der angegebene Snapshot einer verwalteten Datenbank gelöscht.

```
aws lightsail delete-relational-database-snapshot \
  --relational-database-snapshot-name Database-Oregon-1-1566839359
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "b99acae8-735b-4823-922f-30af580e3729",
      "resourceName": "Database-Oregon-1-1566839359",
      "resourceType": "RelationalDatabaseSnapshot",
      "createdAt": 1569875293.58,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "DeleteRelationalDatabaseSnapshot",
      "status": "Succeeded",
      "statusChangedAt": 1569875293.58
    }
  ]
}
```



```
}
```

- API-Einheiten finden Sie unter [DeleteRelationalDatabaseSnapshot AWS CLI Befehlsreferenz](#).

delete-relational-database

Das folgende Codebeispiel zeigt die Verwendung `delete-relational-database`.

AWS CLI

Um eine verwaltete Datenbank zu löschen

Im folgenden `delete-relational-database` Beispiel wird die angegebene verwaltete Datenbank gelöscht.

```
aws lightsail delete-relational-database \  
  --relational-database-name Database-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "3b0c41c1-053d-46f0-92a3-14f76141dc86",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1569875210.999,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "DeleteRelationalDatabase",  
      "status": "Started",  
      "statusChangedAt": 1569875210.999  
    },  
    {  
      "id": "01ddeae8-a87a-4a4b-a1f3-092c71bf9180",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1569875211.029,  
      "location": {
```

```

        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "Database-1-FinalSnapshot-1569875210793",
    "operationType": "CreateRelationalDatabaseSnapshot",
    "status": "Started",
    "statusChangedAt": 1569875211.029
},
{
    "id": "74d73681-30e8-4532-974e-1f23cd3f9f73",
    "resourceName": "Database-1-FinalSnapshot-1569875210793",
    "resourceType": "RelationalDatabaseSnapshot",
    "createdAt": 1569875211.029,
    "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "Database-1",
    "operationType": "CreateRelationalDatabaseSnapshot",
    "status": "Started",
    "statusChangedAt": 1569875211.029
}
]
}

```

- API-Einheiten finden Sie unter [DeleteRelationalDatabase AWS CLI-Befehlsreferenz](#).

detach-static-ip

Das folgende Codebeispiel zeigt die Verwendung `detach-static-ip`.

AWS CLI

Um eine statische IP von einer Instanz zu trennen

Im folgenden `detach-static-ip` Beispiel wird die statische IP `StaticIp-1` von jeder angehängten Instanz getrennt.

```
aws lightsail detach-static-ip \
  --static-ip-name StaticIp-1
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "2a43d8a3-9f2d-4fe7-bdd0-eEXAMPLE3cf3",
      "resourceName": "StaticIp-1",
      "resourceType": "StaticIp",
      "createdAt": 1571088261.999,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "MEAN-1",
      "operationType": "DetachStaticIp",
      "status": "Succeeded",
      "statusChangedAt": 1571088261.999
    },
    {
      "id": "41a7d40c-74e8-4d2e-a837-cEXAMPLEf747",
      "resourceName": "MEAN-1",
      "resourceType": "Instance",
      "createdAt": 1571088262.022,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "StaticIp-1",
      "operationType": "DetachStaticIp",
      "status": "Succeeded",
      "statusChangedAt": 1571088262.022
    }
  ]
}
```

- API-Einheiten finden Sie [DetachStaticIp](#) in der AWS CLI Befehlsreferenz.

get-active-names

Das folgende Codebeispiel zeigt die Verwendung `get-active-names`.

AWS CLI

Um aktive Ressourcennamen zu erhalten

Das folgende `get-active-names` Beispiel gibt die Namen der aktiven Ressourcen in der konfigurierten AWS Region zurück.

```
aws lightsail get-active-names
```

Ausgabe:

```
{
  "activeNames": [
    "WordPress-1",
    "StaticIp-1",
    "MEAN-1",
    "Plesk_Hosting_Stack_on_Ubuntu-1"
  ]
}
```

- API-Einheiten finden Sie [GetActiveNames](#) unter AWS CLI Befehlsreferenz.

get-auto-snapshots

Das folgende Codebeispiel zeigt die Verwendung `get-auto-snapshots`.

AWS CLI

Um die verfügbaren automatischen Snapshots für eine Instanz abzurufen

Das folgende `get-auto-snapshots` Beispiel gibt zum Beispiel die verfügbaren automatischen Snapshots zurück. `WordPress-1`

```
aws lightsail get-auto-snapshots \
  --resource-name WordPress-1
```

Ausgabe:

```
{
  "resourceName": "WordPress-1",
  "resourceType": "Instance",
}
```

```
"autoSnapshots": [
  {
    "date": "2019-10-14",
    "createdAt": 1571033872.0,
    "status": "Success",
    "fromAttachedDisks": []
  },
  {
    "date": "2019-10-13",
    "createdAt": 1570947473.0,
    "status": "Success",
    "fromAttachedDisks": []
  },
  {
    "date": "2019-10-12",
    "createdAt": 1570861072.0,
    "status": "Success",
    "fromAttachedDisks": []
  },
  {
    "date": "2019-10-11",
    "createdAt": 1570774672.0,
    "status": "Success",
    "fromAttachedDisks": []
  }
]
```

Weitere Informationen finden Sie unter [Automatisches Erstellen von Snapshots von Instances oder Festplatten in Amazon Lightsail im Lightsail Dev Guide](#).

- API-Einheiten finden Sie in der Befehlsreferenz. [GetAutoSnapshots](#) AWS CLI

get-blueprints

Das folgende Codebeispiel zeigt die Verwendung `get-blueprints`.

AWS CLI

Um die Blueprints für neue Instanzen zu erhalten

Das folgende `get-blueprints` Beispiel zeigt Details zu allen verfügbaren Blueprints, die verwendet werden können, um neue Instances in Amazon Lightsail zu erstellen.

aws lightsail get-blueprints

Ausgabe:

```
{
  "blueprints": [
    {
      "blueprintId": "wordpress",
      "name": "WordPress",
      "group": "wordpress",
      "type": "app",
      "description": "Bitnami, the leaders in application packaging, and Automattic, the experts behind WordPress, have teamed up to offer this official WordPress image. This image is a pre-configured, ready-to-run image for running WordPress on Amazon Lightsail. WordPress is the world's most popular content management platform. Whether it's for an enterprise or small business website, or a personal or corporate blog, content authors can easily create content using its new Gutenberg editor, and developers can extend the base platform with additional features. Popular plugins like Jetpack, Akismet, All in One SEO Pack, WP Mail, Google Analytics for WordPress, and Amazon Polly are all pre-installed in this image. Let's Encrypt SSL certificates are supported through an auto-configuration script.",
      "isActive": true,
      "minPower": 0,
      "version": "6.5.3-0",
      "versionCode": "1",
      "productUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U",
      "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U#pdp-usage",
      "platform": "LINUX_UNIX"
    },
    {
      "blueprintId": "lamp_8_bitnami",
      "name": "LAMP (PHP 8)",
      "group": "lamp_8",
      "type": "app",
      "description": "LAMP with PHP 8.X packaged by Bitnami enables you to quickly start building your websites and applications by providing a coding framework. As a developer, it provides standalone project directories to store your applications. This blueprint is configured for production environments. It includes SSL auto-configuration with Let's Encrypt certificates, and the latest releases of PHP, Apache, and MariaDB on Linux. This application also includes phpMyAdmin, PHP main modules and Composer.",
    }
  ]
}
```

```

        "isActive": true,
        "minPower": 0,
        "version": "8.2.18-4",
        "versionCode": "1",
        "productUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dву",
        "licenseUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dву#pdp-usage",
        "platform": "LINUX_UNIX"
    },
    {
        "blueprintId": "nodejs",
        "name": "Node.js",
        "group": "node",
        "type": "app",
        "description": "Node.js packaged by Bitnami is a pre-configured, ready
to run image for Node.js on Amazon EC2. It includes the latest version of Node.js,
Apache, Python and Redis. The image supports multiple Node.js applications, each
with its own virtual host and project directory. It is configured for production
use and is secure by default, as all ports except HTTP, HTTPS and SSH ports are
closed. Let's Encrypt SSL certificates are supported through an auto-configuration
script. Developers benefit from instant access to a secure, update and consistent
Node.js environment without having to manually install and configure multiple
components and libraries.",
        "isActive": true,
        "minPower": 0,
        "version": "18.20.2-0",
        "versionCode": "1",
        "productUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAK0",
        "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAK0#pdp-
usage",
        "platform": "LINUX_UNIX"
    },
    ...
}
]
}

```

- API-Einheiten finden Sie unter Befehlsreferenz [GetBlueprints](#).AWS CLI

get-bundles

Das folgende Codebeispiel zeigt die Verwendung `get-bundles`.

AWS CLI

Um die Bundles für neue Instances zu erhalten

Das folgende `get-bundles` Beispiel zeigt Details zu allen verfügbaren Bundles, die zum Erstellen neuer Instances in Amazon Lightsail verwendet werden können.

```
aws lightsail get-bundles
```

Ausgabe:

```
{
  "bundles": [
    {
      "price": 5.0,
      "cpuCount": 2,
      "diskSizeInGb": 20,
      "bundleId": "nano_3_0",
      "instanceType": "nano",
      "isActive": true,
      "name": "Nano",
      "power": 298,
      "ramSizeInGb": 0.5,
      "transferPerMonthInGb": 1024,
      "supportedPlatforms": [
        "LINUX_UNIX"
      ]
    },
    {
      "price": 7.0,
      "cpuCount": 2,
      "diskSizeInGb": 40,
      "bundleId": "micro_3_0",
      "instanceType": "micro",
      "isActive": true,
      "name": "Micro",
      "power": 500,
      "ramSizeInGb": 1.0,
      "transferPerMonthInGb": 2048,
      "supportedPlatforms": [
        "LINUX_UNIX"
      ]
    }
  ],
}
```



```
{
  "price": 12.0,
  "cpuCount": 2,
  "diskSizeInGb": 60,
  "bundleId": "small_3_0",
  "instanceType": "small",
  "isActive": true,
  "name": "Small",
  "power": 1000,
  "ramSizeInGb": 2.0,
  "transferPerMonthInGb": 3072,
  "supportedPlatforms": [
    "LINUX_UNIX"
  ],
  },
  ...
}
]
```

- API-Einheiten finden Sie [GetBundles](#) in AWS CLI der Befehlsreferenz.

get-cloud-formation-stack-records

Das folgende Codebeispiel zeigt die Verwendung `get-cloud-formation-stack-records`.

AWS CLI

Um die CloudFormation Stack-Datensätze und die zugehörigen Stapel abzurufen

Das folgende `get-cloud-formation-stack-records` Beispiel zeigt Details zu den CloudFormation Stack-Datensätzen und den zugehörigen Stacks, die verwendet wurden, um EC2 Amazon-Ressourcen aus exportierten Amazon Lightsail-Snapshots zu erstellen.

```
aws lightsail get-cloud-formation-stack-records
```

Ausgabe:

```
{
  "cloudFormationStackRecords": [
    {
```

```

    "name": "CloudFormationStackRecord-588a4243-
e2d1-490d-8200-3a7513ecebdf",
    "arn": "arn:aws:lightsail:us-
west-2:111122223333:CloudFormationStackRecord/28d646ab-27bc-48d9-a422-1EXAMPLE6d37",
    "createdAt": 1565301666.586,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "CloudFormationStackRecord",
    "state": "Succeeded",
    "sourceInfo": [
      {
        "resourceType": "ExportSnapshotRecord",
        "name": "ExportSnapshotRecord-
e02f23d7-0453-4aa9-9c95-91aa01a141dd",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:ExportSnapshotRecord/f12b8792-f3ea-4d6f-b547-2EXAMPLE8796"
      }
    ],
    "destinationInfo": {
      "id": "arn:aws:cloudformation:us-west-2:111122223333:stack/
Lightsail-Stack-588a4243-e2d1-490d-8200-3EXAMPLEebdf/063203b0-
ba28-11e9-838b-0EXAMPLE8b00",
      "service": "Aws::CloudFormation::Stack"
    }
  }
]
}

```

- API-Einheiten finden Sie unter [GetCloudFormationStackRecords](#) Befehlsreferenz. AWS CLI

get-disk-snapshot

Das folgende Codebeispiel zeigt die Verwendung `get-disk-snapshot`.

AWS CLI

Um Informationen über einen Festplatten-Snapshot abzurufen

Im folgenden `get-disk-snapshot` Beispiel werden Details zum Festplatten-Snapshot angezeigt `Disk-1-1566839161`.

```
aws lightsail get-disk-snapshot \  
--disk-snapshot-name Disk-1-1566839161
```

Ausgabe:

```
{  
  "diskSnapshot": {  
    "name": "Disk-1-1566839161",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/  
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",  
    "supportCode": "6EXAMPLE3362/snap-0EXAMPLE06100d09",  
    "createdAt": 1566839163.749,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "DiskSnapshot",  
    "tags": [],  
    "sizeInGb": 8,  
    "state": "completed",  
    "progress": "100%",  
    "fromDiskName": "Disk-1",  
    "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/  
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",  
    "isFromAutoSnapshot": false  
  }  
}
```

Weitere Informationen finden Sie im Titel des Handbuchs.

- API-Einheiten finden Sie [GetDiskSnapshot](#) in der AWS CLI Befehlsreferenz.

get-disk-snapshots

Das folgende Codebeispiel zeigt die Verwendung `get-disk-snapshots`.

AWS CLI

Um Informationen über alle Festplatten-Snapshots zu erhalten

Im folgenden `get-disk-snapshots` Beispiel werden Details zu allen Festplatten-Snapshots in der AWS konfigurierten Region angezeigt.

aws lightsail get-disk-snapshots

Ausgabe:

```
{
  "diskSnapshots": [
    {
      "name": "Disk-2-1571090588",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/32e889a9-38d4-4687-9f21-eEXAMPLE7839",
      "supportCode": "6EXAMPLE3362/snap-0EXAMPLE1ca192a4",
      "createdAt": 1571090591.226,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "DiskSnapshot",
      "tags": [],
      "sizeInGb": 8,
      "state": "completed",
      "progress": "100%",
      "fromDiskName": "Disk-2",
      "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
      "isFromAutoSnapshot": false
    },
    {
      "name": "Disk-1-1566839161",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",
      "supportCode": "6EXAMPLE3362/snap-0EXAMPLEe06100d09",
      "createdAt": 1566839163.749,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "DiskSnapshot",
      "tags": [],
      "sizeInGb": 8,
      "state": "completed",
      "progress": "100%",
      "fromDiskName": "Disk-1",
    }
  ]
}
```

```
        "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
        "isFromAutoSnapshot": false
    }
]
}
```

- API-Einheiten finden Sie unter [GetDiskSnapshots AWS CLI-Befehlsreferenz](#).

get-disk

Das folgende Codebeispiel zeigt die Verwendung `get-disk`.

AWS CLI

Um Informationen über eine Blockspeicherfestplatte zu erhalten

Im folgenden `get-disk` Beispiel werden Details zur Festplatte angezeigt `Disk-1`.

```
aws lightsail get-disk \
  --disk-name Disk-1
```

Ausgabe:

```
{
  "disk": {
    "name": "Disk-1",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
    "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
    "createdAt": 1566585439.587,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "Disk",
    "tags": [],
    "sizeInGb": 8,
    "isSystemDisk": false,
    "iops": 100,
    "path": "/dev/xvdf",
    "state": "in-use",
```

```
    "attachedTo": "WordPress_Multisite-1",
    "isAttached": true,
    "attachmentState": "attached"
  }
}
```

Weitere Informationen finden Sie im Titel des Handbuchs.

- API-Einheiten finden Sie [GetDisk](#) in der AWS CLI Befehlsreferenz.

get-disks

Das folgende Codebeispiel zeigt die Verwendung `get-disks`.

AWS CLI

Um Informationen über alle Blockspeicherplatten zu erhalten

Im folgenden `get-disks` Beispiel werden Details zu allen Festplatten in der konfigurierten AWS Region angezeigt.

```
aws lightsail get-disks
```

Ausgabe:

```
{
  "disks": [
    {
      "name": "Disk-2",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
      "supportCode": "6EXAMPLE3362/vol-0EXAMPLE929602087",
      "createdAt": 1571090461.634,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "Disk",
      "tags": [],
      "sizeInGb": 8,
      "isSystemDisk": false,
      "iops": 100,
    }
  ]
}
```

```

        "state": "available",
        "isAttached": false,
        "attachmentState": "detached"
    },
    {
        "name": "Disk-1",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
        "createdAt": 1566585439.587,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Disk",
        "tags": [],
        "sizeInGb": 8,
        "isSystemDisk": false,
        "iops": 100,
        "path": "/dev/xvdf",
        "state": "in-use",
        "attachedTo": "WordPress_Multisite-1",
        "isAttached": true,
        "attachmentState": "attached"
    }
]
}

```

- API-Einheiten finden Sie [GetDisks](#) in der AWS CLI Befehlsreferenz.

get-domain

Das folgende Codebeispiel zeigt die Verwendung `get-domain`.

AWS CLI

Um Informationen über eine Domain zu erhalten

Im folgenden `get-domain` Beispiel werden Details zur Domain `example.com` angezeigt.

Hinweis: Die domänenbezogenen API-Operationen von Lightsail sind nur in der Region verfügbar. `us-east-1` AWS. Wenn Ihr CLI-Profil für die Verwendung einer anderen Region konfiguriert ist, müssen Sie den Parameter `--region us-east-1` angeben, sonst schlägt der Befehl fehl.

```
aws lightsail get-domain \  
  --domain-name example.com \  
  --region us-east-1
```

Ausgabe:

```
{  
  "domain": {  
    "name": "example.com",  
    "arn":  
"arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",  
    "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",  
    "createdAt": 1570728588.6,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "global"  
    },  
    "resourceType": "Domain",  
    "tags": [],  
    "domainEntries": [  
      {  
        "id": "-1682899164",  
        "name": "example.com",  
        "target": "192.0.2.0",  
        "isAlias": false,  
        "type": "A"  
      },  
      {  
        "id": "1703104243",  
        "name": "example.com",  
        "target": "ns-137.awsdns-17.com",  
        "isAlias": false,  
        "type": "NS"  
      },  
      {  
        "id": "-1038331153",  
        "name": "example.com",  
        "target": "ns-1710.awsdns-21.co.uk",  
        "isAlias": false,  
        "type": "NS"  
      },  
      {  
        "id": "-2107289565",
```



```

        "name": "example.com",
        "target": "ns-692.awsdns-22.net",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "1582095705",
        "name": "example.com",
        "target": "ns-1436.awsdns-51.org",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "-1769796132",
        "name": "example.com",
        "target": "ns-1710.awsdns-21.co.uk. awsdns-hostmaster.amazon.com. 1
7200 900 1209600 86400",
        "isAlias": false,
        "type": "SOA"
    }
]
}
}

```

- Einzelheiten finden Sie in der Befehlsreferenz. API [GetDomain](#)AWS CLI

get-domains

Das folgende Codebeispiel zeigt die Verwendung `get-domains`.

AWS CLI

Um Informationen über alle Domains zu erhalten

Im folgenden `get-domains` Beispiel werden Details zu allen Domänen in der konfigurierten AWS Region angezeigt.

Hinweis: Die domänenbezogenen API Operationen von Lightsail sind nur in der Region verfügbar. `us-east-1` AWS Wenn Ihr CLI Profil für die Verwendung einer anderen Region konfiguriert ist, müssen Sie den `--region us-east-1` Parameter angeben, sonst schlägt der Befehl fehl.

```
aws lightsail get-domains \
```

```
--region us-east-1
```

Ausgabe:

```
{
  "domains": [
    {
      "name": "example.com",
      "arn":
"arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",
      "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",
      "createdAt": 1570728588.6,
      "location": {
        "availabilityZone": "all",
        "regionName": "global"
      },
      "resourceType": "Domain",
      "tags": [],
      "domainEntries": [
        {
          "id": "-1682899164",
          "name": "example.com",
          "target": "192.0.2.0",
          "isAlias": false,
          "type": "A"
        },
        {
          "id": "1703104243",
          "name": "example.com",
          "target": "ns-137.awsdns-17.com",
          "isAlias": false,
          "type": "NS"
        },
        {
          "id": "-1038331153",
          "name": "example.com",
          "target": "ns-4567.awsdns-21.co.uk",
          "isAlias": false,
          "type": "NS"
        },
        {
          "id": "-2107289565",
          "name": "example.com",
```

```

        "target": "ns-333.awsdns-22.net",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "1582095705",
        "name": "example.com",
        "target": "ns-1111.awsdns-51.org",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "-1769796132",
        "name": "example.com",
        "target": "ns-1234.awsdns-21.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
        "isAlias": false,
        "type": "SOA"
    },
    {
        "id": "1029454894",
        "name": "_dead6a124ede046a0319eb44a4eb3cbc.example.com",
        "target": "_be133b0a0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws",
        "isAlias": false,
        "type": "CNAME"
    }
]
},
{
    "name": "example.net",
    "arn": "arn:aws:lightsail:global:111122223333:Domain/9c9f0d70-
c92e-4753-86c2-6EXAMPLE029d",
    "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLE5TPKMV",
    "createdAt": 1556661071.384,
    "location": {
        "availabilityZone": "all",
        "regionName": "global"
    },
    "resourceType": "Domain",
    "tags": [],
    "domainEntries": [
        {
            "id": "-766320943",

```

```
    "name": "example.net",
    "target": "192.0.2.2",
    "isAlias": false,
    "type": "A"
  },
  {
    "id": "-453913825",
    "name": "example.net",
    "target": "ns-123.awsdns-10.net",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "1553601564",
    "name": "example.net",
    "target": "ns-4444.awsdns-47.co.uk",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "1653797661",
    "name": "example.net",
    "target": "ns-7890.awsdns-61.org",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "706414698",
    "name": "example.net",
    "target": "ns-123.awsdns-44.com",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "337271745",
    "name": "example.net",
    "target": "ns-4444.awsdns-47.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
    "isAlias": false,
    "type": "SOA"
  },
  {
    "id": "-1785431096",
    "name": "www.example.net",
```

```
        "target": "192.0.2.2",
        "isAlias": false,
        "type": "A"
    }
]
},
{
    "name": "example.org",
    "arn": "arn:aws:lightsail:global:111122223333:Domain/
f0f13ba3-3df0-4fdc-8ebb-1EXAMPLEf26e",
    "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEAF038",
    "createdAt": 1556661199.106,
    "location": {
        "availabilityZone": "all",
        "regionName": "global"
    },
    "resourceType": "Domain",
    "tags": [],
    "domainEntries": [
        {
            "id": "2065301345",
            "name": "example.org",
            "target": "192.0.2.4",
            "isAlias": false,
            "type": "A"
        },
        {
            "id": "-447198516",
            "name": "example.org",
            "target": "ns-123.awsdns-45.com",
            "isAlias": false,
            "type": "NS"
        },
        {
            "id": "136463022",
            "name": "example.org",
            "target": "ns-9999.awsdns-15.co.uk",
            "isAlias": false,
            "type": "NS"
        },
        {
            "id": "1395941679",
            "name": "example.org",
            "target": "ns-555.awsdns-01.net",
```

```

        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "872052569",
        "name": "example.org",
        "target": "ns-6543.awsdns-38.org",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "1001949377",
        "name": "example.org",
        "target": "ns-1234.awsdns-15.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
        "isAlias": false,
        "type": "SOA"
    },
    {
        "id": "1046191192",
        "name": "www.example.org",
        "target": "192.0.2.4",
        "isAlias": false,
        "type": "A"
    }
]
}
]
}
}

```

- API-Einheiten finden Sie [GetDomains](#) unter AWS CLI Befehlsreferenz.

get-export-snapshot-record

Das folgende Codebeispiel zeigt die Verwendung `get-export-snapshot-record`.

AWS CLI

Um die Aufzeichnungen von Schnappschüssen nach Amazon zu exportieren EC2

Das folgende `get-export-snapshot-record` Beispiel zeigt Details zur Amazon Lightsail-Instance oder zu Festplatten-Snapshots, die nach Amazon exportiert wurden. EC2

aws lightsail get-export-snapshot-records

Ausgabe:

```
{
  "exportSnapshotRecords": [
    {
      "name": "ExportSnapshotRecord-d2da10ce-0b3c-4ae1-ab3a-2EXAMPLEa586",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:ExportSnapshotRecord/076c7060-b0cc-4162-98f0-2EXAMPLEe28e",
      "createdAt": 1543534665.678,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "ExportSnapshotRecord",
      "state": "Succeeded",
      "sourceInfo": {
        "resourceType": "InstanceSnapshot",
        "createdAt": 1540339310.706,
        "name": "WordPress-512MB-0regon-1-1540339219",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/5446f534-ed60-4c17-b4a5-bEXAMPLEf8b7",
        "fromResourceName": "WordPress-512MB-0regon-1",
        "fromResourceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/4b8f1f24-e4d1-4cf3-88ff-cEXAMPLEa397",
        "instanceSnapshotInfo": {
          "fromBundleId": "nano_2_0",
          "fromBlueprintId": "wordpress_4_9_8",
          "fromDiskInfo": [
            {
              "path": "/dev/sda1",
              "sizeInGb": 20,
              "isSystemDisk": true
            }
          ]
        }
      },
      "destinationInfo": {
        "id": "ami-0EXAMPLEc0d65058e",
        "service": "Aws::EC2::Image"
      }
    },
  ],
}
```

```

    {
      "name": "ExportSnapshotRecord-1c94e884-40ff-4fe1-9302-0EXAMPLE14c2",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:ExportSnapshotRecord/
fb392ce8-6567-4013-9bfd-3EXAMPLE5b4c",
      "createdAt": 1543432110.2,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "ExportSnapshotRecord",
      "state": "Succeeded",
      "sourceInfo": {
        "resourceType": "InstanceSnapshot",
        "createdAt": 1540833603.545,
        "name": "LAMP_PHP_5-512MB-0regon-1-1540833565",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/82334399-b5f2-49ec-8382-0EXAMPLEe45f",
        "fromResourceName": "LAMP_PHP_5-512MB-0regon-1",
        "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/863b9f35-ab1e-4418-bdd2-1EXAMPLEbab2",
        "instanceSnapshotInfo": {
          "fromBundleId": "nano_2_0",
          "fromBlueprintId": "lamp_5_6_37_2",
          "fromDiskInfo": [
            {
              "path": "/dev/sda1",
              "sizeInGb": 20,
              "isSystemDisk": true
            }
          ]
        }
      },
      "destinationInfo": {
        "id": "ami-0EXAMPLE7c5ec84e2",
        "service": "Aws::EC2::Image"
      }
    }
  ]
}

```

- API-Einheiten finden Sie unter Befehlsreferenz [GetExportSnapshotRecord](#).AWS CLI

get-instance-access-details

Das folgende Codebeispiel zeigt die Verwendung `get-instance-access-details`.

AWS CLI

Um Host-Schlüsselinformationen für eine Instance abzurufen

Im folgenden `get-instance-access-details` Beispiel werden zum Beispiel Host-Schlüsselinformationen angezeigt `WordPress_Multisite-1`.

```
aws lightsail get-instance-access-details \
  --instance-name WordPress_Multisite-1
```

Ausgabe:

```
{
  "accessDetails": {
    "certKey": "ssh-rsa-cert-v01@openssh.com
AEXAMPLEEaC1yc2EtY2VydC12MDFAb3B1bnNzaC5jb20AAAAGNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
vEXAMPLE2eBWJyQvn7o1/
i0+s966h5sx8qUD791PB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjb1Lj26Fb37EKda9HVfC0u8pWbvky7Tyn9w29
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/
K1EXAMPLEAAAAAAAAABAAAALS00MzMzMdu4MzA40Dg1MTY2NjM40np6UW1ndHk4UE1RSG9Stit0TG5QSEE9PQAAAAAsAAA
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN
+tiXm7Y80gziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLuV7cnWPB7iU1HQg
+E3LUK1rV4ZFw9pj7X2dFdNKfMxwWgI1ISWKimEXAMPLEEhJrf1Rqc/
QH6TpWCvPfcx8uvwVqdwTfke/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/
KwmM7pCwJbSLDM07n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEEobqfdtyYJHHe11EyyEJs1fWNU3D5JIG1gzcPAV
+Z1bQyUCZXf0os1Sa+HE85f0/
FRq9SVSBSHrmb0fr1PhgMzgSmqLeyhlbr6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPWvRgcr0qI2EL0tAajnpQQ8UZo
Aqter0xN5PhFL0J490WTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8c111327EXAMPLE/
e89GC89KcmKCxfGQniDAUgF8UqofIbq3Z0UgiAAYCVXc1I4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/
Oz0cTR+4sAAAIPIAAAAB3NzaC1yc2EAAAIAQnG/
L0DqiSnLrWhEox4aHqMgd0m0oLLAYx60QH9F0TM9EXAMPLE961rzSCMon7ZgsWnNl00wZQgDG
+rtJ4N0B7H0Vwns4ynUFbzNQ3qFGGeE31KwX1L41vV1iSy7sDk8aI0LmrKJi1LE1Qc118uboR1woX0YEXAMPLEEaUCeX
+10+WEXAMPLEg6Y4U4ZvE2B3xyRdpvysb5TGFNTk5qPs1acnVkoL0GsZZXMPLGJnG40BpQLLtpj9sNMxAgZPCAUjhkqk
+nx0904NUZ2pTwbVSUaV1gm6pug9xbwN01Im21t34JeLlKTqxcJ6zzS8W0c0KKpAm5c4hWkseMbyutS2jav/4hiS
+BhrYgptzfwe5qRXEXAMPLEEHZQr3YfGzYoBJ/
1LK3NHhx0ihhsfAYwMei0BFZT1F/7CT3IH4iitEkIgodio6/
Mw6UDqMPozyQCK11EA6LFhYC0ZG9drWcoRa741M4kY9TP028Za8gDMh1WpkXLq9Gixon50HP8aM/
SEXAMPLEr2+fnkw+1Bto05L6+VKoPlXaGqZ/fBYEXAMPLEAMQHjnLM1JYNvtEEPhp+TNzXHzuixWf/
Ht04m0AVpXrzIDXaS102tXY=",
```

```

    "ipAddress": "192.0.2.0",
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nEXAMPLEBAAKCAQEA+AD3qeU2toBy505v7wnRLVo/tngVickL5+6Jf4tPrPeuoebM
\nfKlA+/ZTwe6uVBeneVWRhbcra8pH0CZ44sKnuxFeWoM7425S49uhW9+xCnWvR1Xw
\njrvKvm75Mu08p/cNvfWugrBuaPB65DspgxNn0fZWMVxpIpSq0SPWmSwQHV597d6C
\nrEXAMPLEe08hJmqz2KFQ09X7fB2lBruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaF10GNGQ8qy1HGUISccquZ9NDIL49n4mXbfsTH
\n0EXAMPLE12ZqsfLiYnSaUYCwjE74qH8ECVPytQIDAQABAoIBAHeZV9Z58JHAjifz
\nCEXAMPLEEqC3do0VDgXS1kKI92qNo4z2VcUEho878paCuVVXVHCcGgSnGeyIh2tN
\nMEXAMPLESohR427BhH3YLA+3Z5SIVnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn
\nIEXAMPLEM93oF9eSZCjclKB/jGHsfb0eCDMP8BshHE2beuqzVMoK1Dx0nvoP3+Fp
\nAEXAMPLESsq6pDpCo9YVUX8g1u3Ro9cPl2LXHDy+oVEY5KhbZQJ7VU1I72W0vppWW
\n0EXAMPLEkgY1q7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoNONQ9ZPaWHyJvymeud
\noQTNuz0CgYEA/LFWNTEZrzdzdR1kJmyNRmAermU0B6utyNENChAlHGSHkB+1lVSh
\nbEXAMPLEQo9ooUew5Ux03YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob
\n8l7CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/E0rB0we0RKYDSzWScCgYEA+z/r
\niob+nJZq0Ybn0SuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJd0NKDdkui0d52aAL
\nVgn7nLo88rVWKhJwVc4tu/rNgZLcR3bP4+kL6zand0KQnMLy0zNA2Ys26aa5udH1\nqwl0WTt9WEm/
h10ndC1kn0MectrvsG17b38y5sMCgYEA54NiRGGz8oCPW6GN/FZA
\nKEXAMPLE5tw34GEH3Uxlcn3CeJDaQmcz0ATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkc8Z71L/
agZEAaVCEog9FqfSqwB
+XTfoKh8qur74X1yCu9p6gof1q6k9\nEXAMPLEechJcNN0g4ETIfMkCgYBdV0RRhE4mqvWp0dzA7v66FdEz2YSkjAXKk
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\nngEXAMPLEBogjw4MqHKL1bPKMHyQU8/
q24PaYgzHPzy13w1H6pTYf1Xq1HdE2D6Vv\nnyEXAMPLEgQC3i/
kVvhky/2XRwRVLC7J02Bg3QGTx38hpmDa5IuofKANjA+Wa3/zy\nnbEXAMPLE6ytQgD9GN/YtBq+uh0
+2ZkvXPL+CWRi0ZRxpPwYDBBFU9Cw0AuWWG1L8\nnwEXAMPLExMlcysRgcWB9RNgf3AuOpFd2i6XT/
riNsvvkpmJ+VooU8g=\\n-----END RSA PRIVATE KEY-----\\n",
    "protocol": "ssh",
    "instanceName": "WordPress_Multisite-1",
    "username": "bitnami",
    "hostKeys": [
        {
            "algorithm": "ssh-rsa",
            "publicKey":
                "AEXAMPLEaC1yc2EAAAAADAQABAAABAQCoer9ieZTjQ3pXCHczuAYZFj1F7t
+uBkXuqeGMREx78pCvmS+DiEXAMPLEEuJ1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdfHFzy3guP
+BKclWqtXJEXAMPLEsBGqZZlrIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0X1Ms7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHpp440g36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
0t6y9HwvykEXAMPLEAfbKjBR42+u6+0Slkr4d339q2U1sTDytJhhs8HUel1wTfGRfp",
            "witnessedAt": 1570744377.699,
            "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
            "fingerprintSHA256": "SHA256:IEEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
        }
    ]
}

```

```

    },
    {
      "algorithm": "ssh-ed25519",
      "publicKey":
"AEXAMPLEaC11ZDI1NTE5AAAAIC1gwGPDfGa0NxEXAMPLEJX3UNap781QxHQmn8nz1rUv",
      "witnessedAt": 1570744377.697,
      "fingerprintSHA1": "SHA1:VEXAMPLE5ReqSmTgv03sSUw9toU",
      "fingerprintSHA256": "SHA256:0EXAMPLEdE6tI95k3TJpG
+qhJbAoknB0yz9nAEaDt3A"
    },
    {
      "algorithm": "ecdsa-sha2-nistp256",
      "publicKey":
"AEXAMPLEZHNhLXNoYTIItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABEXAMPLE9B4mZy8YSsZW7cixCDq5yHSAAxjJkDo5
+EnK1DCsYtUkxxEXAMPLE6V0WL2z63RTKa2AUPgd8irjxWI=",
      "witnessedAt": 1570744377.707,
      "fingerprintSHA1": "SHA1:UEXAMPLE0YCFXsCf2G6tDg+7YG0",
      "fingerprintSHA256": "SHA256:wEXAMPLEQ9a/
iEXAMPLEhRufm6U9vFU4cpkMPHnBsNA"
    }
  ]
}
}

```

- API-Einheiten finden Sie [GetInstanceAccessDetails](#) in der AWS CLI Befehlsreferenz.

get-instance-metric-data

Das folgende Codebeispiel zeigt die Verwendung `get-instance-metric-data`.

AWS CLI

Um metrische Daten für eine Instanz abzurufen

Das folgende `get-instance-metric-data` Beispiel gibt zum Beispiel den durchschnittlichen Prozentsatz CPU-Utilization aller 7200 Sekunden (2 Stunden) zwischen 1571342400 und 1571428800 zurück `MEAN-1`.

Wir empfehlen, einen Unix-Zeitkonverter zu verwenden, um die Start- und Endzeiten zu ermitteln.

```
aws lightsail get-instance-metric-data \
  --instance-name MEAN-1 \
```

```
--metric-name CPUUtilization \  
--period 7200 \  
--start-time 1571342400 \  
--end-time 1571428800 \  
--unit Percent \  
--statistics Average
```

Ausgabe:

```
{  
  "metricName": "CPUUtilization",  
  "metricData": [  
    {  
      "average": 0.26113718770120725,  
      "timestamp": 1571342400.0,  
      "unit": "Percent"  
    },  
    {  
      "average": 0.26861268928111953,  
      "timestamp": 1571392800.0,  
      "unit": "Percent"  
    },  
    {  
      "average": 0.28187475104748777,  
      "timestamp": 1571378400.0,  
      "unit": "Percent"  
    },  
    {  
      "average": 0.2651936960458352,  
      "timestamp": 1571421600.0,  
      "unit": "Percent"  
    },  
    {  
      "average": 0.2561856213712188,  
      "timestamp": 1571371200.0,  
      "unit": "Percent"  
    },  
    {  
      "average": 0.3021383254607764,  
      "timestamp": 1571356800.0,  
      "unit": "Percent"  
    }  
  ]  
}
```

```
    "average": 0.2618381649223539,
    "timestamp": 1571407200.0,
    "unit": "Percent"
  },
  {
    "average": 0.26331929394825787,
    "timestamp": 1571400000.0,
    "unit": "Percent"
  },
  {
    "average": 0.2576348407007818,
    "timestamp": 1571385600.0,
    "unit": "Percent"
  },
  {
    "average": 0.2513008454658378,
    "timestamp": 1571364000.0,
    "unit": "Percent"
  },
  {
    "average": 0.26329974562758346,
    "timestamp": 1571414400.0,
    "unit": "Percent"
  },
  {
    "average": 0.2667092536656445,
    "timestamp": 1571349600.0,
    "unit": "Percent"
  }
]
}
```

- API-Einheiten finden Sie [GetInstanceMetricData](#) in der AWS CLI Befehlsreferenz.

get-instance-port-states

Das folgende Codebeispiel zeigt die Verwendung von `get-instance-port-states`.

AWS CLI

Um Firewall-Informationen für eine Instanz abzurufen

Das folgende `get-instance-port-states` Beispiel gibt die zum Beispiel konfigurierten Firewall-Ports zurückMEAN-1.

```
aws lightsail get-instance-port-states \  
  --instance-name MEAN-1
```

Ausgabe:

```
{  
  "portStates": [  
    {  
      "fromPort": 80,  
      "toPort": 80,  
      "protocol": "tcp",  
      "state": "open"  
    },  
    {  
      "fromPort": 22,  
      "toPort": 22,  
      "protocol": "tcp",  
      "state": "open"  
    },  
    {  
      "fromPort": 443,  
      "toPort": 443,  
      "protocol": "tcp",  
      "state": "open"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetInstancePortStates](#) in der AWS CLI Befehlsreferenz.

get-instance-snapshot

Das folgende Codebeispiel zeigt die Verwendung `get-instance-snapshot`.

AWS CLI

Um Informationen zu einem bestimmten Instanz-Snapshot abzurufen

Im folgenden `get-instance-snapshot` Beispiel werden Details zum angegebenen Instanz-Snapshot angezeigt.

```
aws lightsail get-instance-snapshot \  
--instance-snapshot-name MEAN-1-1571419854
```

Ausgabe:

```
{  
  "instanceSnapshot": {  
    "name": "MEAN-1-1571419854",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/  
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",  
    "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",  
    "createdAt": 1571419891.927,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "InstanceSnapshot",  
    "tags": [],  
    "state": "available",  
    "fromAttachedDisks": [],  
    "fromInstanceName": "MEAN-1",  
    "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/  
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",  
    "fromBlueprintId": "mean",  
    "fromBundleId": "medium_3_0",  
    "isFromAutoSnapshot": false,  
    "sizeInGb": 80  
  }  
}
```

- API-Einheiten finden Sie [GetInstanceSnapshot](#) unter AWS CLI Befehlsreferenz.

get-instance-snapshots

Das folgende Codebeispiel zeigt die Verwendung `get-instance-snapshots`.

AWS CLI

Um Informationen zu all Ihren Instance-Snapshots zu erhalten

Im folgenden `get-instance-snapshots` Beispiel werden Details zu allen Instanz-Snapshots in der AWS konfigurierten Region angezeigt.

```
aws lightsail get-instance-snapshots
```

Ausgabe:

```
{
  "instanceSnapshots": [
    {
      "name": "MEAN-1-1571421498",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/a20e6ebe-b0ee-4ae4-a750-3EXAMPLEcb0c",
      "supportCode": "6EXAMPLE3362/ami-0EXAMPLEe33cabfa1",
      "createdAt": 1571421527.755,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "InstanceSnapshot",
      "tags": [
        {
          "key": "no_delete"
        }
      ],
      "state": "available",
      "fromAttachedDisks": [],
      "fromInstanceName": "MEAN-1",
      "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/1761aa0a-6038-4f25-8b94-2EXAMPLE19fd",
      "fromBlueprintId": "wordpress",
      "fromBundleId": "micro_3_0",
      "isFromAutoSnapshot": false,
      "sizeInGb": 40
    },
    {
      "name": "MEAN-1-1571419854",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
      "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
      "createdAt": 1571419891.927,
      "location": {
        "availabilityZone": "all",

```



```

        "regionName": "us-west-2"
      },
      "resourceType": "InstanceSnapshot",
      "tags": [],
      "state": "available",
      "fromAttachedDisks": [],
      "fromInstanceName": "MEAN-1",
      "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
      "fromBlueprintId": "mean",
      "fromBundleId": "medium_3_0",
      "isFromAutoSnapshot": false,
      "sizeInGb": 80
    }
  ]
}

```

- API-Einheiten finden Sie unter [GetInstanceSnapshots AWS CLI-Befehlsreferenz](#).

get-instance-state

Das folgende Codebeispiel zeigt die Verwendung `get-instance-state`.

AWS CLI

Um Informationen über den Status einer Instanz abzurufen

Das folgende `get-instance-state` Beispiel gibt den Status der angegebenen Instanz zurück.

```
aws lightsail get-instance-state \
  --instance-name MEAN-1
```

Ausgabe:

```
{
  "state": {
    "code": 16,
    "name": "running"
  }
}
```

- API-Einheiten finden Sie [GetInstanceState](#) unter AWS CLI Befehlsreferenz.

get-instance

Das folgende Codebeispiel zeigt die Verwendung `get-instance`.

AWS CLI

Um Informationen über eine Instanz zu erhalten

Im folgenden `get-instance` Beispiel werden Details zur Instanz angezeigt `MEAN-1`.

```
aws lightsail get-instance \  
  --instance-name MEAN-1
```

Ausgabe:

```
{  
  "instance": {  
    "name": "MEAN-1",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-  
a68b-44c5-8dbc-EXAMPLE4bada",  
    "supportCode": "6EXAMPLE3362/i-05EXAMPLE407c97d3",  
    "createdAt": 1570635023.124,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "Instance",  
    "tags": [],  
    "blueprintId": "mean",  
    "blueprintName": "MEAN",  
    "bundleId": "medium_3_0",  
    "isStaticIp": false,  
    "privateIpAddress": "192.0.2.0",  
    "publicIpAddress": "192.0.2.0",  
    "hardware": {  
      "cpuCount": 2,  
      "disks": [  
        {  
          "createdAt": 1570635023.124,  
          "sizeInGb": 80,  
          "isSystemDisk": true,  
          "iops": 240,  
          "path": "/dev/xvda",  
          "attachedTo": "MEAN-1",
```

```
        "attachmentState": "attached"
      }
    ],
    "ramSizeInGb": 4.0
  },
  "networking": {
    "monthlyTransfer": {
      "gbPerMonthAllocated": 4096
    },
    "ports": [
      {
        "fromPort": 80,
        "toPort": 80,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
      },
      {
        "fromPort": 22,
        "toPort": 22,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
      },
      {
        "fromPort": 443,
        "toPort": 443,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
      }
    ]
  },
  "state": {
    "code": 16,
    "name": "running"
  },
  "username": "bitnami",
```

```
    "sshKeyName": "MyKey"
  }
}
```

- API-Einheiten finden Sie [GetInstance](#) unter AWS CLI Befehlsreferenz.

get-instances

Das folgende Codebeispiel zeigt die Verwendung `get-instances`.

AWS CLI

Um Informationen über alle Instanzen zu erhalten

Im folgenden `get-instances` Beispiel werden Details zu allen Instanzen in der konfigurierten AWS Region angezeigt.

```
aws lightsail get-instances
```

Ausgabe:

```
{
  "instances": [
    {
      "name": "Windows_Server_2022-1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/0f44fbb9-8f55-4e47-a25e-EXAMPLE04763",
      "supportCode": "62EXAMPLE362/i-0bEXAMPLE71a686b9",
      "createdAt": 1571332358.665,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "Instance",
      "tags": [],
      "blueprintId": "windows_server_2022",
      "blueprintName": "Windows Server 2022",
      "bundleId": "large_win_3_0",
      "isStaticIp": false,
      "privateIpAddress": "192.0.2.0",
      "publicIpAddress": "192.0.2.0",
      "hardware": {
```

```
    "cpuCount": 1,
    "disks": [
      {
        "createdAt": 1571332358.665,
        "sizeInGb": 160,
        "isSystemDisk": true,
        "iops": 180,
        "path": "/dev/sda1",
        "attachedTo": "Windows_Server_2022-1",
        "attachmentState": "attached"
      },
      {
        "name": "my-disk-for-windows-server",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/4123a81c-484c-49ea-afea-5EXAMPLEda87",
        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEb2b99ca3d",
        "createdAt": 1571355063.494,
        "location": {
          "availabilityZone": "us-west-2a",
          "regionName": "us-west-2"
        },
        "resourceType": "Disk",
        "tags": [],
        "sizeInGb": 128,
        "isSystemDisk": false,
        "iops": 384,
        "path": "/dev/xvdf",
        "state": "in-use",
        "attachedTo": "Windows_Server_2022-1",
        "isAttached": true,
        "attachmentState": "attached"
      }
    ],
    "ramSizeInGb": 8.0
  },
  "networking": {
    "monthlyTransfer": {
      "gbPerMonthAllocated": 3072
    },
    "ports": [
      {
        "fromPort": 80,
        "toPort": 80,
        "protocol": "tcp",
```

```
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
    },
    {
        "fromPort": 22,
        "toPort": 22,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
    },
    {
        "fromPort": 3389,
        "toPort": 3389,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
    }
]
},
"state": {
    "code": 16,
    "name": "running"
},
"username": "Administrator",
"sshKeyName": "LightsailDefaultKeyPair"
},
{
    "name": "MEAN-1",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-8EXAMPLEbada",
    "supportCode": "6EXAMPLE3362/i-0EXAMPLEa407c97d3",
    "createdAt": 1570635023.124,
    "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "resourceType": "Instance",
    "tags": [],
```

```
"blueprintId": "mean",
"blueprintName": "MEAN",
"bundleId": "medium_3_0",
"isStaticIp": false,
"privateIpAddress": "192.0.2.0",
"publicIpAddress": "192.0.2.0",
"hardware": {
  "cpuCount": 2,
  "disks": [
    {
      "name": "Disk-1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
      "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
      "createdAt": 1566585439.587,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "Disk",
      "tags": [
        {
          "key": "test"
        }
      ],
      "sizeInGb": 8,
      "isSystemDisk": false,
      "iops": 240,
      "path": "/dev/xvdf",
      "state": "in-use",
      "attachedTo": "MEAN-1",
      "isAttached": true,
      "attachmentState": "attached"
    },
    {
      "createdAt": 1570635023.124,
      "sizeInGb": 80,
      "isSystemDisk": true,
      "iops": 240,
      "path": "/dev/sda1",
      "attachedTo": "MEAN-1",
      "attachmentState": "attached"
    }
  ],
}
```

```
        "ramSizeInGb": 4.0
    },
    "networking": {
        "monthlyTransfer": {
            "gbPerMonthAllocated": 4096
        },
        "ports": [
            {
                "fromPort": 80,
                "toPort": 80,
                "protocol": "tcp",
                "accessFrom": "Anywhere (0.0.0.0/0)",
                "accessType": "public",
                "commonName": "",
                "accessDirection": "inbound"
            },
            {
                "fromPort": 22,
                "toPort": 22,
                "protocol": "tcp",
                "accessFrom": "Anywhere (0.0.0.0/0)",
                "accessType": "public",
                "commonName": "",
                "accessDirection": "inbound"
            },
            {
                "fromPort": 443,
                "toPort": 443,
                "protocol": "tcp",
                "accessFrom": "Anywhere (0.0.0.0/0)",
                "accessType": "public",
                "commonName": "",
                "accessDirection": "inbound"
            }
        ]
    },
    "state": {
        "code": 16,
        "name": "running"
    },
    "username": "bitnami",
    "sshKeyName": "MyTestKey"
}
]
```



```
}
```

- API-Einheiten finden Sie [GetInstances](#) unter AWS CLI Befehlsreferenz.

get-key-pair

Das folgende Codebeispiel zeigt die Verwendung `get-key-pair`.

AWS CLI

Um Informationen über ein key pair zu erhalten

Im folgenden `get-key-pair` Beispiel werden Details zum angegebenen key pair angezeigt.

```
aws lightsail get-key-pair \  
  --key-pair-name MyKey1
```

Ausgabe:

```
{  
  "keyPair": {  
    "name": "MyKey1",  
    "arn": "arn:aws:lightsail:us-  
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",  
    "supportCode": "6EXAMPLE3362/MyKey1",  
    "createdAt": 1571255026.975,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "KeyPair",  
    "tags": [],  
    "fingerprint": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"  
  }  
}
```

- API-Einheiten finden Sie [GetKeyPair](#) unter AWS CLI Befehlsreferenz.

get-key-pairs

Das folgende Codebeispiel zeigt die Verwendung `get-key-pairs`.

AWS CLI

Um Informationen über alle Schlüsselpaare zu erhalten

Im folgenden `get-key-pairs` Beispiel werden Details zu allen Schlüsselpaaren in der konfigurierten AWS Region angezeigt.

```
aws lightsail get-key-pairs
```

Ausgabe:

```
{
  "keyPairs": [
    {
      "name": "MyKey1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",
      "supportCode": "6EXAMPLE3362/MyKey1",
      "createdAt": 1571255026.975,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "KeyPair",
      "tags": [],
      "fingerprint":
      "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"
    }
  ]
}
```

- API-Einheiten finden Sie [GetKeyPairs](#) unter AWS CLI Befehlsreferenz.

`get-load-balancer-tls-certificates`

Das folgende Codebeispiel zeigt die Verwendung `get-load-balancer-tls-certificates`.

AWS CLI

Um Informationen über TLS Zertifikate für einen Load Balancer zu erhalten

Im folgenden `get-load-balancer-tls-certificates` Beispiel werden Details zu den TLS Zertifikaten für den angegebenen Load Balancer angezeigt.

```
aws lightsail get-load-balancer-tls-certificates \
  --load-balancer-name LoadBalancer-1
```

Ausgabe:

```
{
  "tlsCertificates": [
    {
      "name": "example-com",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:LoadBalancerTlsCertificate/d7bf4643-6a02-4cd4-b3c4-fEXAMPLE9b4d",
      "supportCode": "6EXAMPLE3362/arn:aws:acm:us-west-2:333322221111:certificate/9af8e32c-a54e-4a67-8c63-cEXAMPLEb314",
      "createdAt": 1571678025.3,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "LoadBalancerTlsCertificate",
      "loadBalancerName": "LoadBalancer-1",
      "isAttached": false,
      "status": "ISSUED",
      "domainName": "example.com",
      "domainValidationRecords": [
        {
          "name": "_dEXAMPLE4ede046a0319eb44a4eb3cbc.example.com.",
          "type": "CNAME",
          "value": "_bEXAMPLE0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-validations.aws.",
          "validationStatus": "SUCCESS",
          "domainName": "example.com"
        }
      ],
      "issuedAt": 1571678070.0,
      "issuer": "Amazon",
      "keyAlgorithm": "RSA-2048",
      "notAfter": 1605960000.0,
      "notBefore": 1571616000.0,
      "serial": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff",
    }
  ]
}
```

```

        "signatureAlgorithm": "SHA256WITHRSA",
        "subject": "CN=example.com",
        "subjectAlternativeNames": [
            "example.com"
        ]
    }
]
}

```

- API-Einheiten finden Sie unter [GetLoadBalancerTlsCertificates AWS CLI](#) Befehlsreferenz.

get-load-balancer

Das folgende Codebeispiel zeigt die Verwendung von `get-load-balancer`.

AWS CLI

Um Informationen über einen Load Balancer zu erhalten

Im folgenden `get-load-balancer` Beispiel werden Details zum angegebenen Load Balancer angezeigt.

```

aws lightsail get-load-balancer \
  --load-balancer-name LoadBalancer-1

```

Ausgabe:

```

{
  "loadBalancer": {
    "name": "LoadBalancer-1",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
    "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-west-2:333322221111:loadbalancer/app/bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
    "createdAt": 1571677906.723,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "LoadBalancer",
  }
}

```

```
    "tags": [],
    "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-
west-2.elb.amazonaws.com",
    "state": "active",
    "protocol": "HTTP",
    "publicPorts": [
      80
    ],
    "healthCheckPath": "/",
    "instancePort": 80,
    "instanceHealthSummary": [
      {
        "instanceName": "MEAN-3",
        "instanceHealth": "healthy"
      },
      {
        "instanceName": "MEAN-1",
        "instanceHealth": "healthy"
      },
      {
        "instanceName": "MEAN-2",
        "instanceHealth": "healthy"
      }
    ],
    "tlsCertificateSummaries": [
      {
        "name": "example-com",
        "isAttached": false
      }
    ],
    "configurationOptions": {
      "SessionStickinessEnabled": "false",
      "SessionStickiness_LB_CookieDurationSeconds": "86400"
    }
  }
}
```

- API-Einheiten finden Sie unter [GetLoadBalancer AWS CLI](#) Befehlsreferenz.

get-load-balancers

Das folgende Codebeispiel zeigt die Verwendung `get-load-balancers`.

AWS CLI

Um Informationen über alle Load Balancer zu erhalten

Im folgenden `get-load-balancers` Beispiel werden Details zu allen Load Balancern in der AWS konfigurierten Region angezeigt.

```
aws lightsail get-load-balancers
```

Ausgabe:

```
{
  "loadBalancers": [
    {
      "name": "LoadBalancer-1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
      "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-west-2:333322221111:loadbalancer/app/bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
      "createdAt": 1571677906.723,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "LoadBalancer",
      "tags": [],
      "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-west-2.elb.amazonaws.com",
      "state": "active",
      "protocol": "HTTP",
      "publicPorts": [
        80
      ],
      "healthCheckPath": "/",
      "instancePort": 80,
      "instanceHealthSummary": [
        {
          "instanceName": "MEAN-3",
          "instanceHealth": "healthy"
        },
        {
          "instanceName": "MEAN-1",
```

```

        "instanceHealth": "healthy"
      },
      {
        "instanceName": "MEAN-2",
        "instanceHealth": "healthy"
      }
    ],
    "tlsCertificateSummaries": [
      {
        "name": "example-com",
        "isAttached": false
      }
    ],
    "configurationOptions": {
      "SessionStickinessEnabled": "false",
      "SessionStickiness_LB_CookieDurationSeconds": "86400"
    }
  }
]
}

```

- API-Einheiten finden Sie unter [GetLoadBalancers AWS CLI-Befehlsreferenz](#).

get-operation

Das folgende Codebeispiel zeigt die Verwendung get-operation.

AWS CLI

Um Informationen über eine einzelne Operation zu erhalten

Im folgenden get-operation Beispiel werden Details zur angegebenen Operation angezeigt.

```
aws lightsail get-operation \
  --operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

Ausgabe:

```
{
  "operation": {
    "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
    "resourceName": "Instance-1",

```

```
    "resourceType": "Instance",
    "createdAt": 1571679872.404,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "CreateInstance",
    "status": "Succeeded",
    "statusChangedAt": 1571679890.304
  }
}
```

- API-Einheiten finden Sie [GetOperation](#) unter AWS CLI Befehlsreferenz.

get-operations-for-resource

Das folgende Codebeispiel zeigt die Verwendung `get-operations-for-resource`.

AWS CLI

Um alle Operationen für eine Ressource abzurufen

Im folgenden `get-operations-for-resource` Beispiel werden Details zu allen Vorgängen für die angegebene Ressource angezeigt.

```
aws lightsail get-operations-for-resource \
  --resource-name LoadBalancer-1
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",
      "createdAt": 1571678786.071,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      }
    }
  ]
}
```



```
    },
    "isTerminal": true,
    "operationDetails": "MEAN-1",
    "operationType": "DetachInstancesFromLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571679087.57
  },
  {
    "id": "2d742a18-0e7f-48c8-9705-3EXAMPLEf98a",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571678782.784,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571678798.465
  },
  {
    "id": "6c700fcc-4246-40ab-952b-1EXAMPLEd2",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571678775.297,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-3",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571678842.806
  },
  ...
}
]
```

- API-Einheiten finden Sie [GetOperationsForResource](#) unter AWS CLI Befehlsreferenz.

get-operations

Das folgende Codebeispiel zeigt die Verwendung `get-operations`.

AWS CLI

Um Informationen über alle Operationen zu erhalten

Im folgenden `get-operations` Beispiel werden Details zu allen Vorgängen in der konfigurierten AWS Region angezeigt.

```
aws lightsail get-operations
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
      "resourceName": "Instance-1",
      "resourceType": "Instance",
      "createdAt": 1571679872.404,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "CreateInstance",
      "status": "Succeeded",
      "statusChangedAt": 1571679890.304
    },
    {
      "id": "701a3339-930e-4914-a9f9-7EXAMPLE68d7",
      "resourceName": "WordPress-1",
      "resourceType": "Instance",
      "createdAt": 1571678786.072,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "LoadBalancer-1",
      "operationType": "DetachInstancesFromLoadBalancer",

```

```
    "status": "Succeeded",
    "statusChangedAt": 1571679086.399
  },
  {
    "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571678786.071,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "WordPress-1",
    "operationType": "DetachInstancesFromLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571679087.57
  },
  ...
}
]
```

- API-Einheiten finden Sie [GetOperations](#) unter AWS CLI Befehlsreferenz.

get-regions

Das folgende Codebeispiel zeigt die Verwendung `get-regions`.

AWS CLI

Um alle AWS Regionen für Amazon Lightsail zu erhalten

Im folgenden `get-regions` Beispiel werden Details zu allen AWS Regionen für Amazon Lightsail angezeigt.

```
aws lightsail get-regions
```

Ausgabe:

```
{
```

```

    "regions": [
      {
        "continentCode": "NA",
        "description": "This region is recommended to serve users in the eastern
United States",
        "displayName": "Virginia",
        "name": "us-east-1",
        "availabilityZones": [],
        "relationalDatabaseAvailabilityZones": []
      },
      {
        "continentCode": "NA",
        "description": "This region is recommended to serve users in the eastern
United States",
        "displayName": "Ohio",
        "name": "us-east-2",
        "availabilityZones": [],
        "relationalDatabaseAvailabilityZones": []
      },
      {
        "continentCode": "NA",
        "description": "This region is recommended to serve users in the
northwestern United States, Alaska, and western Canada",
        "displayName": "Oregon",
        "name": "us-west-2",
        "availabilityZones": [],
        "relationalDatabaseAvailabilityZones": []
      },
      ...
    ]
  }
}

```

- API-Einheiten finden Sie unter [GetRegions AWS CLI](#) Befehlsreferenz.

get-relational-database-blueprints

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-blueprints`.

AWS CLI

Um die Blueprints für neue relationale Datenbanken zu erhalten

Das folgende `get-relational-database-blueprints` Beispiel zeigt Details zu allen verfügbaren relationalen Datenbank-Blueprints, die verwendet werden können, um neue relationale Datenbanken in Amazon Lightsail zu erstellen.

```
aws lightsail get-relational-database-blueprints
```

Ausgabe:

```
{
  "blueprints": [
    {
      "blueprintId": "mysql_5_6",
      "engine": "mysql",
      "engineVersion": "5.6.44",
      "engineDescription": "MySQL Community Edition",
      "engineVersionDescription": "MySQL 5.6.44",
      "isEngineDefault": false
    },
    {
      "blueprintId": "mysql_5_7",
      "engine": "mysql",
      "engineVersion": "5.7.26",
      "engineDescription": "MySQL Community Edition",
      "engineVersionDescription": "MySQL 5.7.26",
      "isEngineDefault": true
    },
    {
      "blueprintId": "mysql_8_0",
      "engine": "mysql",
      "engineVersion": "8.0.16",
      "engineDescription": "MySQL Community Edition",
      "engineVersionDescription": "MySQL 8.0.16",
      "isEngineDefault": false
    },
    {
      "blueprintId": "postgres_9_6",
      "engine": "postgres",
      "engineVersion": "9.6.15",
      "engineDescription": "PostgreSQL",
      "engineVersionDescription": "PostgreSQL 9.6.15-R1",
      "isEngineDefault": false
    },
    {
```

```

        "blueprintId": "postgres_10",
        "engine": "postgres",
        "engineVersion": "10.10",
        "engineDescription": "PostgreSQL",
        "engineVersionDescription": "PostgreSQL 10.10-R1",
        "isEngineDefault": false
    },
    {
        "blueprintId": "postgres_11",
        "engine": "postgres",
        "engineVersion": "11.5",
        "engineDescription": "PostgreSQL",
        "engineVersionDescription": "PostgreSQL 11.5-R1",
        "isEngineDefault": true
    }
]
}

```

- API-Einheiten finden Sie unter Befehlsreferenz. [GetRelationalDatabaseBlueprints](#) AWS CLI

get-relational-database-bundles

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-bundles`.

AWS CLI

Um die Bundles für neue relationale Datenbanken zu erhalten

Das folgende `get-relational-database-bundles` Beispiel zeigt Details zu allen verfügbaren relationalen Datenbankpaketen, die zur Erstellung neuer relationaler Datenbanken in Amazon Lightsail verwendet werden können. Beachten Sie, dass die Antwort keine inaktiven Bundles enthält, da das `--include-inactive` Flag nicht im Befehl angegeben ist. Sie können inaktive Bundles nicht verwenden, um neue relationale Datenbanken zu erstellen.

```
aws lightsail get-relational-database-bundles
```

Ausgabe:

```

{
  "bundles": [
    {
      "bundleId": "micro_2_0",

```

```
    "name": "Micro",
    "price": 15.0,
    "ramSizeInGb": 1.0,
    "diskSizeInGb": 40,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "micro_ha_2_0",
    "name": "Micro with High Availability",
    "price": 30.0,
    "ramSizeInGb": 1.0,
    "diskSizeInGb": 40,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "small_2_0",
    "name": "Small",
    "price": 30.0,
    "ramSizeInGb": 2.0,
    "diskSizeInGb": 80,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "small_ha_2_0",
    "name": "Small with High Availability",
    "price": 60.0,
    "ramSizeInGb": 2.0,
    "diskSizeInGb": 80,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "medium_2_0",
```

```
    "name": "Medium",
    "price": 60.0,
    "ramSizeInGb": 4.0,
    "diskSizeInGb": 120,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "medium_ha_2_0",
    "name": "Medium with High Availability",
    "price": 120.0,
    "ramSizeInGb": 4.0,
    "diskSizeInGb": 120,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "large_2_0",
    "name": "Large",
    "price": 115.0,
    "ramSizeInGb": 8.0,
    "diskSizeInGb": 240,
    "transferPerMonthInGb": 200,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "large_ha_2_0",
    "name": "Large with High Availability",
    "price": 230.0,
    "ramSizeInGb": 8.0,
    "diskSizeInGb": 240,
    "transferPerMonthInGb": 200,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  }
]
```



```
}
```

Weitere Informationen finden Sie unter [Creating a database in Amazon Lightsail im Amazon Lightsail Developer Guide](#).

- API-Einheiten finden Sie unter [GetRelationalDatabaseBundles](#) Befehlsreferenz. AWS CLI

get-relational-database-events

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-events`.

AWS CLI

Um die Ereignisse für eine relationale Datenbank abzurufen

Im folgenden `get-relational-database-events` Beispiel werden Details zu Ereignissen der letzten 17 Stunden (1020 Minuten) für die angegebene relationale Datenbank angezeigt.

```
aws lightsail get-relational-database-events \
  --relational-database-name Database-1 \
  --duration-in-minutes 1020
```

Ausgabe:

```
{
  "relationalDatabaseEvents": [
    {
      "resource": "Database-1",
      "createdAt": 1571654146.553,
      "message": "Backing up Relational Database",
      "eventCategories": [
        "backup"
      ]
    },
    {
      "resource": "Database-1",
      "createdAt": 1571654249.98,
      "message": "Finished Relational Database backup",
      "eventCategories": [
        "backup"
      ]
    }
  ]
}
```

```
}
```

- API-Einheiten finden Sie unter [GetRelationalDatabaseEvents AWS CLI](#) Befehlsreferenz.

get-relational-database-log-events

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-log-events`.

AWS CLI

Um Protokollereignisse für eine relationale Datenbank abzurufen

Im folgenden `get-relational-database-log-events` Beispiel werden Details zum angegebenen Protokoll zwischen 1570733176 und 1571597176 für eine relationale Datenbank angezeigt. Database1 Die zurückgegebenen Informationen sind so konfiguriert, dass sie von `head` beginnen.

Es wird empfohlen, einen Unix-Zeitkonverter zu verwenden, um die Start- und Endzeiten zu ermitteln.

```
aws lightsail get-relational-database-log-events \  
  --relational-database-name Database1 \  
  --log-stream-name error \  
  --start-from-head \  
  --start-time 1570733176 \  
  --end-time 1571597176
```

Ausgabe:

```
{  
  "resourceLogEvents": [  
    {  
      "createdAt": 1570820267.0,  
      "message": "2019-10-11 18:57:47 20969 [Warning] IP address '192.0.2.0'  
could not be resolved: Name or service not known"  
    },  
    {  
      "createdAt": 1570860974.0,  
      "message": "2019-10-12 06:16:14 20969 [Warning] IP address '8192.0.2.0'  
could not be resolved: Temporary failure in name resolution"  
    },  
    {
```

```

        "createdAt": 1570860977.0,
        "message": "2019-10-12 06:16:17 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
    },
    {
        "createdAt": 1570860979.0,
        "message": "2019-10-12 06:16:19 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
    },
    {
        "createdAt": 1570860981.0,
        "message": "2019-10-12 06:16:21 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
    },
    {
        "createdAt": 1570860982.0,
        "message": "2019-10-12 06:16:22 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
    },
    {
        "createdAt": 1570860984.0,
        "message": "2019-10-12 06:16:24 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
    },
    {
        "createdAt": 1570860986.0,
        "message": "2019-10-12 06:16:26 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
    },
    ...
}
],
"nextBackwardToken":
"eEXAMPLEZXJUZXh0IjoiZnRwb3F3cUpRS1Q5NndMYThxelRUZlFhR3J6c2dKWEEvM2kvajZMzZVVVWpqRDN0YjFXTj
"nextForwardToken":
"eEXAMPLEZXJUZXh0IjoiT09Lb0Z6ZFRJbHhaNEQ5N2tPbkkwRmwwNUxPZjFTbFFwUk1Qbz1SaWgvMwVXbEk4aG56VH
}

```

- API-Einheiten finden Sie [GetRelationalDatabaseLogEvents](#) in der AWS CLI Befehlsreferenz.

get-relational-database-log-streams

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-log-streams`.

AWS CLI

Um die Logstreams für eine relationale Datenbank abzurufen

Das folgende `get-relational-database-log-streams` Beispiel gibt alle verfügbaren Protokolldatenströme für die angegebene relationale Datenbank zurück.

```
aws lightsail get-relational-database-log-streams \  
--relational-database-name Database1
```

Ausgabe:

```
{  
  "logStreams": [  
    "audit",  
    "error",  
    "general",  
    "slowquery"  
  ]  
}
```

- API-Einheiten finden Sie unter [GetRelationalDatabaseLogStreams AWS CLI](#) Befehlsreferenz.

get-relational-database-master-user-password

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-master-user-password`.

AWS CLI

Um das Masterbenutzerkennwort für eine relationale Datenbank abzurufen

Das folgende `get-relational-database-master-user-password` Beispiel gibt Informationen über das Hauptbenutzerkennwort für die angegebene relationale Datenbank zurück.

```
aws lightsail get-relational-database-master-user-password \  
--relational-database-name Database-1
```

Ausgabe:

```
{
  "masterUserPassword": "VEXAMPLEec.9qvX,_t<)Wkf)kwboM,>2",
  "createdAt": 1571259453.959
}
```

- API-Einheiten finden Sie unter [GetRelationalDatabaseMasterUserPassword AWS CLI-Befehlsreferenz](#).

get-relational-database-metric-data

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-metric-data`.

AWS CLI

Um metrische Daten für eine relationale Datenbank abzurufen

Das folgende `get-relational-database-metric-data` Beispiel gibt die Zählsumme der Metrik `DatabaseConnections` über einen Zeitraum von 24 Stunden (86400 Sekunden) zwischen 1570733176 und 1571597176 für eine relationale Datenbank zurück. `Database1`

Es wird empfohlen, einen Unix-Zeitkonverter zu verwenden, um die Start- und Endzeiten zu ermitteln.

```
aws lightsail get-relational-database-metric-data \
  --relational-database-name Database1 \
  --metric-name DatabaseConnections \
  --period 86400 \
  --start-time 1570733176 \
  --end-time 1571597176 \
  --unit Count \
  --statistics Sum
```

Ausgabe:

```
{
  "metricName": "DatabaseConnections",
  "metricData": [
    {
      "sum": 1.0,
      "timestamp": 1571510760.0,
      "unit": "Count"
    }
  ]
}
```

```
  },
  {
    "sum": 1.0,
    "timestamp": 1570733160.0,
    "unit": "Count"
  },
  {
    "sum": 1.0,
    "timestamp": 1570992360.0,
    "unit": "Count"
  },
  {
    "sum": 0.0,
    "timestamp": 1571251560.0,
    "unit": "Count"
  },
  {
    "sum": 721.0,
    "timestamp": 1570819560.0,
    "unit": "Count"
  },
  {
    "sum": 1.0,
    "timestamp": 1571078760.0,
    "unit": "Count"
  },
  {
    "sum": 2.0,
    "timestamp": 1571337960.0,
    "unit": "Count"
  },
  {
    "sum": 684.0,
    "timestamp": 1570905960.0,
    "unit": "Count"
  },
  {
    "sum": 0.0,
    "timestamp": 1571165160.0,
    "unit": "Count"
  },
  {
    "sum": 1.0,
    "timestamp": 1571424360.0,
```

```

        "unit": "Count"
      }
    ]
  }

```

- API-Einheiten finden Sie [GetRelationalDatabaseMetricData](#) in der AWS CLI Befehlsreferenz.

get-relational-database-parameters

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-parameters`.

AWS CLI

Um Parameter für eine relationale Datenbank abzurufen

Das folgende `get-relational-database-parameters` Beispiel gibt Informationen über alle verfügbaren Parameter für die angegebene relationale Datenbank zurück.

```

aws lightsail get-relational-database-parameters \
  --relational-database-name Database-1

```

Ausgabe:

```

{
  "parameters": [
    {
      "allowedValues": "0,1",
      "applyMethod": "pending-reboot",
      "applyType": "dynamic",
      "dataType": "boolean",
      "description": "Automatically set all granted roles as active after the user has authenticated successfully.",
      "isModifiable": true,
      "parameterName": "activate_all_roles_on_login",
      "parameterValue": "0"
    },
    {
      "allowedValues": "0,1",
      "applyMethod": "pending-reboot",
      "applyType": "static",
      "dataType": "boolean",

```

```

        "description": "Controls whether user-defined functions that have only
an xxx symbol for the main function can be loaded",
        "isModifiable": false,
        "parameterName": "allow-suspicious-udfs"
    },
    {
        "allowedValues": "0,1",
        "applyMethod": "pending-reboot",
        "applyType": "dynamic",
        "dataType": "boolean",
        "description": "Sets the autocommit mode",
        "isModifiable": true,
        "parameterName": "autocommit"
    },
    {
        "allowedValues": "0,1",
        "applyMethod": "pending-reboot",
        "applyType": "static",
        "dataType": "boolean",
        "description": "Controls whether the server autogenerates SSL key and
certificate files in the data directory, if they do not already exist.",
        "isModifiable": false,
        "parameterName": "auto_generate_certs"
    },
    ...
}
]
}

```

Weitere Informationen finden Sie unter [Aktualisieren von Datenbankparametern in Amazon Lightsail im Lightsail Dev Guide](#).

- API-Einheiten finden Sie unter [GetRelationalDatabaseParameters](#) Befehlsreferenz.AWS CLI

get-relational-database-snapshot

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-snapshot`.

AWS CLI

Um Informationen über einen Snapshot einer relationalen Datenbank abzurufen

Im folgenden `get-relational-database-snapshot` Beispiel werden Details zum angegebenen Snapshot einer relationalen Datenbank angezeigt.

```
aws lightsail get-relational-database-snapshot \
  --relational-database-snapshot-name Database-1-1571350042
```

Ausgabe:

```
{
  "relationalDatabaseSnapshot": {
    "name": "Database-1-1571350042",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9EXAMPLEaee3643d2",
    "supportCode": "6EXAMPLE3362/1s-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
    "createdAt": 1571350046.238,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabaseSnapshot",
    "tags": [],
    "engine": "mysql",
    "engineVersion": "8.0.16",
    "sizeInGb": 40,
    "state": "available",
    "fromRelationalDatabaseName": "Database-1",
    "fromRelationalDatabaseArn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
    "fromRelationalDatabaseBundleId": "micro_1_0",
    "fromRelationalDatabaseBlueprintId": "mysql_8_0"
  }
}
```

- API-Einheiten finden Sie unter [GetRelationalDatabaseSnapshot AWS CLI Befehlsreferenz](#).

get-relational-database-snapshots

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database-snapshots`.

AWS CLI

Um Informationen über alle Snapshots relationaler Datenbanken abzurufen

Im folgenden `get-relational-database-snapshots` Beispiel werden Details zu allen Snapshots relationaler Datenbanken in der konfigurierten Region angezeigt. AWS

```
aws lightsail get-relational-database-snapshots
```

Ausgabe:

```
{
  "relationalDatabaseSnapshots": [
    {
      "name": "Database-1-1571350042",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9861-6EXAMPLE43d2",
      "supportCode": "6EXAMPLE3362/1s-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
      "createdAt": 1571350046.238,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "RelationalDatabaseSnapshot",
      "tags": [],
      "engine": "mysql",
      "engineVersion": "8.0.16",
      "sizeInGb": 40,
      "state": "available",
      "fromRelationalDatabaseName": "Database-1",
      "fromRelationalDatabaseArn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
      "fromRelationalDatabaseBundleId": "micro_1_0",
      "fromRelationalDatabaseBlueprintId": "mysql_8_0"
    },
    {
      "name": "Database1-Console",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabaseSnapshot/8b94136e-06ec-4b1a-a3fb-5EXAMPLEe1e9",
      "supportCode": "6EXAMPLE3362/1s-9EXAMPLE14b000d34c8d1c432734e137612d5b5c",
      "createdAt": 1571249981.025,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      }
    }
  ]
}
```

```

    },
    "resourceType": "RelationalDatabaseSnapshot",
    "tags": [
      {
        "key": "test"
      }
    ],
    "engine": "mysql",
    "engineVersion": "5.6.44",
    "sizeInGb": 40,
    "state": "available",
    "fromRelationalDatabaseName": "Database1",
    "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/a6161cb7-4535-4f16-9dcf-8EXAMPLE3d4e",
    "fromRelationalDatabaseBundleId": "micro_1_0",
    "fromRelationalDatabaseBlueprintId": "mysql_5_6"
  }
]
}

```

- API-Einheiten finden Sie unter [GetRelationalDatabaseSnapshots AWS CLI-Befehlsreferenz](#).

get-relational-database

Das folgende Codebeispiel zeigt die Verwendung `get-relational-database`.

AWS CLI

Um Informationen über eine relationale Datenbank zu erhalten

Im folgenden `get-relational-database` Beispiel werden Details zur angegebenen relationalen Datenbank angezeigt.

```

aws lightsail get-relational-database \
  --relational-database-name Database-1

```

Ausgabe:

```

{
  "relationalDatabase": {
    "name": "Database-1",

```

```

    "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
    "supportCode": "6EXAMPLE3362/1s-9EXAMPLE8ad863723b62cc8901a8aa6e794ae0d2",
    "createdAt": 1571259453.795,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabase",
    "tags": [],
    "relationalDatabaseBlueprintId": "mysql_8_0",
    "relationalDatabaseBundleId": "micro_1_0",
    "masterDatabaseName": "dbmaster",
    "hardware": {
      "cpuCount": 1,
      "diskSizeInGb": 40,
      "ramSizeInGb": 1.0
    },
    "state": "available",
    "backupRetentionEnabled": false,
    "pendingModifiedValues": {},
    "engine": "mysql",
    "engineVersion": "8.0.16",
    "masterUsername": "dbmasteruser",
    "parameterApplyStatus": "in-sync",
    "preferredBackupWindow": "10:01-10:31",
    "preferredMaintenanceWindow": "sat:11:14-sat:11:44",
    "publiclyAccessible": true,
    "masterEndpoint": {
      "port": 3306,
      "address": "1s-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
  }
}

```

- API-Einheiten finden Sie unter [GetRelationalDatabase AWS CLI](#) Befehlsreferenz.

get-relational-databases

Das folgende Codebeispiel zeigt die Verwendung `get-relational-databases`.

AWS CLI

Um Informationen über alle relationalen Datenbanken zu erhalten

Im folgenden `get-relational-databases` Beispiel werden Details zu allen relationalen Datenbanken in der AWS konfigurierten Region angezeigt.

```
aws lightsail get-relational-databases
```

Ausgabe:

```
{
  "relationalDatabases": [
    {
      "name": "MySQL",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",
      "supportCode": "6EXAMPLE3362/1s-3EXAMPLEa995d8c3b06b4501356e5f2f28e1aeba",
      "createdAt": 1554306019.155,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "RelationalDatabase",
      "tags": [],
      "relationalDatabaseBlueprintId": "mysql_8_0",
      "relationalDatabaseBundleId": "micro_1_0",
      "masterDatabaseName": "dbmaster",
      "hardware": {
        "cpuCount": 1,
        "diskSizeInGb": 40,
        "ramSizeInGb": 1.0
      },
      "state": "available",
      "backupRetentionEnabled": true,
      "pendingModifiedValues": {},
      "engine": "mysql",
      "engineVersion": "8.0.15",
      "latestRestorableTime": 1571686200.0,
      "masterUsername": "dbmasteruser",
      "parameterApplyStatus": "in-sync",
      "preferredBackupWindow": "07:51-08:21",
```

```
    "preferredMaintenanceWindow": "tue:12:18-tue:12:48",
    "publiclyAccessible": true,
    "masterEndpoint": {
      "port": 3306,
      "address":
"ls-3EXAMPLEa995d8c3b06b4501356e5f2fEXAMPLEa.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
  },
  {
    "name": "Postgres",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/
e9780b6b-d0ab-4af2-85f1-1EXAMPLEac68",
    "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEb4ffffb5cec056220c734713e14bd5fcd",
    "createdAt": 1554306000.814,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabase",
    "tags": [],
    "relationalDatabaseBlueprintId": "postgres_11",
    "relationalDatabaseBundleId": "micro_1_0",
    "masterDatabaseName": "dbmaster",
    "hardware": {
      "cpuCount": 1,
      "diskSizeInGb": 40,
      "ramSizeInGb": 1.0
    },
    "state": "available",
    "backupRetentionEnabled": true,
    "pendingModifiedValues": {},
    "engine": "postgres",
    "engineVersion": "11.1",
    "latestRestorableTime": 1571686339.0,
    "masterUsername": "dbmasteruser",
    "parameterApplyStatus": "in-sync",
    "preferredBackupWindow": "06:19-06:49",
    "preferredMaintenanceWindow": "sun:10:19-sun:10:49",
    "publiclyAccessible": false,
    "masterEndpoint": {
      "port": 5432,
```

```

        "address":
        "1s-3EXAMPLEb4ffffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
        },
        "pendingMaintenanceActions": []
    }
]
}

```

- API-Einheiten finden Sie unter [GetRelationalDatabases AWS CLI Befehlsreferenz](#).

get-static-ip

Das folgende Codebeispiel zeigt die Verwendung `get-static-ip`.

AWS CLI

Um Informationen über eine statische IP zu erhalten

Im folgenden `get-static-ip` Beispiel werden Details zur angegebenen statischen IP angezeigt.

```

aws lightsail get-static-ip \
  --static-ip-name StaticIp-1

```

Ausgabe:

```

{
  "staticIp": {
    "name": "StaticIp-1",
    "arn": "arn:aws:lightsail:us-
west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-82e2-2EXAMPLE23ad",
    "supportCode": "6EXAMPLE3362/192.0.2.0",
    "createdAt": 1571071325.076,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "StaticIp",
    "ipAddress": "192.0.2.0",
    "isAttached": false
  }
}

```

- API-Einheiten finden Sie [GetStaticIp](#) unter AWS CLI Befehlsreferenz.

get-static-ips

Das folgende Codebeispiel zeigt die Verwendung `get-static-ips`.

AWS CLI

Um Informationen über alle statischen Daten zu erhalten IPs

Im folgenden `get-static-ips` Beispiel werden Details zu allen statischen Daten IPs in der konfigurierten AWS Region angezeigt.

```
aws lightsail get-static-ips
```

Ausgabe:

```
{
  "staticIps": [
    {
      "name": "StaticIp-1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-8EXAMPLE16f9423ad",
      "supportCode": "6EXAMPLE3362/192.0.2.0",
      "createdAt": 1571071325.076,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "StaticIp",
      "ipAddress": "192.0.2.0",
      "isAttached": false
    },
    {
      "name": "StaticIP-2",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:StaticIp/c61edb40-e5f0-4fd6-ae7c-8EXAMPLE19f8",
      "supportCode": "6EXAMPLE3362/192.0.2.2",
      "createdAt": 1568305385.681,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      }
    }
  ]
}
```



```
    },
    "resourceType": "StaticIp",
    "ipAddress": "192.0.2.2",
    "attachedTo": "WordPress-1",
    "isAttached": true
  }
]
```

- API-Einheiten finden Sie [GetStaticIps](#) in der AWS CLI Befehlsreferenz.

is-vpc-peered

Das folgende Codebeispiel zeigt die Verwendung `is-vpc-peered`.

AWS CLI

Um festzustellen, ob Ihre Amazon Lightsail Virtual Private Cloud über Peering verfügt

Das folgende `is-vpc-peered` Beispiel gibt den Peering-Status der Amazon Lightsail Virtual Private Cloud (VPC) für die angegebene Region zurück. AWS

```
aws lightsail is-vpc-peered \
  --region us-west-2
```

Ausgabe:

```
{
  "isPeered": true
}
```

- API-Einheiten finden Sie [IsVpcPeered](#) in AWS CLI der Befehlsreferenz.

open-instance-public-ports

Das folgende Codebeispiel zeigt die Verwendung `open-instance-public-ports`.

AWS CLI

Um Firewall-Ports für eine Instanz zu öffnen

Im folgenden `open-instance-public-ports` Beispiel wird TCP Port 22 auf der angegebenen Instanz geöffnet.

```
aws lightsail open-instance-public-ports \  
  --instance-name MEAN-2 \  
  --port-info fromPort=22,protocol=TCP,toPort=22
```

Ausgabe:

```
{  
  "operation": {  
    "id": "719744f0-a022-46f2-9f11-6EXAMPLE4642",  
    "resourceName": "MEAN-2",  
    "resourceType": "Instance",  
    "createdAt": 1571072906.849,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationDetails": "22/tcp",  
    "operationType": "OpenInstancePublicPorts",  
    "status": "Succeeded",  
    "statusChangedAt": 1571072906.849  
  }  
}
```

- API-Einheiten finden Sie [OpenInstancePublicPorts](#) in der AWS CLI Befehlsreferenz.

peer-vpc

Das folgende Codebeispiel zeigt die Verwendung `peer-vpc`.

AWS CLI

Um ein Peering mit der virtuellen privaten Cloud von Amazon Lightsail durchzuführen

Im folgenden `peer-vpc` Beispiel wird ein Peering mit der virtuellen privaten Cloud (VPC) von Amazon Lightsail für die angegebene AWS Region durchgeführt.

```
aws lightsail peer-vpc \  
  --region us-west-2
```

```
--region us-west-2
```

Ausgabe:

```
{
  "operation": {
    "id": "787e846a-54ac-497f-bce2-9EXAMPLE5d91",
    "resourceName": "vpc-0EXAMPLEa5261efb3",
    "resourceType": "PeeredVpc",
    "createdAt": 1571694233.104,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "vpc-e2b3eb9b",
    "operationType": "PeeredVpc",
    "status": "Succeeded",
    "statusChangedAt": 1571694233.104
  }
}
```

- API-Einheiten finden Sie unter [PeerVpc AWS CLI-Befehlsreferenz](#).

reboot-instance

Das folgende Codebeispiel zeigt die Verwendung `reboot-instance`.

AWS CLI

Um eine Instanz neu zu starten

Im folgenden `reboot-instance` Beispiel wird die angegebene Instanz neu gestartet.

```
aws lightsail reboot-instance \
  --instance-name MEAN-1
```

Ausgabe:

```
{
  "operations": [
    {
```

```

    "id": "2b679f1c-8b71-4bb4-8e97-8EXAMPLEed93",
    "resourceName": "MEAN-1",
    "resourceType": "Instance",
    "createdAt": 1571694445.49,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "",
    "operationType": "RebootInstance",
    "status": "Succeeded",
    "statusChangedAt": 1571694445.49
  }
]
}

```

- API-Einheiten finden Sie [RebootInstance](#) in der AWS CLI Befehlsreferenz.

reboot-relational-database

Das folgende Codebeispiel zeigt die Verwendung `reboot-relational-database`.

AWS CLI

Um eine relationale Datenbank neu zu starten

Im folgenden `reboot-relational-database` Beispiel wird die angegebene relationale Datenbank neu gestartet.

```

aws lightsail reboot-relational-database \
  --relational-database-name Database-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "e4c980c0-3137-496c-9c91-1EXAMPLEedec2",
      "resourceName": "Database-1",
      "resourceType": "RelationalDatabase",
      "createdAt": 1571694532.91,

```

```

        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "isTerminal": false,
        "operationDetails": "",
        "operationType": "RebootRelationalDatabase",
        "status": "Started",
        "statusChangedAt": 1571694532.91
    }
]
}

```

- API-Einheiten finden Sie unter [RebootRelationalDatabase AWS CLI Befehlsreferenz](#).

release-static-ip

Das folgende Codebeispiel zeigt die Verwendung `release-static-ip`.

AWS CLI

Um eine statische IP zu löschen

Das folgende `release-static-ip` Beispiel löscht die angegebene statische IP.

```

aws lightsail release-static-ip \
  --static-ip-name StaticIp-1

```

Ausgabe:

```

{
  "operations": [
    {
      "id": "e374c002-dc6d-4c7f-919f-2EXAMPLE13ce",
      "resourceName": "StaticIp-1",
      "resourceType": "StaticIp",
      "createdAt": 1571694962.003,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
    }
  ]
}

```

```
        "operationType": "ReleaseStaticIp",
        "status": "Succeeded",
        "statusChangedAt": 1571694962.003
      }
    ]
  }
}
```

- API-Einheiten finden Sie [ReleaseStaticIp](#) in der AWS CLI Befehlsreferenz.

start-instance

Das folgende Codebeispiel zeigt die Verwendung `start-instance`.

AWS CLI

Um eine Instanz zu starten

Das folgende `start-instance` Beispiel startet die angegebene Instanz.

```
aws lightsail start-instance \
  --instance-name WordPress-1
```

Ausgabe:

```
{
  "operations": [
    {
      "id": "f88d2a93-7cea-4165-afce-2d688cb18f23",
      "resourceName": "WordPress-1",
      "resourceType": "Instance",
      "createdAt": 1571695583.463,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "StartInstance",
      "status": "Started",
      "statusChangedAt": 1571695583.463
    }
  ]
}
```

- API-Einheiten finden Sie [StartInstance](#) in der AWS CLI Befehlsreferenz.

start-relational-database

Das folgende Codebeispiel zeigt die Verwendung `start-relational-database`.

AWS CLI

Um eine relationale Datenbank zu starten

Im folgenden `start-relational-database` Beispiel wird die angegebene relationale Datenbank gestartet.

```
aws lightsail start-relational-database \  
  --relational-database-name Database-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "4d5294ec-a38a-4fda-9e37-aEXAMPLE0d24",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1571695998.822,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "StartRelationalDatabase",  
      "status": "Started",  
      "statusChangedAt": 1571695998.822  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [StartRelationalDatabase AWS CLI](#) Befehlsreferenz.

stop-instance

Das folgende Codebeispiel zeigt die Verwendung `stop-instance`.

AWS CLI

Um eine Instanz zu stoppen

Das folgende `stop-instance` Beispiel stoppt die angegebene Instanz.

```
aws lightsail stop-instance \  
--instance-name WordPress-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "265357e2-2943-4d51-888a-1EXAMPLE7585",  
      "resourceName": "WordPress-1",  
      "resourceType": "Instance",  
      "createdAt": 1571695471.134,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "StopInstance",  
      "status": "Started",  
      "statusChangedAt": 1571695471.134  
    }  
  ]  
}
```

- API-Einheiten finden Sie [StopInstance](#) in der AWS CLI Befehlsreferenz.

stop-relational-database

Das folgende Codebeispiel zeigt die Verwendung `stop-relational-database`.

AWS CLI

Um eine relationale Datenbank zu beenden

Im folgenden `stop-relational-database` Beispiel wird die angegebene relationale Datenbank gestoppt.


```
aws lightsail stop-relational-database \  
  --relational-database-name Database-1
```

Ausgabe:

```
{  
  "operations": [  
    {  
      "id": "cc559c19-4adb-41e4-b75b-5EXAMPLE4e61",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1571695526.29,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "StopRelationalDatabase",  
      "status": "Started",  
      "statusChangedAt": 1571695526.29  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [StopRelationalDatabase AWS CLI Befehlsreferenz](#).

unpeer-vpc

Das folgende Codebeispiel zeigt die Verwendung `unpeer-vpc`.

AWS CLI

So heben Sie das Peering der virtuellen privaten Cloud von Amazon Lightsail auf

Im folgenden `unpeer-vpc` Beispiel wird das Peering der virtuellen privaten Cloud (VPC) von Amazon Lightsail für die angegebene Region aufgehoben. AWS

```
aws lightsail unpeer-vpc \  
  --region us-west-2
```

Ausgabe:

```
{
  "operation": {
    "id": "531aca64-7157-47ab-84c6-eEXAMPLEd898",
    "resourceName": "vpc-0EXAMPLEa5261efb3",
    "resourceType": "PeeredVpc",
    "createdAt": 1571694109.945,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "vpc-e2b3eb9b",
    "operationType": "UnpeeredVpc",
    "status": "Succeeded",
    "statusChangedAt": 1571694109.945
  }
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [UnpeerVpc](#).AWS CLI

Macie-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Macie Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-buckets

Das folgende Codebeispiel zeigt die Verwendung `describe-buckets`.

AWS CLI

Um Daten zu einem oder mehreren S3-Buckets abzufragen, die Amazon Macie für Ihr Konto überwacht und analysiert

Im folgenden `describe-buckets` Beispiel werden Metadaten für alle S3-Buckets abgefragt, deren Namen mit `MY-S3` beginnen und sich in der aktuellen Region befinden. AWS

```
aws macie2 describe-buckets \
  --criteria '{"bucketName":{"prefix":"my-S3"}}'
```

Ausgabe:

```
{
  "buckets": [
    {
      "accountId": "123456789012",
      "allowsUnencryptedObjectUploads": "FALSE",
      "bucketArn": "arn:aws:s3:::MY-S3-DOC-EXAMPLE-BUCKET1",
      "bucketCreatedAt": "2020-05-18T19:54:00+00:00",
      "bucketName": "MY-S3-DOC-EXAMPLE-BUCKET1",
      "classifiableObjectCount": 13,
      "classifiableSizeInBytes": 1592088,
      "jobDetails": {
        "isDefinedInJob": "TRUE",
        "isMonitoredByJob": "TRUE",
        "lastJobId": "08c81dc4a2f3377fae45c9ddaexample",
        "lastJobRunTime": "2021-04-26T14:55:30.270000+00:00"
      },
      "lastAutomatedDiscoveryTime": "2022-12-10T19:11:25.364000+00:00",
      "lastUpdated": "2022-12-13T07:33:06.337000+00:00",
      "objectCount": 13,
      "objectCountByEncryptionType": {
        "customerManaged": 0,
        "kmsManaged": 2,
        "s3Managed": 7,
        "unencrypted": 4,
        "unknown": 0
      },
      "publicAccess": {
        "effectivePermission": "NOT_PUBLIC",
        "permissionConfiguration": {
          "accountLevelPermissions": {
```

```
        "blockPublicAccess": {
            "blockPublicAcls": true,
            "blockPublicPolicy": true,
            "ignorePublicAcls": true,
            "restrictPublicBuckets": true
        }
    },
    "bucketLevelPermissions": {
        "accessControlList": {
            "allowsPublicReadAccess": false,
            "allowsPublicWriteAccess": false
        },
        "blockPublicAccess": {
            "blockPublicAcls": true,
            "blockPublicPolicy": true,
            "ignorePublicAcls": true,
            "restrictPublicBuckets": true
        },
        "bucketPolicy": {
            "allowsPublicReadAccess": false,
            "allowsPublicWriteAccess": false
        }
    }
}
},
"region": "us-west-2",
"replicationDetails": {
    "replicated": false,
    "replicatedExternally": false,
    "replicationAccounts": []
},
"sensitivityScore": 78,
"serverSideEncryption": {
    "kmsMasterKeyId": null,
    "type": "NONE"
},
"sharedAccess": "NOT_SHARED",
"sizeInBytes": 4549746,
"sizeInBytesCompressed": 0,
"tags": [
    {
        "key": "Division",
        "value": "HR"
    }
],
```

```
        {
            "key": "Team",
            "value": "Recruiting"
        }
    ],
    "unclassifiableObjectCount": {
        "fileType": 0,
        "storageClass": 0,
        "total": 0
    },
    "unclassifiableObjectSizeInBytes": {
        "fileType": 0,
        "storageClass": 0,
        "total": 0
    },
    "versioning": true
},
{
    "accountId": "123456789012",
    "allowsUnencryptedObjectUploads": "TRUE",
    "bucketArn": "arn:aws:s3:::MY-S3-DOC-EXAMPLE-BUCKET2",
    "bucketCreatedAt": "2020-11-25T18:24:38+00:00",
    "bucketName": "MY-S3-DOC-EXAMPLE-BUCKET2",
    "classifiableObjectCount": 8,
    "classifiableSizeInBytes": 133810,
    "jobDetails": {
        "isDefinedInJob": "TRUE",
        "isMonitoredByJob": "FALSE",
        "lastJobId": "188d4f6044d621771ef7d65f2example",
        "lastJobRunTime": "2021-04-09T19:37:11.511000+00:00"
    },
    "lastAutomatedDiscoveryTime": "2022-12-12T19:11:25.364000+00:00",
    "lastUpdated": "2022-12-13T07:33:06.337000+00:00",
    "objectCount": 8,
    "objectCountByEncryptionType": {
        "customerManaged": 0,
        "kmsManaged": 0,
        "s3Managed": 8,
        "unencrypted": 0,
        "unknown": 0
    },
    "publicAccess": {
        "effectivePermission": "NOT_PUBLIC",
        "permissionConfiguration": {
```

```
    "accountLevelPermissions": {
      "blockPublicAccess": {
        "blockPublicAcls": true,
        "blockPublicPolicy": true,
        "ignorePublicAcls": true,
        "restrictPublicBuckets": true
      }
    },
    "bucketLevelPermissions": {
      "accessControlList": {
        "allowsPublicReadAccess": false,
        "allowsPublicWriteAccess": false
      },
      "blockPublicAccess": {
        "blockPublicAcls": true,
        "blockPublicPolicy": true,
        "ignorePublicAcls": true,
        "restrictPublicBuckets": true
      },
      "bucketPolicy": {
        "allowsPublicReadAccess": false,
        "allowsPublicWriteAccess": false
      }
    }
  },
  "region": "us-west-2",
  "replicationDetails": {
    "replicated": false,
    "replicatedExternally": false,
    "replicationAccounts": []
  },
  "sensitivityScore": 95,
  "serverSideEncryption": {
    "kmsMasterKeyId": null,
    "type": "AES256"
  },
  "sharedAccess": "EXTERNAL",
  "sizeInBytes": 175978,
  "sizeInBytesCompressed": 0,
  "tags": [
    {
      "key": "Division",
      "value": "HR"
    }
  ]
}
```

```
    },
    {
      "key": "Team",
      "value": "Recruiting"
    }
  ],
  "unclassifiableObjectCount": {
    "fileType": 3,
    "storageClass": 0,
    "total": 3
  },
  "unclassifiableObjectSizeInBytes": {
    "fileType": 2999826,
    "storageClass": 0,
    "total": 2999826
  },
  "versioning": true
}
]
```

Weitere Informationen finden Sie unter [Filtern Ihres S3-Bucket-Inventars](#) im Amazon Macie Macie-Benutzerhandbuch.

- API-Einheiten finden Sie unter [Describe Buckets AWS CLI Befehlsreferenz](#).

Amazon Managed Grafana-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Managed Grafana Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

list-workspaces

Das folgende Codebeispiel zeigt die Verwendung `list-workspaces`.

AWS CLI

Um Arbeitsbereiche für das Konto in der durch die Benutzeranmeldedaten angegebenen Region aufzulisten

Das folgende `list-workspaces` Beispiel listet Grafana-Arbeitsbereiche für die Region des Kontos auf.

```
aws grafana list-workspaces
```

Ausgabe:

```
{
  "workspaces": [
    {
      "authentication": {
        "providers": [
          "AWS_SSO"
        ]
      },
      "created": "2022-04-04T16:20:21.796000-07:00",
      "description": "to test tags",
      "endpoint": "g-949e7b44df.grafana-workspace.us-east-1.amazonaws.com",
      "grafanaVersion": "8.2",
      "id": "g-949e7b44df",
      "modified": "2022-04-04T16:20:21.796000-07:00",
      "name": "testtag2",
      "notificationDestinations": [
        "SNS"
      ],
      "status": "ACTIVE"
    },
    {
      "authentication": {
        "providers": [
          "AWS_SSO"
        ]
      }
    }
  ]
}
```



```
    ]
  },
  "created": "2022-04-20T10:22:15.115000-07:00",
  "description": "ww",
  "endpoint": "g-bffa51ed1b.grafana-workspace.us-east-1.amazonaws.com",
  "grafanaVersion": "8.2",
  "id": "g-bffa51ed1b",
  "modified": "2022-04-20T10:22:15.115000-07:00",
  "name": "ww",
  "notificationDestinations": [
    "SNS"
  ],
  "status": "ACTIVE"
}
]
```

- API-Einheiten finden Sie [ListWorkspaces](#) in der AWS CLI Befehlsreferenz.

MediaConnect Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaConnect.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-flow-outputs

Das folgende Codebeispiel zeigt die Verwendung `add-flow-outputs`.

AWS CLI

Um Ausgaben zu einem Flow hinzuzufügen

Das folgende `add-flow-outputs` Beispiel fügt Ausgaben zum angegebenen Flow hinzu.

```
aws mediacconnect add-flow-outputs \  
--flow-arn arn:aws:mediacconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \  
--outputs Description='NYC  
stream',Destination=192.0.2.12,Name=NYC,Port=3333,Protocol=rtp-  
fec,SmoothingLatency=100 Description='LA  
stream',Destination=203.0.113.9,Name=LA,Port=4444,Protocol=rtp-  
fec,SmoothingLatency=100
```

Ausgabe:

```
{  
  "Outputs": [  
    {  
      "Port": 3333,  
      "OutputArn": "arn:aws:mediacconnect:us-  
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",  
      "Name": "NYC",  
      "Description": "NYC stream",  
      "Destination": "192.0.2.12",  
      "Transport": {  
        "Protocol": "rtp-fec",  
        "SmoothingLatency": 100  
      }  
    },  
    {  
      "Port": 4444,  
      "OutputArn": "arn:aws:mediacconnect:us-  
east-1:111122223333:output:2-987655dEF67hiJ89-c34de5fG678h:LA",  
      "Name": "LA",  
      "Description": "LA stream",  
      "Destination": "203.0.113.9",  
      "Transport": {  
        "Protocol": "rtp-fec",  
        "SmoothingLatency": 100  
      }  
    }  
  ]  
}
```

```

    ],
    "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}

```

Weitere Informationen finden Sie unter [Hinzufügen von Ausgaben zu einem Flow](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [AddFlowOutputs](#) in der AWS CLI Befehlsreferenz.

create-flow

Das folgende Codebeispiel zeigt die Verwendung `create-flow`.

AWS CLI

Um einen Flow zu erstellen

Im folgenden `create-flow` Beispiel wird ein Flow mit der angegebenen Konfiguration erstellt.

```

aws mediacconnect create-flow \
  --availability-zone us-west-2c \
  --name ExampleFlow \
  --source Description='Example source,
backup', IngestPort=1055, Name=BackupSource, Protocol=rtp, WhitelistCidr=10.24.34.0/23

```

Ausgabe:

```

{
  "Flow": {
    "FlowArn": "arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:ExampleFlow",
    "AvailabilityZone": "us-west-2c",
    "EgressIp": "54.245.71.21",
    "Source": {
      "IngestPort": 1055,
      "SourceArn": "arn:aws:mediacconnect:us-
east-1:123456789012:source:2-3aBC45dEF67hiJ89-c34de5fG678h:BackupSource",
      "Transport": {
        "Protocol": "rtp",
        "MaxBitrate": 80000000
      },
      "Description": "Example source, backup",
    }
  }
}

```

```

    "IngestIp": "54.245.71.21",
    "WhitelistCidr": "10.24.34.0/23",
    "Name": "mySource"
  },
  "Entitlements": [],
  "Name": "ExampleFlow",
  "Outputs": [],
  "Status": "STANDBY",
  "Description": "Example source, backup"
}
}

```

Weitere Informationen finden Sie unter [Creating a Flow](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [CreateFlow](#) in der AWS CLI Befehlsreferenz.

delete-flow

Das folgende Codebeispiel zeigt die Verwendung `delete-flow`.

AWS CLI

Um einen Flow zu löschen

Im folgenden `delete-flow` Beispiel wird der angegebene Flow gelöscht.

```

aws mediacconnect delete-flow \
  --flow-arn arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow

```

Ausgabe:

```

{
  "FlowArn": "arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Status": "DELETING"
}

```

Weitere Informationen finden Sie unter [Löschen eines Flows](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [DeleteFlow](#) in der AWS CLI Befehlsreferenz.

describe-flow

Das folgende Codebeispiel zeigt die Verwendung `describe-flow`.

AWS CLI

Um die Details eines Flows anzuzeigen

Im folgenden `describe-flow` Beispiel werden die Details des angegebenen Flows angezeigt ARN, z. B. Availability Zone, Status, Quelle, Berechtigungen und Ausgaben.

```
aws mediacconnect describe-flow \
  --flow-arn arn:aws:mediacconnect:us-
  east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Ausgabe:

```
{
  "Flow": {
    "EgressIp": "54.201.4.39",
    "AvailabilityZone": "us-west-2c",
    "Status": "ACTIVE",
    "FlowArn": "arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Entitlements": [
      {
        "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:123456789012:entitlement:1-AaBb11CcDd22EeFf-34DE5fG12AbC:MyEntitlement",
        "Description": "Assign to this account",
        "Name": "MyEntitlement",
        "Subscribers": [
          "444455556666"
        ]
      }
    ],
    "Description": "NYC awards show",
    "Name": "AwardsShow",
    "Outputs": [
      {
        "Port": 2355,
        "Name": "NYC",
        "Transport": {
          "SmoothingLatency": 0,
```

```

        "Protocol": "rtp-fec"
      },
      "OutputArn": "arn:aws:mediacconnect:us-
east-1:123456789012:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
      "Destination": "192.0.2.0"
    },
    {
      "Port": 3025,
      "Name": "LA",
      "Transport": {
        "SmoothingLatency": 0,
        "Protocol": "rtp-fec"
      },
      "OutputArn": "arn:aws:mediacconnect:us-
east-1:123456789012:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
      "Destination": "192.0.2.0"
    }
  ],
  "Source": {
    "IngestIp": "54.201.4.39",
    "SourceArn": "arn:aws:mediacconnect:us-
east-1:123456789012:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
    "Transport": {
      "MaxBitrate": 80000000,
      "Protocol": "rtp"
    },
    "IngestPort": 1069,
    "Description": "Saturday night show",
    "Name": "ShowSource",
    "WhitelistCidr": "10.24.34.0/23"
  }
}

```

Weitere Informationen finden Sie im AWS Elemental MediaConnect User Guide unter [Details eines Flows anzeigen](#).

- API-Einheiten finden Sie [DescribeFlow](#) in der AWS CLI Befehlsreferenz.

grant-flow-entitlements

Das folgende Codebeispiel zeigt die Verwendung `grant-flow-entitlements`.

AWS CLI

Um eine Berechtigung für einen Flow zu gewähren

Im folgenden `grant-flow-entitlements` Beispiel wird dem angegebenen vorhandenen Flow die Berechtigung erteilt, Ihre Inhalte mit einem anderen AWS Konto zu teilen.

```
aws mediconnect grant-flow-entitlements \
  --flow-arn arn:aws:mediconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --entitlements Description='For AnyCompany',Encryption={'Algorithm=aes128,KeyType=static-key,RoleArn=arn:aws:iam::111122223333:role/MediaConnect-ASM,SecretArn=arn:aws:secretsmanager:us-west-2:111122223333:secret:mySecret1'},Name=AnyCompany_Entitlement,Subscribers=444455556666
  Description='For Example Corp',Name=ExampleCorp,Subscribers=777788889999
```

Ausgabe:

```
{
  "Entitlements": [
    {
      "Name": "AnyCompany_Entitlement",
      "EntitlementArn": "arn:aws:mediconnect:us-west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
      "Subscribers": [
        "444455556666"
      ],
      "Description": "For AnyCompany",
      "Encryption": {
        "SecretArn": "arn:aws:secretsmanager:us-west-2:111122223333:secret:mySecret1",
        "Algorithm": "aes128",
        "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
        "KeyType": "static-key"
      }
    },
    {
      "Name": "ExampleCorp",
      "EntitlementArn": "arn:aws:mediconnect:us-west-2:111122223333:entitlement:1-3333cccc4444dddd-1111aaaa2222:ExampleCorp",
      "Subscribers": [
        "777788889999"
      ]
    }
  ]
}
```

```
    ],
    "Description": "For Example Corp"
  }
],
"FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

Weitere Informationen finden Sie unter [Gewährung einer Berechtigung für einen Flow](#) im AWS Elemental-Benutzerhandbuch. MediaConnect

- API-Einheiten finden Sie [GrantFlowEntitlements](#) in der AWS CLI Befehlsreferenz.

list-entitlements

Das folgende Codebeispiel zeigt die Verwendung `list-entitlements`.

AWS CLI

Um eine Liste von Berechtigungen anzuzeigen

Im folgenden `list-entitlements` Beispiel wird eine Liste aller Berechtigungen angezeigt, die dem Konto gewährt wurden.

```
aws mediacconnect list-entitlements
```

Ausgabe:

```
{
  "Entitlements": [
    {
      "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
      "EntitlementName": "MyEntitlement"
    }
  ]
}
```

Weitere Informationen finden Sie [ListEntitlements](#) in der AWS Elemental Reference MediaConnect API.

- API-Einheiten finden Sie unter [ListEntitlements AWS CLI](#) Befehlsreferenz.

list-flows

Das folgende Codebeispiel zeigt die Verwendung `list-flows`.

AWS CLI

Um eine Liste von Flows anzuzeigen

Im folgenden `list-flows` Beispiel wird eine Liste von Flows angezeigt.

```
aws mediaconnect list-flows
```

Ausgabe:

```
{
  "Flows": [
    {
      "Status": "STANDBY",
      "SourceType": "OWNED",
      "AvailabilityZone": "us-west-2a",
      "Description": "NYC awards show",
      "Name": "AwardsShow",
      "FlowArn": "arn:aws:mediaconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
    },
    {
      "Status": "STANDBY",
      "SourceType": "OWNED",
      "AvailabilityZone": "us-west-2c",
      "Description": "LA basketball game",
      "Name": "BasketballGame",
      "FlowArn": "arn:aws:mediaconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Eine Liste von Flows anzeigen](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [ListFlows](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine MediaConnect Ressource aufzulisten

Im folgenden `list-tags-for-resource` Beispiel werden die Tag-Schlüssel und -Werte angezeigt, die der angegebenen MediaConnect Ressource zugeordnet sind.

```
aws mediconnect list-tags-for-resource \
  --resource-arn arn:aws:mediconnect:us-
  east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
```

Ausgabe:

```
{
  "Tags": {
    "region": "west",
    "stage": "prod"
  }
}
```

Weitere Informationen finden Sie unter [ListTagsForResource TagResource](#), [UntagResource](#) in der AWS Elemental MediaConnect API Reference.

- API-Einheiten finden Sie unter [ListTagsForResource AWS CLI Befehlsreferenz](#).

remove-flow-output

Das folgende Codebeispiel zeigt die Verwendung `remove-flow-output`.

AWS CLI

Um eine Ausgabe aus einem Flow zu entfernen

Im folgenden `remove-flow-output` Beispiel wird eine Ausgabe aus dem angegebenen Flow entfernt.

```
aws mediconnect remove-flow-output \
  --flow-arn arn:aws:mediconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
```

```
--output-arn arn:aws:mediacconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC
```

Ausgabe:

```
{
  "FlowArn": "arn:aws:mediacconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "OutputArn": "arn:aws:mediacconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC"
}
```

Weitere Informationen finden Sie unter [Entfernen von Ausgaben aus einem Flow](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [RemoveFlowOutput](#) in der AWS CLI Befehlsreferenz.

revoke-flow-entitlement

Das folgende Codebeispiel zeigt die Verwendung von `revoke-flow-entitlement`.

AWS CLI

Um einen Anspruch zu widerrufen

Im folgenden `revoke-flow-entitlement` Beispiel wird eine Berechtigung für den angegebenen Flow widerrufen.

```
aws mediacconnect revoke-flow-entitlement \
  --flow-arn arn:aws:mediacconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --entitlement-arn arn:aws:mediacconnect:us-west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
```

Ausgabe:

```
{
  "FlowArn": "arn:aws:mediacconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "EntitlementArn": "arn:aws:mediacconnect:us-west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement"
```

```
}
```

Weitere Informationen finden Sie unter [Widerrufen einer Berechtigung](#) im Elemental User Guide.AWS MediaConnect

- APIEinzelheiten finden Sie [RevokeFlowEntitlement](#)in AWS CLI der Befehlsreferenz.

start-flow

Das folgende Codebeispiel zeigt die Verwendungstart-flow.

AWS CLI

Um einen Flow zu starten

Im folgenden start-flow Beispiel wird der angegebene Flow gestartet.

```
aws mediaconnect start-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",  
  "Status": "STARTING"  
}
```

Weitere Informationen finden Sie unter [Einen Flow starten](#) im AWS Elemental MediaConnect User Guide.

- APIEinzelheiten finden Sie [StartFlow](#)in der AWS CLI Befehlsreferenz.

stop-flow

Das folgende Codebeispiel zeigt die Verwendungstop-flow.

AWS CLI

Um einen Flow zu stoppen

Im folgenden `stop-flow` Beispiel wird der angegebene Flow gestoppt.

```
aws mediaconnect stop-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Ausgabe:

```
{  
  "Status": "STOPPING",  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"  
}
```

Weitere Informationen finden Sie unter [Stoppen eines Flows](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [StopFlow](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer MediaConnect Ressource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel wird der angegebenen MediaConnect Ressource ein Tag mit einem Schlüsselnamen und einem Schlüsselwert hinzugefügt.

```
aws mediaconnect tag-resource \  
  --resource-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame  
  --tags region=west
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [ListTagsForResource TagResource](#), [UntagResource](#) in der AWS Elemental MediaConnect API Reference.

- API-Einheiten finden Sie unter [TagResource AWS CLI](#) Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer MediaConnect Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem angegebenen Schlüsselnamen und dem zugehörigen Wert aus einer MediaConnect Ressource entfernt.

```
aws mediconnect untag-resource \  
  --resource-arn arn:aws:mediconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \  
  --tag-keys region
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [ListTagsForResource TagResource, UntagResource](#) in der AWS Elemental MediaConnect API Reference.

- API-Einheiten finden Sie unter [UntagResource AWS CLI Befehlsreferenz](#).

update-flow-entitlement

Das folgende Codebeispiel zeigt die Verwendung `update-flow-entitlement`.

AWS CLI

Um einen Anspruch zu aktualisieren

Im folgenden `update-flow-entitlement` Beispiel wird die angegebene Berechtigung mit einer neuen Beschreibung und einem neuen Abonnenten aktualisiert.

```
aws mediconnect update-flow-entitlement \  
  --flow-arn arn:aws:mediconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \  
  --entitlement-arn arn:aws:mediconnect:us-  
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement \  
  --description 'For AnyCompany Affiliate' \  
  --subscribers 777788889999
```

Ausgabe:

```
{
  "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "Entitlement": {
    "Name": "AnyCompany_Entitlement",
    "Description": "For AnyCompany Affiliate",
    "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
    "Encryption": {
      "KeyType": "static-key",
      "Algorithm": "aes128",
      "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
      "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
    },
    "Subscribers": [
      "777788889999"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer Berechtigung](#) im AWS Elemental User Guide MediaConnect .

- API-Einheiten finden Sie [UpdateFlowEntitlement](#) in der AWS CLI Befehlsreferenz.

update-flow-output

Das folgende Codebeispiel zeigt die Verwendung `update-flow-output`.

AWS CLI

Um eine Ausgabe in einem Flow zu aktualisieren

Im folgenden `update-flow-output` Beispiel wird eine Ausgabe für den angegebenen Flow aktualisiert.

```
aws mediacconnect update-flow-output \
  --flow-arn arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
```

```
--output-arn arn:aws:mediacconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC \  
--port 3331
```

Ausgabe:

```
{  
  "FlowArn": "arn:aws:mediacconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",  
  "Output": {  
    "Name": "NYC",  
    "Port": 3331,  
    "Description": "NYC stream",  
    "Transport": {  
      "Protocol": "rtp-fec",  
      "SmoothingLatency": 100  
    },  
    "OutputArn": "arn:aws:mediacconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",  
    "Destination": "192.0.2.12"  
  }  
}
```

Weitere Informationen finden Sie unter [Aktualisieren von Ausgaben in einem Flow](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [UpdateFlowOutput](#) in der AWS CLI Befehlsreferenz.

update-flow-source

Das folgende Codebeispiel zeigt die Verwendung `update-flow-source`.

AWS CLI

Um die Quelle eines vorhandenen Flows zu aktualisieren

Im folgenden `update-flow-source` Beispiel wird die Quelle eines vorhandenen Flows aktualisiert.

```
aws mediacconnect update-flow-source \  
  --flow-arn arn:aws:mediacconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \  
  --output-arn arn:aws:mediacconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC \  
  --port 3331
```



```

--source-arn arn:aws:mediacconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource \
--description 'Friday night show' \
--ingest-port 3344 \
--protocol rtp-fec \
--whitelist-cidr 10.24.34.0/23

```

Ausgabe:

```

{
  "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Source": {
    "IngestIp": "34.210.136.56",
    "WhitelistCidr": "10.24.34.0/23",
    "Transport": {
      "Protocol": "rtp-fec"
    },
    "IngestPort": 3344,
    "Name": "ShowSource",
    "Description": "Friday night show",
    "SourceArn": "arn:aws:mediacconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren der Quelle eines Flusses](#) im AWS Elemental MediaConnect User Guide.

- API-Einheiten finden Sie [UpdateFlowSource](#) in der AWS CLI Befehlsreferenz.

MediaConvert Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaConvert.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-job`.

AWS CLI

Um einen Job abubrechen, der sich in einer Warteschlange befindet

Im folgenden `cancel-job` Beispiel wird der Job mit der ID `1234567891234-abc123` storniert. Sie können einen Job nicht stornieren, dessen Verarbeitung der Dienst begonnen hat.

```
aws mediaconvert cancel-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --id 1234567891234-abc123
```

Um Ihren kontospezifischen Endpunkt zu erhalten `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Jobs](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [CancelJob](#) in der AWS CLI Befehlsreferenz.

create-job-template

Das folgende Codebeispiel zeigt die Verwendung `create-job-template`.

AWS CLI

So erstellen Sie eine neue Aufgabenvorlage

Im folgenden `create-job-template` Beispiel wird eine Auftragsvorlage mit den Transcodierungseinstellungen erstellt, die in der Datei angegeben sind `job-template.json`, die sich auf Ihrem System befindet.

```
aws mediaconvert create-job-template \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --name JobTemplate1 \  
  --cli-input-json file://~/job-template.json
```

Wenn Sie Ihre JSON Jobvorlagendatei erstellen, indem Sie die Datei verwenden `get-job-template` und anschließend ändern, entfernen Sie das `JobTemplate` Objekt, behalten aber das untergeordnete Objekt `Settings` darin. Achten Sie außerdem darauf, die folgenden Schlüssel-Wert-Paare zu entfernen: `LastUpdated`, `ArnType`, und `CreatedAt`. Sie können die Kategorie, die Beschreibung, den Namen und die Warteschlange entweder in der JSON Datei oder in der Befehlszeile angeben.

Um Ihren kontospezifischen Endpunkt zu erhalten `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Wenn Ihre Anfrage erfolgreich ist, gibt der Service die JSON Spezifikation für die von Ihnen erstellte Jobvorlage zurück.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Job Templates](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [CreateJobTemplate](#) in der AWS CLI Befehlsreferenz.

create-job

Das folgende Codebeispiel zeigt die Verwendung `create-job`.

AWS CLI

Um einen Job zu erstellen

Im folgenden `create-job` Beispiel wird ein Transcodierungsauftrag mit den Einstellungen erstellt, die in einer Datei angegeben sind `job.json`, die sich auf dem System befindet, von dem aus Sie den Befehl senden. Diese JSON Jobspezifikation kann jede Einstellung einzeln angeben, auf eine Auftragsvorlage verweisen oder auf Ausgabevorgaben verweisen.

```
aws mediaconvert create-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --cli-input-json file://~/job.json
```

```
--region region-name-1 \  
--cli-input-json file:///~/job.json
```

Sie können die AWS MediaConvert Elemental-Konsole verwenden, um die JSON Jobspezifikation zu generieren, indem Sie Ihre Job-Einstellungen auswählen und dann unten JSON im Job-Bereich die Option Job anzeigen wählen.

Um Ihren kontospezifischen Endpunkt zu `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Wenn Ihre Anfrage erfolgreich ist, gibt der Service die JSON Jobspezifikation zurück, die Sie mit Ihrer Anfrage gesendet haben.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Jobs](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [CreateJob](#) in der AWS CLI Befehlsreferenz.

create-preset

Das folgende Codebeispiel zeigt die Verwendung `create-preset`.

AWS CLI

Um eine benutzerdefinierte Ausgabevoreinstellung zu erstellen

Im folgenden `create-preset` Beispiel wird eine benutzerdefinierte Ausgabevorgabe erstellt, die auf den in der Datei angegebenen Ausgabeeinstellungen basiert `preset.json`. Sie können die Kategorie, die Beschreibung und den Namen entweder in der JSON Datei oder in der Befehlszeile angeben.

```
aws mediaconvert create-preset \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --cli-input-json file:///~/preset.json
```

Wenn Sie Ihre JSON Voreinstellungsdatei erstellen, indem Sie die Ausgabedatei verwenden `get-preset` und anschließend ändern, stellen Sie sicher, dass Sie die folgenden Schlüssel-Wert-Paare entfernen: `LastUpdated`, `ArnType`, und `CreatedAt`

Um Ihren kontospezifischen Endpunkt zu erhalten `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Output Presets](#) im AWS Elemental User Guide MediaConvert .

- API-Einheiten finden Sie [CreatePreset](#) in AWS CLI der Befehlsreferenz.

create-queue

Das folgende Codebeispiel zeigt die Verwendung `create-queue`.

AWS CLI

Um eine benutzerdefinierte Warteschlange zu erstellen

Im folgenden `create-queue` Beispiel wird eine benutzerdefinierte Transcodierungswarteschlange erstellt.

```
aws mediaconvert create-queue \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --name Queue1 \  
  --description "Keep this queue empty unless job is urgent."
```

Um Ihren kontospezifischen Endpunkt abzurufen `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Ausgabe:

```
{  
  "Queue": {  
    "Status": "ACTIVE",  
    "Name": "Queue1",  
    "LastUpdated": 1518034928,  
    "Arn": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",  
    "Type": "CUSTOM",  
    "CreatedAt": 1518034928,  
    "Description": "Keep this queue empty unless job is urgent."  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Queues](#) im AWS Elemental User Guide MediaConvert .

- API-Einheiten finden Sie [CreateQueue](#) in AWS CLI der Befehlsreferenz.

delete-job-template

Das folgende Codebeispiel zeigt die Verwendung `delete-job-template`.

AWS CLI

Um eine Jobvorlage zu löschen

Im folgenden `delete-job-template` Beispiel wird die angegebene benutzerdefinierte Jobvorlage gelöscht.

```
aws mediaconvert delete-job-template \  
  --name "DASH Streaming" \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Führen Sie `aws mediaconvert list-job-templates` den Befehl aus, um zu bestätigen, dass Ihre Vorlage gelöscht wurde.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Job Templates](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [DeleteJobTemplate](#) in der AWS CLI Befehlsreferenz.

delete-preset

Das folgende Codebeispiel zeigt die Verwendung `delete-preset`.

AWS CLI

Um eine benutzerdefinierte On-Demand-Warteschlange zu löschen

Im folgenden `delete-preset` Beispiel wird die angegebene benutzerdefinierte Voreinstellung gelöscht.

```
aws mediaconvert delete-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Führen Sie `aws mediaconvert list-presets` den Befehl aus, um zu bestätigen, dass Ihre Voreinstellung gelöscht wurde.

Weitere Informationen finden Sie unter [Arbeiten mit AWS MediaConvert Elemental-Ausgabevoreinstellungen](#) im AWS Elemental-Benutzerhandbuch. MediaConvert

- API-Einheiten finden Sie [DeletePreset](#) in AWS CLI der Befehlsreferenz.

delete-queue

Das folgende Codebeispiel zeigt die Verwendung `delete-queue`.

AWS CLI

Um eine benutzerdefinierte On-Demand-Warteschlange zu löschen

Im folgenden `delete-queue` Beispiel wird die angegebene benutzerdefinierte On-Demand-Warteschlange gelöscht.

Sie können Ihre Standardwarteschlange nicht löschen. Sie können keine reservierte Warteschlange löschen, die ein aktives Preismodell hat oder unverarbeitete Aufträge enthält.

```
aws mediaconvert delete-queue \  
  --name Customer1 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Führen Sie `aws mediaconvert list-queues` den Befehl aus, um zu bestätigen, dass Ihre Warteschlange gelöscht wurde.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Queues](#) im AWS Elemental User Guide MediaConvert .

- API-Einheiten finden Sie [DeleteQueue](#) in AWS CLI der Befehlsreferenz.

describe-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-endpoints`.

AWS CLI

Um Ihren kontospezifischen Endpunkt zu erhalten

Im folgenden `describe-endpoints` Beispiel wird der Endpunkt abgerufen, den Sie benötigen, um jede andere Anfrage an den Dienst zu senden.

```
aws mediaconvert describe-endpoints
```

Ausgabe:

```
{
  "Endpoints": [
    {
      "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit der MediaConvert Verwendung von API in der AWS](#) Elementar-Referenz MediaConvert API.

- API-Einheiten finden Sie unter [DescribeEndpoints AWS CLI](#) Befehlsreferenz.

get-job-template

Das folgende Codebeispiel zeigt die Verwendung `get-job-template`.

AWS CLI

Um Details für eine Jobvorlage abzurufen

Im folgenden `get-job-template` Beispiel wird die JSON-Definition der angegebenen benutzerdefinierten Jobvorlage angezeigt.

```
aws mediaconvert get-job-template \
  --name "DASH Streaming" \
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```

Ausgabe:


```
{
  "JobTemplate": {
    "StatusUpdateInterval": "SECONDS_60",
    "LastUpdated": 1568652998,
    "Description": "Create a DASH streaming ABR stack",
    "CreatedAt": 1568652998,
    "Priority": 0,
    "Name": "DASH Streaming",
    "Settings": {
      ...<truncatedforbrevity>...
    },
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH
Streaming",
    "Type": "CUSTOM"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Job Templates](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [GetJobTemplate](#) in der AWS CLI Befehlsreferenz.

get-job

Das folgende Codebeispiel zeigt die Verwendung `get-job`.

AWS CLI

Um Details für einen bestimmten Job abzurufen

Im folgenden Beispiel werden die Informationen für den Job mit der ID `1234567890987-1ab2c3` angefordert, was in diesem Beispiel mit einem Fehler endete.

```
aws mediaconvert get-job \
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
  --region region-name-1 \
  --id 1234567890987-1ab2c3
```

Um Ihren kontospezifischen Endpunkt abzurufen `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Wenn Ihre Anfrage erfolgreich ist, gibt der Service eine JSON Datei mit Jobinformationen zurück, einschließlich Jobeinstellungen, allen zurückgegebenen Fehlern und anderen Jobdaten, wie folgt:

```
{
  "Job": {
    "Status": "ERROR",
    "Queue": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",
    "Settings": {
      ...<truncated for brevity>...
    },
    "ErrorMessage": "Unable to open input file [s3://my-input-bucket/file-name.mp4]: [Failed probe/open: [Failed to read data: AssumeRole failed]]",
    "ErrorCode": 1434,
    "Role": "arn:aws:iam::012345678998:role/MediaConvertServiceRole",
    "Arn": "arn:aws:mediaconvert:us-west-1:012345678998:jobs/1234567890987-1ab2c3",
    "UserMetadata": {},
    "Timing": {
      "FinishTime": 1517442131,
      "SubmitTime": 1517442103,
      "StartTime": 1517442104
    },
    "Id": "1234567890987-1ab2c3",
    "CreatedAt": 1517442103
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Jobs](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [GetJob](#) in der AWS CLI Befehlsreferenz.

get-preset

Das folgende Codebeispiel zeigt die Verwendung get-preset.

AWS CLI

Um Details für eine bestimmte Voreinstellung abzurufen

Im folgenden get-preset Beispiel wird die JSON Definition der angegebenen benutzerdefinierten Voreinstellung angefordert.

```
aws mediaconvert get-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "Preset": {  
    "Description": "Creates basic MP4 file. No filtering or preprocessing.",  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4",  
    "LastUpdated": 1568843141,  
    "Name": "SimpleMP4",  
    "Settings": {  
      "ContainerSettings": {  
        "Mp4Settings": {  
          "FreeSpaceBox": "EXCLUDE",  
          "CslgAtom": "INCLUDE",  
          "MoovPlacement": "PROGRESSIVE_DOWNLOAD"  
        },  
        "Container": "MP4"  
      },  
      "AudioDescriptions": [  
        {  
          "LanguageCodeControl": "FOLLOW_INPUT",  
          "AudioTypeControl": "FOLLOW_INPUT",  
          "CodecSettings": {  
            "AacSettings": {  
              "RawFormat": "NONE",  
              "CodecProfile": "LC",  
              "AudioDescriptionBroadcasterMix": "NORMAL",  
              "SampleRate": 48000,  
              "Bitrate": 96000,  
              "RateControlMode": "CBR",  
              "Specification": "MPEG4",  
              "CodingMode": "CODING_MODE_2_0"  
            },  
            "Codec": "AAC"  
          }  
        }  
      ],  
      "VideoDescription": {  
        "RespondToAfd": "NONE",  
        "TimecodeInsertion": "DISABLED",
```

```
"Sharpness": 50,
"ColorMetadata": "INSERT",
"CodecSettings": {
  "H264Settings": {
    "FramerateControl": "INITIALIZE_FROM_SOURCE",
    "SpatialAdaptiveQuantization": "ENABLED",
    "Softness": 0,
    "Telecine": "NONE",
    "CodecLevel": "AUTO",
    "QualityTuningLevel": "SINGLE_PASS",
    "UnregisteredSeiTimecode": "DISABLED",
    "Slices": 1,
    "Syntax": "DEFAULT",
    "GopClosedCadence": 1,
    "AdaptiveQuantization": "HIGH",
    "EntropyEncoding": "CABAC",
    "InterlaceMode": "PROGRESSIVE",
    "ParControl": "INITIALIZE_FROM_SOURCE",
    "NumberBFramesBetweenReferenceFrames": 2,
    "GopSizeUnits": "FRAMES",
    "RepeatPps": "DISABLED",
    "CodecProfile": "MAIN",
    "FieldEncoding": "PAFF",
    "GopSize": 90.0,
    "SlowPal": "DISABLED",
    "SceneChangeDetect": "ENABLED",
    "GopBReference": "DISABLED",
    "RateControlMode": "CBR",
    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
    "FlickerAdaptiveQuantization": "DISABLED",
    "DynamicSubGop": "STATIC",
    "MinIInterval": 0,
    "TemporalAdaptiveQuantization": "ENABLED",
    "Bitrate": 400000,
    "NumberReferenceFrames": 3
  },
  "Codec": "H_264"
},
"AfdSignaling": "NONE",
"AntiAlias": "ENABLED",
"ScalingBehavior": "DEFAULT",
"DropFrameTimecode": "ENABLED"
},
},
```

```
    "Type": "CUSTOM",
    "CreatedAt": 1568841521
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS MediaConvert Elemental-Ausgabevoreinstellungen](#) im AWS Elemental-Benutzerhandbuch. MediaConvert

- API-Einheiten finden Sie [GetPreset](#) in AWS CLI der Befehlsreferenz.

get-queue

Das folgende Codebeispiel zeigt die Verwendung `get-queue`.

AWS CLI

Um Details für eine Warteschlange abzurufen

Im folgenden `get-queue` Beispiel werden die Details der angegebenen benutzerdefinierten Warteschlange abgerufen.

```
aws mediaconvert get-queue \
  --name Customer1 \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{
  "Queue": {
    "LastUpdated": 1526428502,
    "Type": "CUSTOM",
    "SubmittedJobsCount": 0,
    "Status": "ACTIVE",
    "PricingPlan": "ON_DEMAND",
    "CreatedAt": 1526428502,
    "ProgressingJobsCount": 0,
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
    "Name": "Customer1"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Queues](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [GetQueue](#) in AWS CLI der Befehlsreferenz.

list-job-templates

Das folgende Codebeispiel zeigt die Verwendung `list-job-templates`.

AWS CLI

Beispiel 1: Um Ihre benutzerdefinierten Jobvorlagen aufzulisten

Das folgende `list-job-templates` Beispiel listet alle benutzerdefinierten Jobvorlagen in der aktuellen Region auf. Eine Liste der System-Jobvorlagen finden Sie im nächsten Beispiel.

```
aws mediaconvert list-job-templates \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "JobTemplates": [  
    {  
      "Description": "Create a DASH streaming ABR stack",  
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH  
Streaming",  
      "Name": "DASH Streaming",  
      "LastUpdated": 1568653007,  
      "Priority": 0,  
      "Settings": {  
        ...<truncatedforbrevity>...  
      },  
      "Type": "CUSTOM",  
      "StatusUpdateInterval": "SECONDS_60",  
      "CreatedAt": 1568653007  
    },  
    {  
      "Description": "Create a high-res file",  
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/File",  
      "Name": "File",  
      "LastUpdated": 1568653007,  
      "Priority": 0,  
    }  
  ]  
}
```

```

    "Settings": {
      ...<truncatedforbrevity>...
    },
    "Type": "CUSTOM",
    "StatusUpdateInterval": "SECONDS_60",
    "CreatedAt": 1568653023
  }
]
}

```

Beispiel 2: Um die MediaConvert Systemjobvorlagen aufzulisten

Das folgende `list-job-templates` Beispiel listet alle Systemjobvorlagen auf.

```

aws mediaconvert list-job-templates \
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \
  --list-by SYSTEM

```

Ausgabe:

```

{
  "JobTemplates": [
    {
      "CreatedAt": 1568321779,
      "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:jobTemplates/System-
Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
      "Name": "System-Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
      "Description": "GENERIC, MP4, AVC + HEV1(HEVC,SDR), AAC, SDR, QVBR",
      "Category": "GENERIC",
      "Settings": {
        "AdAvailOffset": 0,
        "OutputGroups": [
          {
            "Outputs": [
              {
                "Extension": "mp4",
                "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5Mbps_Qvbr_Vq9",
                "NameModifier":
                "_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5000Kbps_Qvbr_Vq9"
              },
              {
                "Extension": "mp4",

```

```

        "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10Mbps_Qvbr_Vq9",
        "NameModifier":
        "_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10000Kbps_Qvbr_Vq9"
    },
    {
        "Extension": "mp4",
        "Preset": "System-
Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_0.8Mbps_Qvbr_Vq7",
        "NameModifier":
        "_Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_800Kbps_Qvbr_Vq7"
    },
    {
        "Extension": "mp4",
        "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4Mbps_Qvbr_Vq9",
        "NameModifier":
        "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4000Kbps_Qvbr_Vq9"
    },
    {
        "Extension": "mp4",
        "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8Mbps_Qvbr_Vq9",
        "NameModifier":
        "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8000Kbps_Qvbr_Vq9"
    },
    {
        "Extension": "mp4",
        "Preset": "System-
Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12Mbps_Qvbr_Vq9",
        "NameModifier":
        "_Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12000Kbps_Qvbr_Vq9"
    }
],
"OutputGroupSettings": {
    "FileGroupSettings": {

    },
    "Type": "FILE_GROUP_SETTINGS"
},
"Name": "File Group"
}
]
},

```



```
        "Type": "SYSTEM",
        "LastUpdated": 1568321779
    },
    ...<truncatedforbrevity>...
]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Job Templates](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [ListJobTemplates](#) in der AWS CLI Befehlsreferenz.

list-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-jobs`.

AWS CLI

Um Details für alle Jobs in einer Region abzurufen

Im folgenden Beispiel werden die Informationen für alle Ihre Jobs in der angegebenen Region abgefragt.

```
aws mediaconvert list-jobs \
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
  --region region-name-1
```

Um Ihren kontospezifischen Endpunkt abzurufen `describe-endpoints`, verwenden oder senden Sie den Befehl ohne den Endpunkt. Der Dienst gibt einen Fehler und Ihren Endpunkt zurück.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Jobs](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [ListJobs](#) in der AWS CLI Befehlsreferenz.

list-presets

Das folgende Codebeispiel zeigt die Verwendung `list-presets`.

AWS CLI

Beispiel 1: Um Ihre benutzerdefinierten Ausgabevoreinstellungen aufzulisten

Das folgende `list-presets` Beispiel listet Ihre benutzerdefinierten Ausgabevoreinstellungen auf. Eine Liste der Systemvoreinstellungen finden Sie im nächsten Beispiel.

```
aws mediaconvert list-presets \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "Presets": [  
    {  
      "Name": "SimpleMP4",  
      "CreatedAt": 1568841521,  
      "Settings": {  
        .....  
      },  
      "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",  
      "Type": "CUSTOM",  
      "LastUpdated": 1568843141,  
      "Description": "Creates basic MP4 file. No filtering or preprocessing."  
    },  
    {  
      "Name": "SimpleTS",  
      "CreatedAt": 1568843113,  
      "Settings": {  
        ... truncated for brevity ...  
      },  
      "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleTS",  
      "Type": "CUSTOM",  
      "LastUpdated": 1568843113,  
      "Description": "Create a basic transport stream."  
    }  
  ]  
}
```

Beispiel 2: Um die Systemausgabevoreinstellungen aufzulisten

Das folgende `list-presets` Beispiel listet die verfügbaren MediaConvert Systemvoreinstellungen auf. Eine Liste Ihrer benutzerdefinierten Voreinstellungen finden Sie im vorherigen Beispiel.

```
aws mediaconvert list-presets \  
  --list-by SYSTEM \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "Presets": [  
    {  
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/System-Avc_16x9_1080p_29_97fps_8500kbps",  
      "Name": "System-Avc_16x9_1080p_29_97fps_8500kbps",  
      "CreatedAt": 1568321789,  
      "Description": "Wifi, 1920x1080, 16:9, 29.97fps, 8500kbps",  
      "LastUpdated": 1568321789,  
      "Type": "SYSTEM",  
      "Category": "HLS",  
      "Settings": {  
        ...<output settings removed for brevity>...  
      }  
    },  
    ...<list of presets shortened for brevity>...  
    {  
      "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:presets/System-Xdcam_HD_1080i_29_97fps_35mpbs",  
      "Name": "System-Xdcam_HD_1080i_29_97fps_35mpbs",  
      "CreatedAt": 1568321790,  
      "Description": "XDCAM MPEG HD, 1920x1080i, 29.97fps, 35mbps",  
      "LastUpdated": 1568321790,  
      "Type": "SYSTEM",  
      "Category": "MXF",  
      "Settings": {  
        ...<output settings removed for brevity>...  
      }  
    }  
  ]  
}
```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS MediaConvert Elemental-Ausgabevoreinstellungen](#) im AWS MediaConvert Elemental-Benutzerhandbuch.

- API-Einheiten finden Sie [ListPresets](#) in AWS CLI der Befehlsreferenz.

list-queues

Das folgende Codebeispiel zeigt die Verwendung `list-queues`.

AWS CLI

Um Ihre Warteschlangen aufzulisten

Das folgende `list-queues` Beispiel listet alle Ihre MediaConvert Warteschlangen auf.

```
aws mediaconvert list-queues \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "Queues": [  
    {  
      "PricingPlan": "ON_DEMAND",  
      "Type": "SYSTEM",  
      "Status": "ACTIVE",  
      "CreatedAt": 1503451595,  
      "Name": "Default",  
      "SubmittedJobsCount": 0,  
      "ProgressingJobsCount": 0,  
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Default",  
      "LastUpdated": 1534549158  
    },  
    {  
      "PricingPlan": "ON_DEMAND",  
      "Type": "CUSTOM",  
      "Status": "ACTIVE",  
      "CreatedAt": 1537460025,  
      "Name": "Customer1",  
      "SubmittedJobsCount": 0,  
      "Description": "Jobs we run for our cusotmer."  
    }  
  ]  
}
```

```

    "ProgressingJobsCount": 0,
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
    "LastUpdated": 1537460025
  },
  {
    "ProgressingJobsCount": 0,
    "Status": "ACTIVE",
    "Name": "transcode-library",
    "SubmittedJobsCount": 0,
    "LastUpdated": 1564066204,
    "ReservationPlan": {
      "Status": "ACTIVE",
      "ReservedSlots": 1,
      "PurchasedAt": 1564066203,
      "Commitment": "ONE_YEAR",
      "ExpiresAt": 1595688603,
      "RenewalType": "EXPIRE"
    },
    "PricingPlan": "RESERVED",
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/transcode-
library",
    "Type": "CUSTOM",
    "CreatedAt": 1564066204
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Queues](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [ListQueues](#) in AWS CLI der Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags in einer MediaConvert Warteschlange, einer Jobvorlage oder einer Ausgabevoreinstellung aufzulisten

Im folgenden `list-tags-for-resource` Beispiel werden die Tags in der angegebenen Ausgabevoreinstellung aufgeführt.

```
aws mediaconvert list-tags-for-resource \  
  --arn arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "ResourceTags": {  
    "Tags": {  
      "customer": "zippyVideo"  
    },  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging AWS Elemental MediaConvert Queues, Job Templates und Output Presets](#) im AWS Elemental User Guide. MediaConvert

- API-Einheiten finden Sie in der Befehlsreferenz. [ListTagsForResource](#) AWS CLI

update-job-template

Das folgende Codebeispiel zeigt die Verwendung `update-job-template`.

AWS CLI

Um eine Jobvorlage zu ändern

Im folgenden `update-job-template` Beispiel wird die JSON-Definition der angegebenen benutzerdefinierten Auftragsvorlage durch die JSON-Definition in der bereitgestellten Datei ersetzt.

```
aws mediaconvert update-job-template --name File1 https://abcd1234.mediaconvert.us-west-2.amazonaws.com --endpoint-url -- file: ~/ .json cli-input-json job-template-update
```

Inhalt von `job-template-update.json`:

```
{  
  "Description": "A simple job template that generates a single file output.",  
  "Queue": "arn:aws:mediaconvert:us-east-1:012345678998:queues/Default",  
  "Name": "SimpleFile",  
  "Settings": {
```

```
"OutputGroups": [  
  {  
    "Name": "File Group",  
    "Outputs": [  
      {  
        "ContainerSettings": {  
          "Container": "MP4",  
          "Mp4Settings": {  
            "CslgAtom": "INCLUDE",  
            "FreeSpaceBox": "EXCLUDE",  
            "MoovPlacement": "PROGRESSIVE_DOWNLOAD"  
          }  
        },  
      },  
      "VideoDescription": {  
        "ScalingBehavior": "DEFAULT",  
        "TimecodeInsertion": "DISABLED",  
        "AntiAlias": "ENABLED",  
        "Sharpness": 50,  
        "CodecSettings": {  
          "Codec": "H_264",  
          "H264Settings": {  
            "InterlaceMode": "PROGRESSIVE",  
            "NumberReferenceFrames": 3,  
            "Syntax": "DEFAULT",  
            "Softness": 0,  
            "GopClosedCadence": 1,  
            "GopSize": 90,  
            "Slices": 1,  
            "GopBReference": "DISABLED",  
            "SlowPal": "DISABLED",  
            "SpatialAdaptiveQuantization": "ENABLED",  
            "TemporalAdaptiveQuantization": "ENABLED",  
            "FlickerAdaptiveQuantization": "DISABLED",  
            "EntropyEncoding": "CABAC",  
            "Bitrate": 400000,  
            "FramerateControl": "INITIALIZE_FROM_SOURCE",  
            "RateControlMode": "CBR",  
            "CodecProfile": "MAIN",  
            "Telecine": "NONE",  
            "MinIInterval": 0,  
            "AdaptiveQuantization": "HIGH",  
            "CodecLevel": "AUTO",  
            "FieldEncoding": "PAFF",  
            "SceneChangeDetect": "ENABLED",
```

```
    "QualityTuningLevel": "SINGLE_PASS",
    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
    "UnregisteredSeiTimecode": "DISABLED",
    "GopSizeUnits": "FRAMES",
    "ParControl": "INITIALIZE_FROM_SOURCE",
    "NumberBFramesBetweenReferenceFrames": 2,
    "RepeatPps": "DISABLED",
    "DynamicSubGop": "STATIC"
  }
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"AudioDescriptions": [
  {
    "AudioTypeControl": "FOLLOW_INPUT",
    "CodecSettings": {
      "Codec": "AAC",
      "AacSettings": {
        "AudioDescriptionBroadcasterMix": "NORMAL",
        "Bitrate": 96000,
        "RateControlMode": "CBR",
        "CodecProfile": "LC",
        "CodingMode": "CODING_MODE_2_0",
        "RawFormat": "NONE",
        "SampleRate": 48000,
        "Specification": "MPEG4"
      }
    }
  },
  {
    "LanguageCodeControl": "FOLLOW_INPUT"
  }
]
}
],
"OutputGroupSettings": {
  "Type": "FILE_GROUP_SETTINGS",
  "FileGroupSettings": {}
}
}
],
"AdAvailOffset": 0
},
```



```

    "StatusUpdateInterval": "SECONDS_60",
    "Priority": 0
  }

```

Das System gibt die JSON Payload zurück, die Sie mit Ihrer Anfrage gesendet haben, auch wenn die Anfrage zu einem Fehler führt. Daher handelt es sich bei der JSON zurückgegebenen Datei nicht unbedingt um die neue Definition der Jobvorlage.

Da die JSON Payload lang sein kann, müssen Sie möglicherweise nach oben scrollen, um Fehlermeldungen zu sehen.

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Job Templates](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [UpdateJobTemplate](#) in der AWS CLI Befehlsreferenz.

update-preset

Das folgende Codebeispiel zeigt die Verwendung `update-preset`.

AWS CLI

Um eine Voreinstellung zu ändern

Das folgende `update-preset` Beispiel ersetzt die Beschreibung für die angegebene Voreinstellung.

```

aws mediaconvert update-preset \
--name Customer1 \
--description "New description text."
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```

{
  "Preset": {
    "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",
    "Settings": {
      ...<output settings removed for brevity>...
    },
    "Type": "CUSTOM",
  }
}

```

```
"LastUpdated": 1568938411,  
"Description": "New description text.",  
"Name": "SimpleMP4",  
"CreatedAt": 1568938240  
}  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Output Presets](#) im AWS Elemental User Guide MediaConvert .

- API-Einheiten finden Sie [UpdatePreset](#) in AWS CLI der Befehlsreferenz.

update-queue

Das folgende Codebeispiel zeigt die Verwendung `update-queue`.

AWS CLI

Um eine Warteschlange zu ändern

Im folgenden `update-queue` Beispiel wird die angegebene Warteschlange angehalten, indem ihr Status geändert wird. `PAUSED`

```
aws mediaconvert update-queue \  
--name Customer1 \  
--status PAUSED \  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Ausgabe:

```
{  
  "Queue": {  
    "LastUpdated": 1568839845,  
    "Status": "PAUSED",  
    "ProgressingJobsCount": 0,  
    "CreatedAt": 1526428516,  
    "Arn": "arn:aws:mediaconvert:us-west-1:123456789012:queues/Customer1",  
    "Name": "Customer1",  
    "SubmittedJobsCount": 0,  
    "PricingPlan": "ON_DEMAND",  
    "Type": "CUSTOM"  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit AWS Elemental MediaConvert Queues](#) im AWS Elemental MediaConvert User Guide.

- API-Einheiten finden Sie [UpdateQueue](#) in AWS CLI der Befehlsreferenz.

MediaLive Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaLive.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-channel

Das folgende Codebeispiel zeigt die Verwendung `create-channel`.

AWS CLI

Um einen Kanal zu erstellen

Im folgenden `create-channel` Beispiel wird ein Kanal erstellt, indem eine JSON Datei übergeben wird, die die Parameter enthält, die Sie angeben möchten.

Der Kanal in diesem Beispiel nimmt eine HLS PULL Eingabe auf, die eine Verbindung zu einer Quelle herstellt, die Video, Audio und eingebettete Untertitel enthält. Der Kanal erstellt eine HLS Ausgabegruppe mit einem Akamai-Server als Ziel. Die Ausgabegruppe enthält zwei Ausgänge: einen für H.265-Video und AAC -Audio und einen für die VTT Webuntertitel, nur in englischer Sprache.

Der Kanal JSON für dieses Beispiel enthält die mindestens erforderlichen Parameter für einen Kanal, der eine HLS PULL Eingabe verwendet und eine HLS Ausgangsgruppe mit Akamai als Ziel erzeugt. Der JSON enthält die folgenden Hauptabschnitte:

InputAttachments, das eine Quelle für das Audio und eine Quelle für die Untertitel angibt. Es gibt keinen Videoauswahlschalter an, was bedeutet, dass das erste Video MediaLive extrahiert wird, das in der Quelle gefunden wird. **Destinations**, das die beiden IP-Adressen (URLs) für die einzelne Ausgangsgruppe in diesem Kanal enthält. Für diese Adressen sind Kennwörter erforderlich. **EncoderSettings**, das Unterabschnitte enthält. **AudioDescriptions**, was angibt, dass der Kanal ein Audioausgabeelement enthält, das die Quelle verwendet und Audio im AAC Format erzeugt. **InputAttachments CaptionDescriptions**, was angibt, dass der Kanal ein Untertitelausgabe-Asset enthält **InputAttachments**, das die Quelle verwendet und Untertitel im Webformat erzeugt. **VTT VideoDescriptions**, was angibt, dass der Kanal ein Videoausgabe-Asset mit der angegebenen Auflösung enthält. **OutputGroups**, das die Ausgangsgruppen spezifiziert. In diesem Beispiel gibt es eine Gruppe mit dem Namen `Akamai`. Die Verbindung wird hergestellt mit `HLSPUT`. Die Ausgangsgruppe enthält zwei Ausgänge. Eine Ausgabe ist für das Video-Asset (benannt `Video_high`) und das Audio-Asset (benannt `Audio_EN`) vorgesehen. Eine Ausgabe ist für das Untertitel-Asset (benannt `WebVTT_EN`).

In diesem Beispiel enthalten einige Parameter keinen Wert oder enthalten verschachtelte leere Parameter. Beispielsweise enthält die `Video_and_audio` Ausgabe mehrere verschachtelte Parameter, `OutputSettings` die mit einem leeren Parameter `m3U8Settings` enden. Dieser Parameter muss enthalten sein, aber Sie können eines, mehrere oder alle untergeordneten Elemente weglassen, was bedeutet, dass das untergeordnete Element seinen Standardwert annimmt oder den Wert `Null` hat.

Alle Parameter, die für diesen Beispielkanal gelten, aber nicht in dieser Datei angegeben sind, nehmen entweder den Standardwert an, werden auf `Null` gesetzt oder nehmen einen eindeutigen Wert an, der von MediaLive generiert wird.

```
aws medialive create-channel \
  --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

Inhalt von `channel-in-hls-out-hls-akamai.json`:

```
{
  "Name": "News_West",
  "RoleArn": "arn:aws:iam::111122223333:role/MediaLiveAccessRole",
  "InputAttachments": [
```

```
{
  "InputAttachmentName": "local_news",
  "InputId": "1234567",
  "InputSettings": {
    "AudioSelectors": [
      {
        "Name": "English-Audio",
        "SelectorSettings": {
          "AudioLanguageSelection": {
            "LanguageCode": "EN"
          }
        }
      }
    ],
    "CaptionSelectors": [
      {
        "LanguageCode": "ENE",
        "Name": "English_embedded"
      }
    ]
  }
},
"Destinations": [
  {
    "Id": "akamai-server-west",
    "Settings": [
      {
        "PasswordParam": "/medialive/examplecorp1",
        "Url": "http://203.0.113.55/news/news_west",
        "Username": "examplecorp"
      },
      {
        "PasswordParam": "/medialive/examplecorp2",
        "Url": "http://203.0.113.82/news/news_west",
        "Username": "examplecorp"
      }
    ]
  }
],
"EncoderSettings": {
  "AudioDescriptions": [
    {
      "AudioSelectorName": "English-Audio",
```

```
        "CodecSettings": {
            "AacSettings": {}
        },
        "Name": "Audio_EN"
    }
],
"CaptionDescriptions": [
    {
        "CaptionSelectorName": "English_embedded",
        "DestinationSettings": {
            "WebvttDestinationSettings": {}
        },
        "Name": "WebVTT_EN"
    }
],
"VideoDescriptions": [
    {
        "Height": 720,
        "Name": "Video_high",
        "Width": 1280
    }
],
"OutputGroups": [
    {
        "Name": "Akamai",
        "OutputGroupSettings": {
            "HlsGroupSettings": {
                "Destination": {
                    "DestinationRefId": "akamai-server-west"
                },
                "HlsCdnSettings": {
                    "HlsBasicPutSettings": {}
                }
            }
        }
    },
    {
        "Outputs": [
            {
                "AudioDescriptionNames": [
                    "Audio_EN"
                ],
                "OutputName": "Video_and_audio",
                "OutputSettings": {
                    "HlsOutputSettings": {
                        "HlsSettings": {
```

```

        "StandardHlsSettings": {
            "M3u8Settings": {}
        }
    },
    "NameModifier": "_1"
}
},
"VideoDescriptionName": "Video_high"
},
{
    "CaptionDescriptionNames": [
        "WebVTT_EN"
    ],
    "OutputName": "Captions-WebVTT",
    "OutputSettings": {
        "HlsOutputSettings": {
            "HlsSettings": {
                "StandardHlsSettings": {
                    "M3u8Settings": {}
                }
            },
            "NameModifier": "_2"
        }
    }
}
]
}
},
"TimecodeConfig": {
    "Source": "EMBEDDED"
}
}
}

```

Ausgabe:

Die Ausgabe wiederholt den Inhalt der JSON Datei plus die folgenden Werte. Alle Parameter sind alphabetisch sortiert.

ARN für den Kanal. Der letzte Teil von ARN ist die eindeutige Kanal-ID. EgressEndpoints ist in diesem Beispielkanal leer, da er nur für PUSH Eingaben verwendet wird. Wenn es zutrifft, werden die Adressen angezeigt MediaLive , an die der Inhalt gesendet wurde. OutputGroups, Outputs. Diese zeigen alle Parameter für die Ausgabegruppe und die Ausgaben, einschließlich der

Parameter, die Sie nicht aufgenommen haben, die aber für diesen Kanal relevant sind. Die Parameter sind möglicherweise leer (was möglicherweise darauf hindeutet, dass der Parameter oder die Funktion in dieser Kanalkonfiguration deaktiviert ist) oder sie zeigen möglicherweise den Standardwert an, der gilt. `LogLevel` ist auf die Standardeinstellung (DISABLED) gesetzt. `Tags` ist auf den Standard (null) gesetzt. `PipelinesRunningCount` und `State` zeigt den aktuellen Status des Kanals an.

Weitere Informationen finden Sie unter [Einen Kanal von Grund auf neu erstellen](#) im AWS MediaLive Elemental-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateChannel](#) in der AWS CLI Befehlsreferenz.

create-input

Das folgende Codebeispiel zeigt die Verwendung `create-input`.

AWS CLI

Um eine Eingabe zu erstellen

Im folgenden `create-input` Beispiel wird eine HLS PULL Eingabe erstellt, indem eine JSON Datei übergeben wird, die die Parameter enthält, die für diesen Typ von Eingabe gelten. Die Eingabe JSON für dieses Beispiel spezifiziert zwei Quellen (Adressen) für die Eingabe, um Redundanz bei der Aufnahme zu unterstützen. Für diese Adressen sind Passwörter erforderlich.

```
aws medialive create-input \  
  --cli-input-json file://input-hls-pull-news.json
```

Inhalt von `input-hls-pull-news.json`:

```
{  
  "Name": "local_news",  
  "RequestId": "cli000059",  
  "Sources": [  
    {  
      "Url": "https://203.0.113.13/newschannel/anytownusa.m3u8",  
      "Username": "examplecorp",  
      "PasswordParam": "/medialive/examplecorp1"  
    },  
    {
```



```
        "Url": "https://198.51.100.54/fillervideos/oceanwaves.mp4",
        "Username": "examplecorp",
        "PasswordParam": "examplecorp2"
    },
    "Type": "URL_PULL"
}
```

Ausgabe:

Die Ausgabe wiederholt den Inhalt der JSON Datei sowie die folgenden Werte. Alle Parameter sind alphabetisch sortiert.

Arn für die Eingabe. Der letzte Teil von ARN ist die eindeutige Eingabe-ID. Attached Channels, der bei einer neu erstellten Eingabe immer leer ist. Destinations, das in diesem Beispiel leer ist, weil es nur mit einer PUSH Eingabe verwendet wird. Id für die Eingabe entspricht der ID in der ARN. MediaConnectFlows, das in diesem Beispiel leer ist, weil es nur mit einer Eingabe vom Typ verwendet wird MediaConnect. SecurityGroups, das in diesem Beispiel leer ist, weil es nur mit einer PUSH Eingabe verwendet wird. Statedieser Eingabe. Tags, das leer ist (die Standardeinstellung für diesen Parameter).

Weitere Informationen finden Sie unter [Eingabe erstellen](#) im AWS Elemental MediaLive User Guide.

- API Einheiten finden Sie [CreateInput](#) in der AWS CLI Befehlsreferenz.

MediaPackage Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaPackage.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-channel

Das folgende Codebeispiel zeigt die Verwendung `create-channel`.

AWS CLI

Um einen Kanal zu erstellen

Der folgende `create-channel` Befehl erstellt einen Kanal, der `sportschannel` im aktuellen Konto benannt ist.

```
aws mediapackage create-channel --id sportschannel
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
        "Password": "generatedwebdavpassword1",
        "Url": "https://f31c86aed53b815a.mediapackage.us-west-2.amazonaws.com/in/v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
        "Username": "generatedwebdavusername1"
      },
      {
        "Id": "2daa32878af24803b24183727211b8ff",
        "Password": "generatedwebdavpassword2",
        "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-west-2.amazonaws.com/in/v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
        "Username": "generatedwebdavusername2"
      }
    ]
  },
  "Id": "sportschannel",
  "Tags": {
```

```
    "region": "west"
  }
}
```

Weitere Informationen finden Sie unter [Einen Kanal erstellen](#) im AWS MediaPackage Elemental-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateChannel](#) in der AWS CLI Befehlsreferenz.

create-origin-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-origin-endpoint`.

AWS CLI

Um einen Origin-Endpunkt zu erstellen

Der folgende `create-origin-endpoint` Befehl erstellt einen Ausgangsendpunkt, der `cmf sports` mit den in einer JSON-Datei bereitgestellten Paketeinstellungen und den angegebenen Endpunkteinstellungen benannt wird.

```
aws mediapackage create-origin-endpoint \
  --channel-id sportschannel \
  --id cmf sports \
  --cmf-package file://file/path/cmafpkg.json --description "cmf output of sports" \
  --id cmf_sports \
  --manifest-name sports_channel \
  --startover-window-seconds 300 \
  --tags region=west,media=sports \
  --time-delay-seconds 10
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
  "ChannelId": "sportschannel",
  "CmfPackage": {
    "HlsManifests": [
      {
        "AdMarkers": "PASSTHROUGH",
```

```

        "Id": "cmf_sports_endpoint",
        "IncludeIframeOnlyStream": true,
        "ManifestName": "index",
        "PlaylistType": "EVENT",
        "PlaylistWindowSeconds": 300,
        "ProgramDateTimeIntervalSeconds": 300,
        "Url": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmef_sports_endpoint/
index.m3u8"
    }
],
    "SegmentDurationSeconds": 2,
    "SegmentPrefix": "sportschannel"
},
    "Description": "cmef output of sports",
    "Id": "cmef_sports",
    "ManifestName": "sports_channel",
    "StartoverWindowSeconds": 300,
    "Tags": {
        "region": "west",
        "media": "sports"
    },
    "TimeDelaySeconds": 10,
    "Url": "",
    "Whitelist": []
}

```

Weitere Informationen finden Sie unter [Erstellen eines Endpunkts](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [CreateOriginEndpoint](#) in der AWS CLI Befehlsreferenz.

delete-channel

Das folgende Codebeispiel zeigt die Verwendung `delete-channel`.

AWS CLI

Um einen Kanal zu löschen

Der folgende `delete-channel` Befehl löscht den genannten `test` Kanal.

```
aws mediapackage delete-channel \
```

```
--id test
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Kanals](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DeleteChannel](#) in der AWS CLI Befehlsreferenz.

delete-origin-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-origin-endpoint`.

AWS CLI

Um einen Ursprungsendpoint zu löschen

Mit dem folgenden `delete-origin-endpoint` Befehl wird der angegebene Ausgangsendpoint gelöscht. `tester2`

```
aws mediapackage delete-origin-endpoint \  
  --id tester2
```

Weitere Informationen finden Sie unter [Löschen eines Endpunkts](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DeleteOriginEndpoint](#) in der AWS CLI Befehlsreferenz.

describe-channel

Das folgende Codebeispiel zeigt die Verwendung `describe-channel`.

AWS CLI

Um einen Kanal zu beschreiben

Der folgende `describe-channel` Befehl zeigt alle Details des genannten Kanals `antest`.

```
aws mediapackage describe-channel \  
  --id test
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "584797f1740548c389a273585dd22a63",
        "Password": "webdavgeneratedpassword1",
        "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
        "Username": "webdavgeneratedusername1"
      },
      {
        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
        "Password": "webdavgeneratedpassword2",
        "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
        "Username": "webdavgeneratedusername2"
      }
    ]
  },
  "Id": "test",
  "Tags": {}
}
```

Weitere Informationen finden Sie unter [Kanaldetails anzeigen< https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html>](https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html) im Elemental User Guide AWS MediaPackage

- Einzelheiten API finden Sie in der Befehlsreferenz. [DescribeChannel](#)AWS CLI

describe-origin-endpoint

Das folgende Codebeispiel zeigt die Verwendung `describe-origin-endpoint`.

AWS CLI

Um einen Ausgangsendpunkt zu beschreiben

Der folgende `describe-origin-endpoint` Befehl zeigt alle Details des genannten Ausgangsendpunkts `ancmaf_sports`.

```
aws mediapackage describe-origin-endpoint \  
  --id cmaf_sports
```

Ausgabe:

```
{  
  "Arn": "arn:aws:mediapackage:us-  
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "NONE",  
        "Id": "cmaf_sports_endpoint",  
        "IncludeIframeOnlyStream": false,  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 60,  
        "ProgramDateTimeIntervalSeconds": 0,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-  
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/  
index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  },  
  "Id": "cmaf_sports",  
  "ManifestName": "index",  
  "StartoverWindowSeconds": 0,  
  "Tags": {  
    "region": "west",  
    "media": "sports"  
  },  
  "TimeDelaySeconds": 0,  
  "Url": "",  
  "Whitelist": []  
}
```

Weitere Informationen finden Sie unter [Anzeigen eines einzelnen Endpunkts](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DescribeOriginEndpoint](#) in der AWS CLI Befehlsreferenz.

list-channels

Das folgende Codebeispiel zeigt die Verwendung `list-channels`.

AWS CLI

Um alle Kanäle aufzulisten

Der folgende `list-channels` Befehl listet alle Kanäle auf, die auf dem aktuellen AWS Konto konfiguriert sind.

```
aws mediapackage list-channels
```

Ausgabe:

```
{
  "Channels": [
    {
      "Arn": "arn:aws:mediapackage:us-west-2:111222333:channels/584797f1740548c389a273585dd22a63",
      "HlsIngest": {
        "IngestEndpoints": [
          {
            "Id": "584797f1740548c389a273585dd22a63",
            "Password": "webdavgeneratedpassword1",
            "Url": "https://9be9c4405c474882.mediapackage.us-west-2.amazonaws.com/in/v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
            "Username": "webdavgeneratedusername1"
          },
          {
            "Id": "7d187c8616fd455f88aaa5a9fcf74442",
            "Password": "webdavgeneratedpassword2",
            "Url": "https://7bf454c57220328d.mediapackage.us-west-2.amazonaws.com/in/v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
            "Username": "webdavgeneratedusername2"
          }
        ]
      },
      "Id": "test",
      "Tags": {}
    }
  ]
}
```



```
]
}
```

Weitere Informationen finden Sie unter [Kanaldetails anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [ListChannels](#) in der AWS CLI Befehlsreferenz.

list-origin-endpoints

Das folgende Codebeispiel zeigt die Verwendung `list-origin-endpoints`.

AWS CLI

Um alle Origin-Endpoints auf einem Kanal aufzulisten

Der folgende `list-origin-endpoints` Befehl listet alle Origin-Endpoints auf, die auf dem genannten Kanal konfiguriert sind. `test`

```
aws mediapackage list-origin-endpoints \
  --channel-id test
```

Ausgabe:

```
{
  "OriginEndpoints": [
    {
      "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/247cff871f2845d3805129be22f2c0a2",
      "ChannelId": "test",
      "DashPackage": {
        "ManifestLayout": "FULL",
        "ManifestWindowSeconds": 60,
        "MinBufferTimeSeconds": 30,
        "MinUpdatePeriodSeconds": 15,
        "PeriodTriggers": [],
        "Profile": "NONE",
        "SegmentDurationSeconds": 2,
        "SegmentTemplateFormat": "NUMBER_WITH_TIMELINE",
        "StreamSelection": {
          "MaxVideoBitsPerSecond": 2147483647,
          "MinVideoBitsPerSecond": 0,
          "StreamOrder": "ORIGINAL"
        }
      }
    }
  ]
}
```

```

    },
    "SuggestedPresentationDelaySeconds": 25
  },
  "Id": "tester2",
  "ManifestName": "index",
  "StartoverWindowSeconds": 0,
  "Tags": {},
  "TimeDelaySeconds": 0,
  "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/247cff871f2845d3805129be22f2c0a2/index.mpd",
  "Whitelist": []
},
{
  "Arn": "arn:aws:mediapackage:us-
west-2:111222333:origin_endpoints/869e237f851549e9bcf10e3bc2830839",
  "ChannelId": "test",
  "HlsPackage": {
    "AdMarkers": "NONE",
    "IncludeIframeOnlyStream": false,
    "PlaylistType": "EVENT",
    "PlaylistWindowSeconds": 60,
    "ProgramDateTimeIntervalSeconds": 0,
    "SegmentDurationSeconds": 6,
    "StreamSelection": {
      "MaxVideoBitsPerSecond": 2147483647,
      "MinVideoBitsPerSecond": 0,
      "StreamOrder": "ORIGINAL"
    },
    "UseAudioRenditionGroup": false
  },
  "Id": "tester",
  "ManifestName": "index",
  "StartoverWindowSeconds": 0,
  "Tags": {},
  "TimeDelaySeconds": 0,
  "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/869e237f851549e9bcf10e3bc2830839/index.m3u8",
  "Whitelist": []
}
]
}

```

Weitere Informationen finden Sie im AWS Elemental MediaPackage User Guide unter [Alle Endpoints anzeigen, die einem Kanal zugeordnet](#) sind.

- API-Einheiten finden Sie [ListOriginEndpoints](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die einer Ressource zugewiesenen Tags aufzulisten

Der folgende `list-tags-for-resource` Befehl listet die Tags auf, die der angegebenen Ressource zugewiesen sind.

```
aws mediapackage list-tags-for-resource \  
  --resource-arn arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0
```

Ausgabe:

```
{  
  "Tags": {  
    "region": "west"  
  }  
}
```

Weitere Informationen finden Sie unter [Tagging Resources in AWS Elemental MediaPackage](#) im AWS Elemental User Guide MediaPackage .

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

rotate-ingest-endpoint-credentials

Das folgende Codebeispiel zeigt die Verwendung `rotate-ingest-endpoint-credentials`.

AWS CLI

Um die Ingest-Anmeldeinformationen zu rotieren

Mit dem folgenden `rotate-ingest-endpoint-credentials` Befehl werden der DAV Web-Benutzername und das Kennwort für den angegebenen Ingest-Endpoint rotiert.

```
aws mediapackage rotate-ingest-endpoint-credentials \
  --id test \
  --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:channels/584797f1740548c389a273585dd22a63",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "584797f1740548c389a273585dd22a63",
        "Password": "webdavregeneratedpassword1",
        "Url": "https://9be9c4405c474882.mediapackage.us-west-2.amazonaws.com/in/v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
        "Username": "webdavregeneratedusername1"
      },
      {
        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
        "Password": "webdavgeneratedpassword2",
        "Url": "https://7bf454c57220328d.mediapackage.us-west-2.amazonaws.com/in/v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
        "Username": "webdavgeneratedusername2"
      }
    ]
  },
  "Id": "test",
  "Tags": {}
}
```

Weitere Informationen finden Sie unter [Rotation von Anmeldeinformationen auf einer Eingabe URL](#) im AWS Elemental User Guide MediaPackage .

- API-Einheiten finden Sie unter [RotatIngestEndpointCredentials AWS CLI Befehlsreferenz](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einer Ressource einen Tag hinzu

Die folgenden `tag-resource` Befehle fügen der angegebenen Ressource ein `region=west` Schlüssel- und Wertepaar hinzu.

```
aws mediapackage tag-resource \  
  --resource-arn arn:aws:mediapackage:us-  
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \  
  --tags region=west
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Resources in AWS Elemental MediaPackage](#) im AWS Elemental User Guide MediaPackage .

- API-Einheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Der folgende `untag-resource` Befehl entfernt das Tag mit dem Schlüssel `region` aus dem angegebenen Kanal.

```
aws mediapackage untag-resource \  
  --resource-arn arn:aws:mediapackage:us-  
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \  
  --tag-keys region
```

Weitere Informationen finden Sie unter [Tagging Resources in AWS Elemental MediaPackage](#) im AWS Elemental User Guide MediaPackage .

- API-Einheiten finden Sie [UntagResource](#) in AWS CLI der Befehlsreferenz.

update-channel

Das folgende Codebeispiel zeigt die Verwendung `update-channel`.

AWS CLI

Um einen Kanal zu aktualisieren

Mit dem folgenden `update-channel` Befehl wird der angegebene Kanal so aktualisiert, dass er die Beschreibung `24x7 sports` enthält.

```
aws mediapackage update-channel \
  --id sportschannel \
  --description "24x7 sports"
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
  "Description": "24x7 sports",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
        "Password": "generatedwebdavpassword1",
        "Url": "https://f31c86aed53b815a.mediapackage.us-west-2.amazonaws.com/in/v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
        "Username": "generatedwebdavusername1"
      },
      {
        "Id": "2daa32878af24803b24183727211b8ff",
        "Password": "generatedwebdavpassword2",
        "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-west-2.amazonaws.com/in/v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
        "Username": "generatedwebdavusername2"
      }
    ]
  },
  "Id": "sportschannel",
  "Tags": {}
}
```

```
}
```

Weitere Informationen finden Sie unter [Bearbeiten eines Kanals](#) im AWS MediaPackage Elemental-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateChannel](#) in der AWS CLI Befehlsreferenz.

update-origin-endpoint

Das folgende Codebeispiel zeigt die Verwendung `update-origin-endpoint`.

AWS CLI

Um einen Origin-Endpunkt zu aktualisieren

Mit dem folgenden `update-origin-endpoint` Befehl wird der angegebene Ursprungsendpunkt aktualisiert `cmaf_sports`. Er ändert die Zeitverzögerung auf 0 Sekunden.

```
aws mediapackage update-origin-endpoint \  
  --id cmaf_sports \  
  --time-delay-seconds 0
```

Ausgabe:

```
{  
  "Arn": "arn:aws:mediapackage:us-  
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "NONE",  
        "Id": "cmaf_sports_endpoint",  
        "IncludeIframeOnlyStream": false,  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 60,  
        "ProgramDateTimeIntervalSeconds": 0,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-  
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/  
index.m3u8"  
      }  
    ],  
  },  
}
```

```
    "SegmentDurationSeconds": 2,
    "SegmentPrefix": "sportschannel"
  },
  "Id": "cmf_sports",
  "ManifestName": "index",
  "StartoverWindowSeconds": 0,
  "Tags": {
    "region": "west",
    "media": "sports"
  },
  "TimeDelaySeconds": 0,
  "Url": "",
  "Whitelist": []
}
```

Weitere Informationen finden Sie unter [Bearbeiten eines Endpunkts](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [UpdateOriginEndpoint](#) in der AWS CLI Befehlsreferenz.

MediaPackage VOD-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaPackage VOD.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-asset

Das folgende Codebeispiel zeigt die Verwendung `create-asset`.

AWS CLI

Um ein Asset zu erstellen

Im folgenden `create-asset` Beispiel wird ein Vermögenswert mit dem Namen des `Chicken_Asset` AWS Girokontos erstellt. Das Asset nimmt die Datei `30sec_chicken.smil` auf `MediaPackage`.

```
aws mediapackage-vod create-asset \
  --id chicken_asset \
  --packaging-group-id hls_chicken_gp \
  --source-role-arn arn:aws:iam::111122223333:role/EMP_Vod \
  --source-arn arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:assets/chicken_asset",
  "Id": "chicken_asset",
  "PackagingGroupId": "hls_chicken_gp",
  "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
  "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod",
  "EgressEndpoints": [
    {
      "PackagingConfigurationId": "New_config_1",
      "Url": "https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/
v1/6644b55df1744261ab3732a8e5cdaf07/904b06a58c7645e08d57d40d064216ac/
f5b2e633ff4942228095d164c10074f3/index.m3u8"
    },
    {
      "PackagingConfigurationId": "new_hls",
      "Url": "https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/6644b55df1744261ab3732a8e5cdaf07/
fe8f1f00a80e424cb4f8da4095835e9e/7370ec57432343af816332356d2bd5c6/string.m3u8"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Ein Asset aufnehmen](#) im AWS Elemental-Benutzerhandbuch. MediaPackage

- API-Einheiten finden Sie [CreateAsset](#) in der AWS CLI Befehlsreferenz.

create-packaging-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-packaging-configuration`.

AWS CLI

Um eine Verpackungskonfiguration zu erstellen

Im folgenden `create-packaging-configuration` Beispiel wird eine Verpackungskonfiguration erstellt, die `new_hls` in der angegebenen Verpackungsgruppe benannt ist `hls_chicken`. In diesem Beispiel wird eine Datei auf der Festplatte mit dem Namen `hls_pc.json`, um die Details bereitzustellen.

```
aws mediapackage-vod create-packaging-configuration \
  --id new_hls \
  --packaging-group-id hls_chicken \
  --hls-package file://hls_pc.json
```

Inhalt von `hls_pc.json`:

```
{
  "HlsManifests":[
    {
      "AdMarkers":"NONE",
      "IncludeIframeOnlyStream":false,
      "ManifestName":"string",
      "ProgramDateTimeIntervalSeconds":60,
      "RepeatExtXKey":true,
      "StreamSelection":{
        "MaxVideoBitsPerSecond":1000,
        "MinVideoBitsPerSecond":0,
        "StreamOrder":"ORIGINAL"
      }
    }
  ],
  "SegmentDurationSeconds":6,
  "UseAudioRenditionGroup":false
}
```

Ausgabe:

```
{
```

```

    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/
new_hls",
    "Id": "new_hls",
    "PackagingGroupId": "hls_chicken",
    "HlsManifests": {
      "SegmentDurationSeconds": 6,
      "UseAudioRenditionGroup": false,
      "HlsMarkers": [
        {
          "AdMarkers": "NONE",
          "IncludeIframeOnlyStream": false,
          "ManifestName": "string",
          "ProgramDateTimeIntervalSeconds": 60,
          "RepeatExtXKey": true,
          "StreamSelection": {
            "MaxVideoBitsPerSecond": 1000,
            "MinVideoBitsPerSecond": 0,
            "StreamOrder": "ORIGINAL"
          }
        }
      ]
    }
  }
}

```

Weitere Informationen finden Sie unter [Creating a Packaging Configuration](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [CreatePackagingConfiguration](#) in der AWS CLI Befehlsreferenz.

create-packaging-group

Das folgende Codebeispiel zeigt die Verwendung `create-packaging-group`.

AWS CLI

Um eine Verpackungsgruppe zu erstellen

Im folgenden `create-packaging-group` Beispiel werden alle Verpackungsgruppen aufgeführt, die im aktuellen AWS Konto konfiguriert sind.

```

aws mediapackage-vod create-packaging-group \
  --id hls_chicken

```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
hls_chicken",
  "Id": "hls_chicken"
}
```

Weitere Informationen finden Sie im AWS Elemental MediaPackage User Guide unter [Creating a Packaging Group](#).

- API-Einheiten finden Sie [CreatePackagingGroup](#) in der AWS CLI Befehlsreferenz.

delete-asset

Das folgende Codebeispiel zeigt die Verwendung `delete-asset`.

AWS CLI

Um ein Asset zu löschen

Im folgenden `delete-asset` Beispiel wird das Objekt mit dem Namen `30sec_chicken` gelöscht.

```
aws mediapackage-vod delete-asset \
  --id 30sec_chicken
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Assets](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DeleteAsset](#) in der AWS CLI Befehlsreferenz.

delete-packaging-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-packaging-configuration`.

AWS CLI

Um eine Verpackungskonfiguration zu löschen

Im folgenden `delete-packaging-configuration` Beispiel wird die angegebene Verpackungskonfiguration gelöscht. `CMAF`

```
aws mediapackage-vod delete-packaging-configuration \  
  --id CMAF
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Verpackungskonfiguration](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DeletePackagingConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-packaging-group

Das folgende Codebeispiel zeigt die Verwendung `delete-packaging-group`.

AWS CLI

Um eine Verpackungsgruppe zu löschen

Im folgenden `delete-packaging-group` Beispiel wird die angegebene Verpackungsgruppe gelöscht. `Dash_widevine`

```
aws mediapackage-vod delete-packaging-group \  
  --id Dash_widevine
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Verpackungsgruppe](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DeletePackagingGroup](#) in der AWS CLI Befehlsreferenz.

describe-asset

Das folgende Codebeispiel zeigt die Verwendung `describe-asset`.

AWS CLI

Um ein Asset zu beschreiben

Im folgenden `describe-asset` Beispiel werden alle Details des genannten Assets angezeigt `30sec_chicken`.

```
aws mediapackage-vod describe-asset \
  --id 30sec_chicken
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:assets/30sec_chicken",
  "Id": "30sec_chicken",
  "PackagingGroupId": "Packaging_group_1",
  "SourceArn": "arn:aws:s3:111122223333:video-bucket/A/30sec_chicken.smil",
  "SourceRoleArn": "arn:aws:iam:111122223333:role/EMP_Vod",
  "EgressEndpoints": [
    {
      "PackagingConfigurationId": "DASH",
      "Url": "https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/66c25aff456d463aae0855172b3beb27/4ddfda6da17c4c279a1b8401cb
index.mpd"
    },
    {
      "PackagingConfigurationId": "HLS",
      "Url": "https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/6e5bf286a3414254a2bf0d22ae148d7e/06b5875b4d004c3cbdc4da2dc4
index.m3u8"
    },
    {
      "PackagingConfigurationId": "CMAF",
      "Url": "https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/628fb5d8d89e4702958b020af27fde0e/05eb062214064238ad6330a443
index.m3u8"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Asset-Details anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DescribeAsset](#) in der AWS CLI Befehlsreferenz.

describe-packaging-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-packaging-configuration`.

AWS CLI

Um eine Verpackungskonfiguration zu beschreiben

Im folgenden `describe-packaging-configuration` Beispiel werden alle Details der genannten Verpackungskonfiguration angezeigtDASH.

```
aws mediapackage-vod describe-packaging-configuration \  
  --id DASH
```

Ausgabe:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/  
DASH",  
  "Id": "DASH",  
  "PackagingGroupId": "Packaging_group_1",  
  "DashPackage": [  
    {  
      "SegmentDurationSeconds": "2"  
    },  
    {  
      "DashManifests": {  
        "ManifestName": "index",  
        "MinBufferTimeSeconds": "30",  
        "Profile": "NONE"  
      }  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verpackungskonfigurationsdetails anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API Einheiten finden Sie [DescribePackagingConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-packaging-group

Das folgende Codebeispiel zeigt die Verwendung `describe-packaging-group`.

AWS CLI

Um eine Verpackungsgruppe zu beschreiben

Im folgenden `describe-packaging-group` Beispiel werden alle Details der genannten Verpackungsgruppe angezeigt `Packaging_group_1`.

```
aws mediapackage-vod describe-packaging-group \
  --id Packaging_group_1
```

Ausgabe:

```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
Packaging_group_1",
  "Id": "Packaging_group_1"
}
```

Weitere Informationen finden Sie unter [Details zur Verpackungsgruppe anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [DescribePackagingGroup](#) in der AWS CLI Befehlsreferenz.

list-assets

Das folgende Codebeispiel zeigt die Verwendung `list-assets`.

AWS CLI

Um alle Vermögenswerte aufzulisten

Das folgende `list-assets` Beispiel listet alle Vermögenswerte auf, die im AWS Girokonto konfiguriert sind.

```
aws mediapackage-vod list-assets
```

Ausgabe:

```
{
  "Assets": [
    "Arn": "arn:aws:mediapackage-vod:us-
west-2:111122223333:assets/30sec_chicken",
```



```

    "Id": "30sec_chicken",
    "PackagingGroupId": "Packaging_group_1",
    "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
    "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod"
  ]
}

```

Weitere Informationen finden Sie unter [Asset-Details anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [ListAssets](#) in der AWS CLI Befehlsreferenz.

list-packaging-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-packaging-configurations`.

AWS CLI

Um alle Verpackungskonfigurationen aufzulisten

Im folgenden `list-packaging-configurations` Beispiel werden alle Verpackungskonfigurationen aufgeführt, die für die angegebene Verpackungsgruppe konfiguriert sind `Packaging_group_1`.

```

aws mediapackage-vod list-packaging-configurations \
  --packaging-group-id Packaging_group_1

```

Ausgabe:

```

{
  "PackagingConfigurations": [
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/CMAF",
      "Id": "CMAF",
      "PackagingGroupId": "Packaging_group_1",
      "CmafPackage": [
        {
          "SegmentDurationSeconds": "2"
        },
        {
          "HlsManifests": {
            "AdMarkers": "NONE",

```

```

        "RepeatExtXKey":"False",
        "ManifestName":"index",
        "ProgramDateTimeIntervalSeconds":"0",
        "IncludeIframeOnlyStream":"False"
    }
}
],
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/DASH",
    "Id":"DASH",
    "PackagingGroupId":"Packaging_group_1",
    "DashPackage":[
        {
            "SegmentDurationSeconds":"2"
        },
        {
            "DashManifests":{
                "ManifestName":"index",
                "MinBufferTimeSeconds":"30",
                "Profile":"NONE"
            }
        }
    ]
},
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/HLS",
    "Id":"HLS",
    "PackagingGroupId":"Packaging_group_1",
    "HlsPackage":[
        {
            "SegmentDurationSeconds":"6",
            "UseAudioRenditionGroup":"False"
        },
        {
            "HlsManifests":{
                "AdMarkers":"NONE",
                "RepeatExtXKey":"False",
                "ManifestName":"index",
                "ProgramDateTimeIntervalSeconds":"0",
                "IncludeIframeOnlyStream":"False"
            }
        }
    ]
}

```

```

    }
  ]
},
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/New_config_0_copy",
  "Id": "New_config_0_copy",
  "PackagingGroupId": "Packaging_group_1",
  "HlsPackage": [
    {
      "SegmentDurationSeconds": "6",
      "UseAudioRenditionGroup": "False"
    },
    {
      "Encryption": {
        "EncryptionMethod": "AWS_128",
        "SpekeKeyProvider": {
          "RoleArn": "arn:aws:iam:111122223333::role/SPEKERole",
          "Url": "https://1fgubdvs97.execute-api.us-
west-2.amazonaws.com/EkeStage/copyProtection/",
          "SystemIds": [
            "81376844-f976-481e-a84e-cc25d39b0b33"
          ]
        }
      }
    },
    {
      "HlsManifests": {
        "AdMarkers": "NONE",
        "RepeatExtXKey": "False",
        "ManifestName": "index",
        "ProgramDateTimeIntervalSeconds": "0",
        "IncludeIframeOnlyStream": "False"
      }
    }
  ]
}
]
}

```

Weitere Informationen finden Sie unter [Verpackungskonfigurationsdetails anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [ListPackagingConfigurations](#) in der AWS CLI Befehlsreferenz.

list-packaging-groups

Das folgende Codebeispiel zeigt die Verwendung `list-packaging-groups`.

AWS CLI

Um alle Verpackungsgruppen aufzulisten

Im folgenden `list-packaging-groups` Beispiel werden alle Verpackungsgruppen aufgeführt, die im aktuellen AWS Konto konfiguriert sind.

```
aws mediapackage-vod list-packaging-groups
```

Ausgabe:

```
{
  "PackagingGroups": [
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Dash_widevine",
      "Id": "Dash_widevine"
    },
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Encrypted_HLS",
      "Id": "Encrypted_HLS"
    },
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
groups/Packaging_group_1",
      "Id": "Packaging_group_1"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Paketgruppendetails anzeigen](#) im AWS Elemental MediaPackage User Guide.

- API-Einheiten finden Sie [ListPackagingGroups](#) in der AWS CLI Befehlsreferenz.

MediaStore Beispiele für Datenebene mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with MediaStore Data Plane Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-object

Das folgende Codebeispiel zeigt die Verwendung `delete-object`.

AWS CLI

Um ein Objekt zu löschen

Im folgenden `delete-object` Beispiel wird das angegebene Objekt gelöscht.

```
aws mediastore-data delete-object \  
  --endpoint=https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path=/folder_name/README.md
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines Objekts](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [DeleteObject](#) in der AWS CLI Befehlsreferenz.

describe-object

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-object`.

AWS CLI

Um die Header für ein Objekt anzuzeigen

Im folgenden `describe-object` Beispiel werden die Header für ein Objekt im angegebenen Pfad angezeigt.

```
aws mediastore-data describe-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg
```

Ausgabe:

```
{  
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",  
  "ContentType": "image/jpeg",  
  "ContentLength": "3860266",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"  
}
```

Weitere Informationen finden Sie unter [Anzeigen der Details eines Objekts](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [DescribeObject](#) in der AWS CLI Befehlsreferenz.

get-object

Das folgende Codebeispiel zeigt, wie man es benutzt `get-object`.

AWS CLI

Beispiel 1: Um ein ganzes Objekt herunterzuladen

Im folgenden `get-object` Beispiel wird das angegebene Objekt heruntergeladen.

```
aws mediastore-data get-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg setup.jpg
```

Ausgabe:

```
{
  "ContentType": "image/jpeg",
  "StatusCode": 200,
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",
  "ContentLength": "3860266",
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"
}
```

Beispiel 2: Um einen Teil eines Objekts herunterzuladen

Im folgenden `get-object` Beispiel wird der angegebene Teil eines Objekts heruntergeladen.

```
aws mediastore-data get-object \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \
  --path events/baseball/setup.jpg setup.jpg \
  --range "bytes=0-100"
```

Ausgabe:

```
{
  "StatusCode": 206,
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
  "ContentType": "image/jpeg",
  "ContentRange": "bytes 0-100/3860266",
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",
  "ContentLength": "101"
}
```

Weitere Informationen finden Sie unter [Objekt herunterladen](#) im AWS MediaStore Elemental-Benutzerhandbuch.

- API-Einheiten finden Sie [GetObject](#) in der AWS CLI Befehlsreferenz.

list-items

Das folgende Codebeispiel zeigt, wie man es benutzt `list-items`.

AWS CLI

Beispiel 1: Um eine Liste von Elementen (Objekten und Ordnern) anzuzeigen, die in einem Container gespeichert sind

Im folgenden `list-items` Beispiel wird eine Liste von Elementen (Objekten und Ordnern) angezeigt, die im angegebenen Container gespeichert sind.

```
aws mediastore-data list-items \
  --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

Ausgabe:

```
{
  "Items": [
    {
      "Type": "OBJECT",
      "ContentLength": 3784,
      "Name": "setup.jpg",
      "ETag":
"2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379
    },
    {
      "Type": "FOLDER",
      "Name": "events"
    }
  ]
}
```

Beispiel 2: Um eine Liste von Elementen (Objekten und Ordnern) anzuzeigen, die in einem Ordner gespeichert sind

Im folgenden `list-items` Beispiel wird eine Liste von Elementen (Objekten und Ordnern) angezeigt, die im angegebenen Ordner gespeichert sind.

```
aws mediastore-data list-items \
  --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
  --path events/baseball
```

Ausgabe:

```
{
  "Items": [
    {
```


Beispiel 2: Um ein Objekt in einen Ordner innerhalb eines Containers hochzuladen

Im folgenden `put-object` Beispiel wird ein Objekt in den angegebenen Ordner innerhalb eines Containers hochgeladen.

```
aws mediastore-data put-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --body ReadMe.md \  
  --path /september-events/ReadMe.md \  
  --cache-control "max-age=6, public" \  
  --content-type binary/octet-stream
```

Ausgabe:

```
{  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",  
  "ContentSHA256":  
  "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",  
  "StorageClass": "TEMPORAL"  
}
```

Weitere Informationen finden Sie unter [Hochladen eines Objekts](#) im AWS Elemental MediaStore User Guide.

- API-Einheiten finden Sie [PutObject](#) in der AWS CLI Befehlsreferenz.

MediaTailor Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren MediaTailor.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-playback-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-playback-configuration`.

AWS CLI

So löschen Sie eine Konfiguration

Im Folgenden wird eine Konfiguration mit dem Namen `delete-playback-configuration` `campaign_short` gelöscht.

```
aws mediatailor delete-playback-configuration \  
  --name campaign_short
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Konfiguration](#) im AWS Elemental MediaTailor User Guide.

- API-Einheiten finden Sie [DeletePlaybackConfiguration](#) in der AWS CLI Befehlsreferenz.

get-playback-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-playback-configuration`.

AWS CLI

Um eine Konfiguration zu beschreiben

Im Folgenden `get-playback-configuration` werden alle Details der genannten Konfiguration angezeigt `west_campaign`.

```
aws mediatailor get-playback-configuration \  
  --name west_campaign
```

Ausgabe:

```
{  
  "AdDecisionServerUrl": "http://your.ads.url",  
  "CdnConfiguration": {},
```

```

    "DashConfiguration": {
      "ManifestEndpointPrefix":
        "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
      "MpdLocation": "EMT_DEFAULT",
      "OriginManifestType": "MULTI_PERIOD"
    },
    "HlsConfiguration": {
      "ManifestEndpointPrefix":
        "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
    },
    "Name": "west_campaign",
    "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/west_campaign",
    "PlaybackEndpointPrefix":
      "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
    "SessionInitializationEndpointPrefix":
      "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
    "Tags": {},
    "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-
west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
  }

```

Weitere Informationen finden Sie unter [Konfiguration anzeigen](#) im AWS Elemental MediaTailor User Guide.

- API-Einheiten finden Sie [GetPlaybackConfiguration](#) in der AWS CLI Befehlsreferenz.

list-playback-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-playback-configurations`.

AWS CLI

Um alle Konfigurationen aufzulisten

Im Folgenden `list-playback-configurations` werden alle Details der Konfiguration für das aktuelle AWS Konto angezeigt.

```
aws mediatailor list-playback-configurations
```

Ausgabe:

```
{
  "Items": [
    {
      "AdDecisionServerUrl": "http://your.ads.url",
      "CdnConfiguration": {},
      "DashConfiguration": {
        "ManifestEndpointPrefix":
          "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
          dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
        "MpdLocation": "EMT_DEFAULT",
        "OriginManifestType": "MULTI_PERIOD"
      },
      "HlsConfiguration": {
        "ManifestEndpointPrefix":
          "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
          master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
      },
      "Name": "west_campaign",
      "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
      west-2:123456789012:playbackConfiguration/west_campaign",
      "PlaybackEndpointPrefix":
        "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
      "SessionInitializationEndpointPrefix":
        "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/
        session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
      "Tags": {},
      "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-
      west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
    },
    {
      "AdDecisionServerUrl": "http://your.ads.url",
      "CdnConfiguration": {},
      "DashConfiguration": {
        "ManifestEndpointPrefix":
          "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
          dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
        "MpdLocation": "DISABLED",
        "OriginManifestType": "MULTI_PERIOD"
      },
      "HlsConfiguration": {
```

```

    "ManifestEndpointPrefix":
      "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
      master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/"
    },
    "Name": "sports_campaign",
    "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
    west-2:123456789012:playbackConfiguration/sports_campaign",
    "PlaybackEndpointPrefix":
      "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com",
    "SessionInitializationEndpointPrefix":
      "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediatailor.us-west-2.amazonaws.com/v1/
      session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
    "SlateAdUrl": "http://s3.bucket/slate_ad.mp4",
    "Tags": {},
    "VideoContentSourceUrl": "https://c4af3793bf76b33c.mediapackage.us-
    west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/sports_endpoint/
    index.m3u8"
  }
]
}

```

Weitere Informationen finden Sie unter Konfiguration anzeigen < <https://docs.aws.amazon.com/mediatailor/latest/ug/configurations-view.html> > im Elemental User Guide. AWS MediaTailor

- Einzelheiten API finden Sie in der Befehlsreferenz. [ListPlaybackConfigurations](#) AWS CLI

put-playback-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-playback-configuration`.

AWS CLI

Um eine Konfiguration zu erstellen

Im Folgenden `put-playback-configuration` wird eine Konfiguration mit dem Namen `campaign_short` erstellt.

```

aws mediatailor put-playback-configuration \
  --name campaign_short \
  --ad-decision-server-url http://your.ads.url \
  --video-content-source-url http://video.bucket/index.m3u8

```

Ausgabe:

```
{
  "AdDecisionServerUrl": "http://your.ads.url",
  "CdnConfiguration": {},
  "DashConfiguration": {
    "ManifestEndpointPrefix":
    "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",
    "MpdLocation": "EMT_DEFAULT",
    "OriginManifestType": "MULTI_PERIOD"
  },
  "HlsConfiguration": {
    "ManifestEndpointPrefix":
    "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/"
  },
  "Name": "campaign_short",
  "PlaybackConfigurationArn": "arn:aws:mediatailor:us-
west-2:123456789012:playbackConfiguration/campaign_short",
  "PlaybackEndpointPrefix":
  "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com",
  "SessionInitializationEndpointPrefix":
  "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",
  "Tags": {},
  "VideoContentSourceUrl": "http://video.bucket/index.m3u8"
}
```

Weitere Informationen finden Sie unter [Creating a Configuration](#) im AWS Elemental MediaTailor User Guide.

- API-Einheiten finden Sie [PutPlaybackConfiguration](#) in der AWS CLI Befehlsreferenz.

MemoryDB-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit MemoryDB Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

copy-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-snapshot`.

AWS CLI

Um einen Snapshot zu kopieren

Im folgenden `copy-snapshot` Beispiel wird eine Kopie eines Snapshots erstellt.

```
aws memorydb copy-snapshot \  
  --source-snapshot-name my-cluster-snapshot \  
  --target-snapshot-name my-cluster-snapshot-copy
```

Output

```
{  
  "Snapshot": {  
    "Name": "my-cluster-snapshot-copy",  
    "Status": "creating",  
    "Source": "manual",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:snapshot/my-cluster-snapshot-copy",  
    "ClusterConfiguration": {  
      "Name": "my-cluster",  
      "Description": " ",  
      "NodeType": "db.r6g.large",  
      "EngineVersion": "6.2",  
      "MaintenanceWindow": "wed:03:00-wed:04:00",  
      "Port": 6379,  
      "ParameterGroupName": "default.memorydb-redis6",  
      "SubnetGroupName": "my-sg",  
      "VpcId": "vpc-xx2574fc",  
      "SnapshotRetentionLimit": 0,  
    }  
  }  
}
```



```
        "SnapshotWindow": "04:30-05:30",
        "NumShards": 2
    }
}
```

Weitere Informationen finden Sie unter [Kopieren eines Snapshots](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [CopySnapshot](#) in der AWS CLI Befehlsreferenz.

create-acl

Das folgende Codebeispiel zeigt die Verwendung `create-acl`.

AWS CLI

Um ein zu erstellen ACL

Im folgenden `create-acl` Beispiel wird eine neue Zugriffskontrollliste erstellt.

```
aws memorydb create-acl \
  --acl-name "new-acl-1" \
  --user-names "my-user"
```

Ausgabe:

```
{
  "ACL": {
    "Name": "new-acl-1",
    "Status": "creating",
    "UserNames": [
      "my-user"
    ],
    "MinimumEngineVersion": "6.2",
    "Clusters": [],
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"
  }
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateAcl](#).AWS CLI

create-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-cluster`.

AWS CLI

Um einen Cluster zu erstellen

Das folgende `create-cluster` Beispiel erstellt einen neuen Cluster.

```
aws memorydb create-cluster \  
  --cluster-name my-new-cluster \  
  --node-type db.r6g.large \  
  --acl-name my-acl \  
  --subnet-group my-sg
```

Ausgabe:

```
{  
  "Cluster": {  
    "Name": "my-new-cluster",  
    "Status": "creating",  
    "NumberOfShards": 1,  
    "AvailabilityMode": "MultiAZ",  
    "ClusterEndpoint": {  
      "Port": 6379  
    },  
    "NodeType": "db.r6g.large",  
    "EngineVersion": "6.2",  
    "EnginePatchVersion": "6.2.6",  
    "ParameterGroupName": "default.memorydb-redis6",  
    "ParameterGroupStatus": "in-sync",  
    "SubnetGroupName": "my-sg",  
    "TLSEnabled": true,  
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-new-cluster",  
    "SnapshotRetentionLimit": 0,  
    "MaintenanceWindow": "sat:10:00-sat:11:00",  
    "SnapshotWindow": "07:30-08:30",  
    "ACLName": "my-acl",  
    "AutoMinorVersionUpgrade": true  
  }  
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Managing Clusters](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCluster](#) in der AWS CLI Befehlsreferenz.

create-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `create-parameter-group`.

AWS CLI

Um eine Parametergruppe zu erstellen

Im folgenden `create-parameter-group` Beispiel wird eine Parametergruppe erstellt.

```
aws memorydb create-parameter-group \  
  --parameter-group-name myRedis6x \  
  --family memorydb_redis6 \  
  --description "my-parameter-group"
```

Ausgabe:

```
{  
  "ParameterGroup": {  
    "Name": "myredis6x",  
    "Family": "memorydb_redis6",  
    "Description": "my-parameter-group",  
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Parametergruppe](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateParameterGroup AWS CLI](#) Befehlsreferenz.

create-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-snapshot`.

AWS CLI

Um einen Snapshot zu erstellen

Im folgenden `create-snapshot` Beispiel wird ein Snapshot erstellt.

```
aws memorydb create-snapshot \  
  --cluster-name my-cluster \  
  --snapshot-name my-cluster-snapshot
```

Ausgabe:

```
{  
  "Snapshot": {  
    "Name": "my-cluster-snapshot1",  
    "Status": "creating",  
    "Source": "manual",  
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-snapshot",  
    "ClusterConfiguration": {  
      "Name": "my-cluster",  
      "Description": "",  
      "NodeType": "db.r6g.large",  
      "EngineVersion": "6.2",  
      "MaintenanceWindow": "wed:03:00-wed:04:00",  
      "Port": 6379,  
      "ParameterGroupName": "default.memorydb-redis6",  
      "SubnetGroupName": "my-sg",  
      "VpcId": "vpc-862xxxxc",  
      "SnapshotRetentionLimit": 0,  
      "SnapshotWindow": "04:30-05:30",  
      "NumShards": 2  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Manuelles Erstellen von Snapshots](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSnapshot](#) in AWS CLI der Befehlsreferenz.

create-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `create-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu erstellen

Im folgenden `create-subnet-group` Beispiel wird eine Subnetzgruppe erstellt.

```
aws memorydb create-subnet-group \  
  --subnet-group-name mysubnetgroup \  
  --description "my subnet group" \  
  --subnet-ids subnet-5623xxxx
```

Ausgabe:

```
{  
  "SubnetGroup": {  
    "Name": "mysubnetgroup",  
    "Description": "my subnet group",  
    "VpcId": "vpc-86257xxx",  
    "Subnets": [  
      {  
        "Identifier": "subnet-5623xxxx",  
        "AvailabilityZone": {  
          "Name": "us-east-1a"  
        }  
      }  
    ],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateSubnetGroup](#).AWS CLI

create-user

Das folgende Codebeispiel zeigt die Verwendung `create-user`.

AWS CLI

Um einen Benutzer zu erstellen

Das folgende `create-user` Beispiel erstellt einen neuen Benutzer.

```
aws memorydb create-user \  
  --user-name user-name-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --authentication-mode \  
    Passwords="enterapasswordhere",Type=password
```

Ausgabe:

```
{  
  "User": {  
    "Name": "user-name-1",  
    "Status": "active",  
    "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",  
    "ACLNames": [],  
    "MinimumEngineVersion": "6.2",  
    "Authentication": {  
      "Type": "password",  
      "PasswordCount": 1  
    },  
    "ARN": "arn:aws:memorydb:us-west-2:491658xxxxxx:user/user-name-1"  
  }  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [CreateUser](#).AWS CLI

`delete-acl`

Das folgende Codebeispiel zeigt die Verwendung `delete-acl`.

AWS CLI

Um ein zu löschen ACL

Im folgenden `delete-acl` Beispiel wird eine Zugriffskontrollliste gelöscht.

```
aws memorydb delete-acl \  
  --acl-name "new-acl-1"
```

Ausgabe:

```
{  
  "ACL": {  
    "Name": "new-acl-1",  
    "Status": "deleting",  
    "UserNames": [  
      "pat"  
    ],  
    "MinimumEngineVersion": "6.2",  
    "Clusters": [],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"  
  }  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteAcl](#).AWS CLI

delete-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster`.

AWS CLI

Löschen eines Clusters

Das folgende `delete-cluster` Beispiel löscht einen Cluster.

```
aws memorydb delete-cluster \  
  --cluster-name my-new-cluster
```

Ausgabe:

```
{  
  "Cluster": {  
    "Name": "my-new-cluster",
```

```

    "Status": "deleting",
    "NumberOfShards": 1,
    "ClusterEndpoint": {
      "Address": "clustercfg.my-new-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
      "Port": 6379
    },
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "EnginePatchVersion": "6.2.6",
    "ParameterGroupName": "default.memorydb-redis6",
    "ParameterGroupStatus": "in-sync",
    "SubnetGroupName": "my-sg",
    "TLSEnabled": true,
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-new-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "sat:10:00-sat:11:00",
    "SnapshotWindow": "07:30-08:30",
    "AutoMinorVersionUpgrade": true
  }
}

```

Weitere Informationen finden Sie unter [Löschen eines Clusters](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteCluster](#) in der AWS CLI Befehlsreferenz.

delete-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `delete-parameter-group`.

AWS CLI

Um eine Parametergruppe zu löschen

Im folgenden `delete-parameter-group` Beispiel wird eine Parametergruppe gelöscht.

```

aws memorydb delete-parameter-group \
  --parameter-group-name myRedis6x

```

Ausgabe:

```

{
  "ParameterGroup": {

```



```
    "Name": "myredis6x",
    "Family": "memorydb_redis6",
    "Description": "my-parameter-group",
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"
  }
}
```

Weitere Informationen finden Sie unter [Löschen einer Parametergruppe](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteParameterGroup AWS CLI Befehlsreferenz](#).

delete-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-snapshot`.

AWS CLI

So löschen Sie einen Snapshot

Das folgende `delete-snapshot` Beispiel löscht einen Snapshot.

```
aws memorydb delete-snapshot \  
  --snapshot-name my-cluster-snapshot
```

Ausgabe:

```
{  
  "Snapshot": {  
    "Name": "my-cluster-snapshot",  
    "Status": "deleting",  
    "Source": "manual",  
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-snapshot",  
    "ClusterConfiguration": {  
      "Name": "my-cluster",  
      "Description": "",  
      "NodeType": "db.r6g.large",  
      "EngineVersion": "6.2",  
      "MaintenanceWindow": "wed:03:00-wed:04:00",  
      "Port": 6379,  
      "ParameterGroupName": "default.memorydb-redis6",  
      "SubnetGroupName": "my-sg",
```

```
        "VpcId": "vpc-862xxxxc",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "04:30-05:30",
        "NumShards": 2
    }
}
```

Weitere Informationen finden Sie unter [Löschen eines Snapshots](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSnapshot](#) in der AWS CLI Befehlsreferenz.

delete-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `delete-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu löschen

Im folgenden `delete-subnet-group` Beispiel wird ein Subnetz gelöscht.

```
aws memorydb delete-subnet-group \
  --subnet-group-name mysubnetgroup
```

Ausgabe:

```
{
  "SubnetGroup": {
    "Name": "mysubnetgroup",
    "Description": "my subnet group",
    "VpcId": "vpc-86xxxx4fc",
    "Subnets": [
      {
        "Identifier": "subnet-56xxx61b",
        "AvailabilityZone": {
          "Name": "us-east-1a"
        }
      }
    ],
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Löschen einer Subnetzgruppe](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteSubnetGroup](#).AWS CLI

delete-user

Das folgende Codebeispiel zeigt die Verwendung `delete-user`.

AWS CLI

Benutzer löschen

Das folgende `delete-user` Beispiel löscht einen Benutzer.

```
aws memorydb delete-user \  
  --user-name my-user
```

Ausgabe:

```
{  
  "User": {  
    "Name": "my-user",  
    "Status": "deleting",  
    "AccessString": "on ~app:* resetchannels -@all +@read",  
    "ACLNames": [  
      "my-acl"  
    ],  
    "MinimumEngineVersion": "6.2",  
    "Authentication": {  
      "Type": "password",  
      "PasswordCount": 1  
    },  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"  
  }  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteUser](#).AWS CLI

describe-acls

Das folgende Codebeispiel zeigt die Verwendung `describe-acls`.

AWS CLI

Um eine Liste von zurückzugeben ACLs

Das folgende `describe-acls`` gibt eine Liste von zurück. ACLs

```
aws memorydb describe-acls
```

Ausgabe:

```
{
  "ACLs": [
    {
      "Name": "open-access",
      "Status": "active",
      "UserNames": [
        "default"
      ],
      "MinimumEngineVersion": "6.2",
      "Clusters": [],
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/open-access"
    },
    {
      "Name": "my-acl",
      "Status": "active",
      "UserNames": [],
      "MinimumEngineVersion": "6.2",
      "Clusters": [
        "my-cluster"
      ],
      "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxxx:acl/my-acl"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DescribeAcls](#).AWS CLI

describe-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-clusters`.

AWS CLI

Um eine Liste von Clustern zurückzugeben

Der folgende Befehl `describe-clusters`` gibt eine Liste von Clustern zurück.

```
aws memorydb describe-clusters
```

Ausgabe:

```
{
  "Clusters": [
    {
      "Name": "my-cluster",
      "Status": "available",
      "NumberOfShards": 2,
      "ClusterEndpoint": {
        "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
        "Port": 6379
      },
      "NodeType": "db.r6g.large",
      "EngineVersion": "6.2",
      "EnginePatchVersion": "6.2.6",
      "ParameterGroupName": "default.memorydb-redis6",
      "ParameterGroupStatus": "in-sync",
      "SecurityGroups": [
        {
          "SecurityGroupId": "sg-0a1434xxxxxc9fae",
          "Status": "active"
        }
      ],
      "SubnetGroupName": "pat-sg",
      "TLSEnabled": true,
      "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-cluster",
      "SnapshotRetentionLimit": 0,
      "MaintenanceWindow": "wed:03:00-wed:04:00",
      "SnapshotWindow": "04:30-05:30",
      "ACLName": "my-acl",
    }
  ]
}
```

```
        "AutoMinorVersionUpgrade": true
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Cluster verwalten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeClusters](#) in der AWS CLI Befehlsreferenz.

describe-engine-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-engine-versions`.

AWS CLI

Um eine Liste von Engine-Versionen zurückzugeben

Das folgende `describe-engine-versions` gibt eine Liste von Engine-Versionen zurück.

```
aws memorydb describe-engine-versions
```

Ausgabe:

```
{
  "EngineVersions": [
    {
      "EngineVersion": "6.2",
      "EnginePatchVersion": "6.2.6",
      "ParameterGroupFamily": "memorydb_redis6"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Engine-Versionen und Upgrades](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEngineVersions](#) in der AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um eine Liste von Ereignissen zurückzugeben

Der folgende Befehl `describe-events` gibt eine Liste von Ereignissen zurück.

```
aws memorydb describe-events
```

Ausgabe:

```
{
  "Events": [
    {
      "SourceName": "my-cluster",
      "SourceType": "cluster",
      "Message": "Increase replica count started for replication group my-cluster on 2022-07-22T14:09:01.440Z",
      "Date": "2022-07-22T07:09:01.443000-07:00"
    },
    {
      "SourceName": "my-user",
      "SourceType": "user",
      "Message": "Create user my-user operation completed.",
      "Date": "2022-07-22T07:00:02.975000-07:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Überwachung von Ereignissen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

`describe-parameter-groups`

Das folgende Codebeispiel zeigt die Verwendung `describe-parameter-groups`.

AWS CLI

Um eine Liste von Parametergruppen zurückzugeben

Das folgende `describe-parameter-groups` gibt eine Liste von Parametergruppen zurück.

```
aws memorydb describe-parameter-groups
```

Ausgabe:

```
{
  "ParameterGroups": [
    {
      "Name": "default.memorydb-redis6",
      "Family": "memorydb_redis6",
      "Description": "Default parameter group for memorydb_redis6",
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/default.memorydb-redis6"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Konfiguration von Engine-Parametern mithilfe von Parametergruppen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeParameterGroups AWS CLI Befehlsreferenz](#).

describe-parameters

Das folgende Codebeispiel zeigt die Verwendung von `describe-parameters`.

AWS CLI

Um eine Liste von Parametern zurückzugeben

Der folgende Befehl vom Typ `describe-parameters` gibt eine Liste von Parametern zurück.

```
aws memorydb describe-parameters
```

Ausgabe:

```
{
  "Parameters": [
    {
      "Name": "acllog-max-len",
      "Value": "128",
      "Description": "The maximum length of the ACL Log",
      "DataType": "integer",
    }
  ]
}
```



```

    "AllowedValues": "1-10000",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "activedefrag",
    "Value": "no",
    "Description": "Enabled active memory defragmentation",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-cycle-max",
    "Value": "75",
    "Description": "Maximal effort for defrag in CPU percentage",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-cycle-min",
    "Value": "5",
    "Description": "Minimal effort for defrag in CPU percentage",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-ignore-bytes",
    "Value": "104857600",
    "Description": "Minimum amount of fragmentation waste to start active
defrag",
    "DataType": "integer",
    "AllowedValues": "1048576-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-max-scan-fields",
    "Value": "1000",
    "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
    "DataType": "integer",
    "AllowedValues": "1-1000000",
    "MinimumEngineVersion": "6.2.4"
  }

```

```

    },
    {
      "Name": "active-defrag-threshold-lower",
      "Value": "10",
      "Description": "Minimum percentage of fragmentation to start active
defrag",
      "DataType": "integer",
      "AllowedValues": "1-100",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "active-defrag-threshold-upper",
      "Value": "100",
      "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
      "DataType": "integer",
      "AllowedValues": "1-100",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "active-expire-effort",
      "Value": "1",
      "Description": "The amount of effort that redis uses to expire items in
the active expiration job",
      "DataType": "integer",
      "AllowedValues": "1-10",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "activeresharding",
      "Value": "yes",
      "Description": "Apply rehashing or not",
      "DataType": "string",
      "AllowedValues": "yes,no",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "client-output-buffer-limit-normal-hard-limit",
      "Value": "0",
      "Description": "Normal client output buffer hard limit in bytes",
      "DataType": "integer",
      "AllowedValues": "0-",
      "MinimumEngineVersion": "6.2.4"
    },
  },

```

```
{
  "Name": "client-output-buffer-limit-normal-soft-limit",
  "Value": "0",
  "Description": "Normal client output buffer soft limit in bytes",
  "DataType": "integer",
  "AllowedValues": "0-",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "client-output-buffer-limit-normal-soft-seconds",
  "Value": "0",
  "Description": "Normal client output buffer soft limit in seconds",
  "DataType": "integer",
  "AllowedValues": "0-",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "client-output-buffer-limit-pubsub-hard-limit",
  "Value": "33554432",
  "Description": "Pubsub client output buffer hard limit in bytes",
  "DataType": "integer",
  "AllowedValues": "0-",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "client-output-buffer-limit-pubsub-soft-limit",
  "Value": "8388608",
  "Description": "Pubsub client output buffer soft limit in bytes",
  "DataType": "integer",
  "AllowedValues": "0-",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "client-output-buffer-limit-pubsub-soft-seconds",
  "Value": "60",
  "Description": "Pubsub client output buffer soft limit in seconds",
  "DataType": "integer",
  "AllowedValues": "0-",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "hash-max-ziplist-entries",
  "Value": "512",
```

```
    "Description": "The maximum number of hash entries in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hash-max-ziplist-value",
    "Value": "64",
    "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hll-sparse-max-bytes",
    "Value": "3000",
    "Description": "HyperLogLog sparse representation bytes limit",
    "DataType": "integer",
    "AllowedValues": "1-16000",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-eviction",
    "Value": "no",
    "Description": "Perform an asynchronous delete on evictions",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-expire",
    "Value": "no",
    "Description": "Perform an asynchronous delete on expired keys",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-server-del",
    "Value": "no",
    "Description": "Perform an asynchronous delete on key updates",
    "DataType": "string",
```

```
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-user-del",
    "Value": "no",
    "Description": "Specifies whether the default behavior of DEL command
acts the same as UNLINK",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lfu-decay-time",
    "Value": "1",
    "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lfu-log-factor",
    "Value": "10",
    "Description": "The log factor for incrementing key counter for LFU
eviction policy",
    "DataType": "integer",
    "AllowedValues": "1-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "list-compress-depth",
    "Value": "0",
    "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "maxmemory-policy",
    "Value": "noeviction",
    "Description": "Max memory policy",
```

```

        "DataType": "string",
        "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "maxmemory-samples",
        "Value": "3",
        "Description": "Max memory samples",
        "DataType": "integer",
        "AllowedValues": "1-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "notify-keyspace-events",
        "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
        "DataType": "string",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "set-max-intset-entries",
        "Value": "512",
        "Description": "The limit in the size of the set in order for the
dataset to be compressed",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "slowlog-log-slower-than",
        "Value": "10000",
        "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command",
        "DataType": "integer",
        "AllowedValues": "-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "slowlog-max-len",
        "Value": "128",

```

```

        "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "stream-node-max-bytes",
        "Value": "4096",
        "Description": "The maximum size of a single node in a stream in bytes",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "stream-node-max-entries",
        "Value": "100",
        "Description": "The maximum number of items a single node in a stream
can contain",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "tcp-keepalive",
        "Value": "300",
        "Description": "If non-zero, send ACKs every given number of seconds",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "timeout",
        "Value": "0",
        "Description": "Close connection if client is idle for a given number of
seconds, or never if 0",
        "DataType": "integer",
        "AllowedValues": "0,20-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "tracking-table-max-keys",
        "Value": "1000000",

```

```

        "Description": "The maximum number of keys allowed for the tracking
table for client side caching",
        "DataType": "integer",
        "AllowedValues": "1-100000000",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "zset-max-ziplist-entries",
        "Value": "128",
        "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "zset-max-ziplist-value",
        "Value": "64",
        "Description": "The threshold of biggest sorted set entries in order for
the dataset to be compressed",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    }
]
}

```

Weitere Informationen finden Sie unter [Konfiguration von Engine-Parametern mithilfe von Parametergruppen im MemoryDB-Benutzerhandbuch](#).

- API-Einheiten finden Sie unter [DescribeParameters AWS CLI-Befehlsreferenz](#).

describe-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshots`.

AWS CLI

Um eine Liste von Schnappschüssen zurückzugeben

Der folgende Befehl `describe-snapshots`` gibt eine Liste von Schnappschüssen zurück.

```
aws memorydb describe-snapshots
```


Ausgabe:

```
{
  "Snapshots": [
    {
      "Name": "my-cluster-snapshot",
      "Status": "available",
      "Source": "manual",
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx2:snapshot/my-cluster-
snapshot",
      "ClusterConfiguration": {
        "Name": "my-cluster",
        "Description": " ",
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "Port": 6379,
        "ParameterGroupName": "default.memorydb-redis6",
        "SubnetGroupName": "my-sg",
        "VpcId": "vpc-862574fc",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "04:30-05:30",
        "NumShards": 2
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSnapshots](#) in der AWS CLI Befehlsreferenz.

describe-subnet-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-subnet-groups`.

AWS CLI

Um eine Liste von Subnetzgruppen zurückzugeben

Das folgende `describe-subnet-groups` gibt eine Liste von Subnetzgruppen zurück.

```
aws memorydb describe-subnet-groups
```

Output

```
{
  "SubnetGroups": [
    {
      "Name": "my-sg",
      "Description": "pat-sg",
      "VpcId": "vpc-86xxx4fc",
      "Subnets": [
        {
          "Identifier": "subnet-faxx84a6",
          "AvailabilityZone": {
            "Name": "us-east-1b"
          }
        },
        {
          "Identifier": "subnet-56xxf61b",
          "AvailabilityZone": {
            "Name": "us-east-1a"
          }
        }
      ],
      "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:subnetgroup/my-sg"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeSubnetGroups](#) Befehlsreferenz.AWS CLI

describe-users

Das folgende Codebeispiel zeigt die Verwendung `describe-users`.

AWS CLI

Um eine Liste von Benutzern zurückzugeben

Der folgende Befehl `describe-users`` gibt eine Liste von Benutzern zurück.

```
aws memorydb describe-users
```

Output

```
{
  "Users": [
    {
      "Name": "default",
      "Status": "active",
      "AccessString": "on ~* &* +@all",
      "ACLNames": [
        "open-access"
      ],
      "MinimumEngineVersion": "6.0",
      "Authentication": {
        "Type": "no-password"
      },
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/default"
    },
    {
      "Name": "my-user",
      "Status": "active",
      "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
      "ACLNames": [],
      "MinimumEngineVersion": "6.2",
      "Authentication": {
        "Type": "password",
        "PasswordCount": 2
      },
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DescribeUsers](#).AWS CLI

failover-shard

Das folgende Codebeispiel zeigt die Verwendung `failover-shard`.

AWS CLI

Um ein Failover für einen Shard durchzuführen

Beim folgenden Failover-Shell-Befehl wird ein Failover für einen Shard durchgeführt.

```
aws memorydb failover-shard \  
  --cluster-name my-cluster --shard-name 0001
```

Ausgabe:

```
{  
  "Cluster": {  
    "Name": "my-cluster",  
    "Status": "available",  
    "NumberOfShards": 2,  
    "ClusterEndpoint": {  
      "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-  
east-1.amazonaws.com",  
      "Port": 6379  
    },  
    "NodeType": "db.r6g.large",  
    "EngineVersion": "6.2",  
    "EnginePatchVersion": "6.2.6",  
    "ParameterGroupName": "default.memorydb-redis6",  
    "ParameterGroupStatus": "in-sync",  
    "SecurityGroups": [  
      {  
        "SecurityGroupId": "sg-0a143xxxx45c9fae",  
        "Status": "active"  
      }  
    ],  
    "SubnetGroupName": "my-sg",  
    "TLSEnabled": true,  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",  
    "SnapshotRetentionLimit": 0,  
    "MaintenanceWindow": "wed:03:00-wed:04:00",  
    "SnapshotWindow": "04:30-05:30",  
    "AutoMinorVersionUpgrade": true  
  }  
}
```

Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten mit MultiAZ](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [FailoverShard](#).AWS CLI

list-allowed-node-type-updates

Das folgende Codebeispiel zeigt die Verwendung `list-allowed-node-type-updates`.

AWS CLI

Um eine Liste der zulässigen Knotentyp-Updates zurückzugeben

Der folgende Befehl `list-allowed-node-type -updates` gibt eine Liste verfügbarer Knotentyp-Updates zurück.

```
aws memorydb list-allowed-node-type-updates
```

Ausgabe:

```
{
  "Cluster": {
    "Name": "my-cluster",
    "Status": "available",
    "NumberOfShards": 2,
    "ClusterEndpoint": {
      "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
      "Port": 6379
    },
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "EnginePatchVersion": "6.2.6",
    "ParameterGroupName": "default.memorydb-redis6",
    "ParameterGroupStatus": "in-sync",
    "SecurityGroups": [
      {
        "SecurityGroupId": "sg-0a143xxxx45c9fae",
        "Status": "active"
      }
    ],
    "SubnetGroupName": "my-sg",
    "TLSEnabled": true,
  }
}
```

```
"ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
"SnapshotRetentionLimit": 0,
"MaintenanceWindow": "wed:03:00-wed:04:00",
"SnapshotWindow": "04:30-05:30",
"AutoMinorVersionUpgrade": true
}
}
```

Weitere Informationen finden Sie unter [Skalierung](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAllowedNodeTypeUpdates](#) in der AWS CLI Befehlsreferenz.

list-tags

Das folgende Codebeispiel zeigt die Verwendung `list-tags`.

AWS CLI

Um eine Liste von Tags zurückzugeben

Die folgenden List-Tags geben eine Liste von Tags zurück.

```
aws memorydb list-tags \
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster
```

Ausgabe:

```
{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging resources](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTags](#) in der AWS CLI Befehlsreferenz.

reset-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `reset-parameter-group`.

AWS CLI

Um eine Parametergruppe zurückzusetzen

Das folgende `reset-parameter-group` setzt eine Parametergruppe zurück.

```
aws memorydb reset-parameter-group \  
  --parameter-group-name my-parameter-group \  
  --all-parameters
```

Ausgabe:

```
{  
  "ParameterGroup": {  
    "Name": "my-parameter-group",  
    "Family": "memorydb_redis6",  
    "Description": "my parameter group",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-group"  
  }  
}
```

Weitere Informationen finden Sie unter [Konfiguration von Engine-Parametern mithilfe von Parametergruppen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ResetParameterGroup AWS CLI Befehlsreferenz](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

Die folgende Tag-Ressource fügt einer Ressource ein Tag hinzu.

```
aws memorydb tag-resource \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster \  
  --tags Key="mykey",Value="myvalue"
```

Ausgabe:

```
{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    },
    {
      "Key": "mykey",
      "Value": "myvalue"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Ressourcen taggen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in AWS CLI der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein zu aktualisieren ACL

Die folgende Update-ACL aktualisiert einen ACL durch Hinzufügen eines Benutzers.

```
aws memorydb untag-resource \
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \
  --tag-keys mykey
```

Ausgabe:

```
{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    }
  ]
}
```


Weitere Informationen finden Sie unter [Ressourcen taggen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in AWS CLI der Befehlsreferenz.

update-cluster

Das folgende Codebeispiel zeigt die Verwendung `update-cluster`.

AWS CLI

Um einen Cluster zu aktualisieren

Der folgende Update-Cluster`` aktualisiert die Parametergruppe eines Clusters auf `my-parameter-group`

```
aws memorydb update-cluster \
  --cluster-name my-cluster \
  --parameter-group-name my-parameter-group
```

Ausgabe:

```
{
  "Cluster": {
    "Name": "my-cluster",
    "Status": "available",
    "NumberOfShards": 2,
    "AvailabilityMode": "MultiAZ",
    "ClusterEndpoint": {
      "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
      "Port": 6379
    },
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "EnginePatchVersion": "6.2.6",
    "ParameterGroupName": "my-parameter-group",
    "ParameterGroupStatus": "in-sync",
    "SecurityGroups": [
      {
        "SecurityGroupId": "sg-0a143xxxxxc9fae",
        "Status": "active"
      }
    ],
  },
}
```

```

    "SubnetGroupName": "pat-sg",
    "TLSEnabled": true,
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "SnapshotWindow": "04:30-05:30",
    "ACLName": "my-acl",
    "AutoMinorVersionUpgrade": true
  }
}

```

Weitere Informationen finden Sie unter [Modifizieren eines Clusters](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateCluster](#) in der AWS CLI Befehlsreferenz.

update-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `update-parameter-group`.

AWS CLI

Um eine Parametergruppe zu aktualisieren

Das folgende `update-parameter-group` aktualisiert eine Parametergruppe.

```

aws memorydb update-parameter-group \
  --parameter-group-name my-parameter-group \
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"

```

Ausgabe:

```

{
  "ParameterGroup": {
    "Name": "my-parameter-group",
    "Family": "memorydb_redis6",
    "Description": "my parameter group",
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
  }
}

```

Weitere Informationen finden Sie unter [Ändern einer Parametergruppe](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UpdateParameterGroup AWS CLI Befehlsreferenz](#).

update-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `update-subnet-group`.

AWS CLI

Um eine Subnetzgruppe zu aktualisieren

Das folgende `update-subnet-group` aktualisiert die Subnetz-ID einer Subnetzgruppe.

```
aws memorydb update-subnet-group \  
  --subnet-group-name my-sg \  
  --subnet-ids subnet-01f29d458f3xxxxx
```

Ausgabe:

```
{  
  "SubnetGroup": {  
    "Name": "my-sg-1",  
    "Description": "my-sg",  
    "VpcId": "vpc-09d2cfc01xxxxxxx",  
    "Subnets": [  
      {  
        "Identifier": "subnet-01f29d458fxxxxxx",  
        "AvailabilityZone": {  
          "Name": "us-east-1a"  
        }  
      }  
    ],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"  
  }  
}
```

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UpdateSubnetGroup Befehlsreferenz](#). AWS CLI

update-user

Das folgende Codebeispiel zeigt die Verwendung `update-user`.

AWS CLI

Um einen Benutzer zu aktualisieren

Im Folgenden `update-user` wird die Zugriffszeichenfolge eines Benutzers geändert.

```
aws memorydb update-user \  
  --user-name my-user \  
  --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"
```

Ausgabe:

```
{  
  "User": {  
    "Name": "my-user",  
    "Status": "modifying",  
    "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",  
    "ACLNames": [  
      "myt-acl"  
    ],  
    "MinimumEngineVersion": "6.2",  
    "Authentication": {  
      "Type": "password",  
      "PasswordCount": 2  
    },  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"  
  }  
}
```

Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Zugriffskontrolllisten](#) im MemoryDB-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [UpdateUser](#).AWS CLI

MSK Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren MSK.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-cluster`.

AWS CLI

Um einen MSK Amazon-Cluster zu erstellen

Im folgenden `create-cluster` Beispiel wird ein MSK Cluster `MessagingCluster` mit dem Namen `drei Broker-Knoten` erstellt. Eine JSON Datei mit dem Namen `brokernodegroupinfo.json` gibt die drei Subnetze an, über die Amazon die Broker-Knoten verteilen MSK soll. In diesem Beispiel wird die Überwachungsebene nicht angegeben, sodass der Cluster die DEFAULT Ebene erhält.

```
aws kafka create-cluster \  
  --cluster-name "MessagingCluster" \  
  --broker-node-group-info file://brokernodegroupinfo.json \  
  --kafka-version "2.2.1" \  
  --number-of-broker-nodes 3
```

Inhalt von `brokernodegroupinfo.json`:

```
{  
  "InstanceType": "kafka.m5.xlarge",  
  "BrokerAZDistribution": "DEFAULT",  
  "ClientSubnets": [  
    "subnet-0123456789111abcd",  
    "subnet-0123456789222abcd",  
    "subnet-0123456789333abcd"
```

```
]
}
```

Ausgabe:

```
{
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "ClusterName": "MessagingCluster",
  "State": "CREATING"
}
```

Weitere Informationen finden Sie unter [Erstellen eines MSK Amazon-Clusters im Amazon Managed Streaming for Apache Kafka](#).

- API-Einheiten finden Sie unter [CreateCluster AWS CLI-Befehlsreferenz](#).

create-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-configuration`.

AWS CLI

Um eine benutzerdefinierte MSK Amazon-Konfiguration zu erstellen

Im folgenden `create-configuration` Beispiel wird eine benutzerdefinierte MSK Konfiguration mit den Servereigenschaften erstellt, die in der Eingabedatei angegeben sind.

```
aws kafka create-configuration \
  --name "CustomConfiguration" \
  --description "Topic autocreation enabled; Apache ZooKeeper timeout 2000 ms; Log
rolling 604800000 ms." \
  --kafka-versions "2.2.1" \
  --server-properties file://configuration.txt
```

Inhalt von `configuration.txt`:

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Ausgabe:

```
{
  "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "CreationTime": "2019-10-09T15:26:05.548Z",
  "LatestRevision":
    {
      "CreationTime": "2019-10-09T15:26:05.548Z",
      "Description": "Topic autocreation enabled; Apache ZooKeeper timeout
2000 ms; Log rolling 604800000 ms.",
      "Revision": 1
    },
  "Name": "CustomConfiguration"
}
```

Weitere Informationen finden Sie unter [Amazon MSK Configuration Operations](#) im Amazon Managed Streaming for Apache Kafka Developer Guide.

- API-Einheiten finden Sie [CreateConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-cluster

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster`.

AWS CLI

Um einen Cluster zu beschreiben

Das folgende `describe-cluster` Beispiel beschreibt einen MSK Amazon-Cluster.

```
aws kafka describe-cluster \
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

Ausgabe:

```
{
  "ClusterInfo": {
    "BrokerNodeGroupInfo": {
      "BrokerAZDistribution": "DEFAULT",
      "ClientSubnets": [
        "subnet-cbfff283",
        "subnet-6746046b"
      ]
    }
  }
}
```

```
    ],
    "InstanceType": "kafka.m5.large",
    "SecurityGroups": [
      "sg-f839b688"
    ],
    "StorageInfo": {
      "EbsStorageInfo": {
        "VolumeSize": 100
      }
    }
  },
  "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
  "ClusterName": "demo-cluster-1",
  "CreationTime": "2020-07-09T02:31:36.223000+00:00",
  "CurrentBrokerSoftwareInfo": {
    "KafkaVersion": "2.2.1"
  },
  "CurrentVersion": "K3AEGXETSR30VB",
  "EncryptionInfo": {
    "EncryptionAtRest": {
      "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
    },
    "EncryptionInTransit": {
      "ClientBroker": "TLS_PLAINTEXT",
      "InCluster": true
    }
  },
  "EnhancedMonitoring": "DEFAULT",
  "OpenMonitoring": {
    "Prometheus": {
      "JmxExporter": {
        "EnabledInBroker": false
      },
      "NodeExporter": {
        "EnabledInBroker": false
      }
    }
  },
  "NumberOfBrokerNodes": 2,
  "State": "ACTIVE",
  "Tags": {},
```



```
"ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:2181"  
  }  
}
```

Weitere Informationen finden Sie unter [MSK Amazon-Cluster auflisten im Amazon Managed Streaming for Apache Kafka Developer Guide](#).

- API-Einheiten finden Sie [DescribeCluster](#) in der AWS CLI Befehlsreferenz.

get-bootstrap-brokers

Das folgende Codebeispiel zeigt die Verwendung `get-bootstrap-brokers`.

AWS CLI

Um Bootstrap-Broker zu bekommen

Im folgenden `get-bootstrap-brokers` Beispiel werden die Bootstrap-Broker-Informationen für einen MSK Amazon-Cluster abgerufen.

```
aws kafka get-bootstrap-brokers \  
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-  
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

Ausgabe:

```
{  
  "BootstrapBrokerString": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:9092,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:9092",  
  "BootstrapBrokerStringTls": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:9094,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-  
east-1.amazonaws.com:9094"  
}
```

Weitere Informationen finden Sie unter [Getting the Bootstrap Brokers](#) im Amazon Managed Streaming for Apache Kafka Developer Guide.

- API-Einheiten finden Sie [GetBootstrapBrokers](#) in der AWS CLI Befehlsreferenz.

list-clusters

Das folgende Codebeispiel zeigt die Verwendung `list-clusters`.

AWS CLI

Um die verfügbaren Cluster aufzulisten

Das folgende `list-clusters` Beispiel listet die MSK Amazon-Cluster in Ihrem AWS Konto auf.

```
aws kafka list-clusters
```

Ausgabe:

```
{
  "ClusterInfoList": [
    {
      "BrokerNodeGroupInfo": {
        "BrokerAZDistribution": "DEFAULT",
        "ClientSubnets": [
          "subnet-cbfff283",
          "subnet-6746046b"
        ],
        "InstanceType": "kafka.m5.large",
        "SecurityGroups": [
          "sg-f839b688"
        ],
        "StorageInfo": {
          "EbsStorageInfo": {
            "VolumeSize": 100
          }
        }
      },
      "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
      "ClusterName": "demo-cluster-1",
      "CreationTime": "2020-07-09T02:31:36.223000+00:00",
      "CurrentBrokerSoftwareInfo": {
        "KafkaVersion": "2.2.1"
      },
      "CurrentVersion": "K3AEGXETSR30VB",
      "EncryptionInfo": {
        "EncryptionAtRest": {
```

```

        "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
    },
    "EncryptionInTransit": {
        "ClientBroker": "TLS_PLAINTEXT",
        "InCluster": true
    }
},
"EnhancedMonitoring": "DEFAULT",
"OpenMonitoring": {
    "Prometheus": {
        "JmxExporter": {
            "EnabledInBroker": false
        },
        "NodeExporter": {
            "EnabledInBroker": false
        }
    }
},
"NumberOfBrokerNodes": 2,
"State": "ACTIVE",
"Tags": {},
"ZookeeperConnectionString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
    }
]
}

```

Weitere Informationen finden Sie unter [MSK Amazon-Cluster auflisten im Amazon Managed Streaming for Apache Kafka Developer Guide](#).

- API-Einheiten finden Sie [ListClusters](#) in der AWS CLI Befehlsreferenz.

update-broker-storage

Das folgende Codebeispiel zeigt die Verwendung `update-broker-storage`.

AWS CLI

Um den EBS Speicher für Makler zu aktualisieren

Im folgenden `update-broker-storage` Beispiel wird die EBS Speichermenge für alle Broker im Cluster aktualisiert. Amazon MSK legt den Zielspeicherbetrag für jeden Broker auf den im Beispiel angegebenen Betrag fest. Sie können die aktuelle Version des Clusters abrufen, indem Sie den Cluster beschreiben oder indem Sie alle Cluster auflisten.

```
aws kafka update-broker-storage \  
  --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \  
  --current-version "K21V3IB1VIZYYH" \  
  --target-broker-efs-volume-info "KafkaBrokerNodeId=ALL,VolumeSizeGB=1100"
```

Die Ausgabe gibt ARN für diesen `update-broker-storage` Vorgang eine zurück. Um festzustellen, ob dieser Vorgang abgeschlossen ist, verwenden Sie den `describe-cluster-operation` Befehl mit diesem ARN als Eingabe.

```
{  
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-  
operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-  
bcde-33333EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Update the EBS Storage for Brokers](#) im Amazon Managed Streaming for Apache Kafka Developer Guide.

- APIEinzelheiten finden Sie [UpdateBrokerStorage](#) in der AWS CLI Befehlsreferenz.

update-cluster-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-cluster-configuration`.

AWS CLI

Um die Konfiguration eines MSK Amazon-Clusters zu aktualisieren

Das folgende `update-cluster-configuration` Beispiel aktualisiert die Konfiguration des angegebenen vorhandenen MSK Clusters. Es verwendet eine benutzerdefinierte MSK Konfiguration.

```
aws kafka update-cluster-configuration \  
  --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \  
  --current-version "K21V3IB1VIZYYH" \  
  --target-broker-efs-volume-info "KafkaBrokerNodeId=ALL,VolumeSizeGB=1100"
```

```
--cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \  
--configuration-info file://configuration-info.json \  
--current-version "K21V3IB1VIZYYH"
```

Inhalt von configuration-info.json:

```
{  
  "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "Revision": 1  
}
```

Die Ausgabe gibt an ARN für diesen update-cluster-configuration Vorgang zurück. Um festzustellen, ob dieser Vorgang abgeschlossen ist, verwenden Sie den describe-cluster-operation Befehl mit diesem ARN als Eingabe.

```
{  
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-bcde-33333EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Aktualisieren der Konfiguration eines MSK Amazon-Clusters im Amazon](#) Managed Streaming for Apache Kafka Developer Guide.

- API-Einheiten finden Sie [UpdateClusterConfiguration](#) in der AWS CLI Befehlsreferenz.

Network Manager-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Network Manager Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-customer-gateway

Das folgende Codebeispiel zeigt die Verwendung `associate-customer-gateway`.

AWS CLI

Um ein Kunden-Gateway zuzuordnen

Im folgenden `associate-customer-gateway` Beispiel wird das Kunden-Gateway `cgw-11223344556677889` im angegebenen globalen Netzwerk dem Gerät `zugeordnetdevice-07f6fd08867abc123`.

```
aws networkmanager associate-customer-gateway \
  --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
  --global-network-id global-network-01231231231231231 \
  --device-id device-07f6fd08867abc123 \
  --region us-west-2
```

Ausgabe:

```
{
  "CustomerGatewayAssociation": {
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "State": "PENDING"
  }
}
```

Weitere Informationen finden Sie unter [Customer Gateway Associations](#) im Transit Gateway Network Manager Guide.

- API-Einheiten finden Sie [AssociateCustomerGateway](#) unter AWS CLI Befehlsreferenz.

associate-link

Das folgende Codebeispiel zeigt die Verwendung `associate-link`.

AWS CLI

Um einen Link zuzuordnen

Das folgende `associate-link` Beispiel verknüpft einen Link `link-11112222aaaabbbb1` mit einem Gerät `device-07f6fd08867abc123`. Der Link und das Gerät befinden sich im angegebenen globalen Netzwerk.

```
aws networkmanager associate-link \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "LinkAssociation": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkAssociationState": "PENDING"  
  }  
}
```

Weitere Informationen finden Sie unter [Geräte- und Verbindungszuordnungen](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [AssociateLink](#) unter AWS CLI Befehlsreferenz.

create-core-network

Das folgende Codebeispiel zeigt die Verwendung `create-core-network`.

AWS CLI

Um ein Kernnetzwerk zu erstellen

Im folgenden `create-core-network` Beispiel wird ein Kernnetzwerk mithilfe einer optionalen Beschreibung und Tags innerhalb eines WAN globalen AWS Cloud-Netzwerks erstellt.

```
aws networkmanager create-core-network \  
  --global-network-id global-network-cdef-EXAMPLE22222 \  
  --description "Main headquarters location" \  
  --tags Key=Name,Value="New York City office"
```

Ausgabe:

```
{  
  "CoreNetwork": {  
    "GlobalNetworkId": "global-network-cdef-EXAMPLE22222",  
    "CoreNetworkId": "core-network-cdef-EXAMPLE33333",  
    "CoreNetworkArn": "arn:aws:networkmanager::987654321012:core-network/core-network-cdef-EXAMPLE33333",  
    "Description": "Main headquarters location",  
    "CreatedAt": "2022-01-10T19:53:59+00:00",  
    "State": "AVAILABLE",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "New York City office"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie im AWS WANCloud-Benutzerhandbuch unter [Globale Netzwerke und Kernnetzwerke](#).

- API-Einheiten finden Sie [CreateCoreNetwork](#) in der AWS CLI Befehlsreferenz.

create-device

Das folgende Codebeispiel zeigt die Verwendung `create-device`.

AWS CLI

Um ein Gerät zu erstellen

Im folgenden `create-device` Beispiel wird ein Gerät im angegebenen globalen Netzwerk erstellt. Zu den Gerätedetails gehören eine Beschreibung, der Typ, der Hersteller, das Modell und die Seriennummer.

```
aws networkmanager create-device
  --global-network-id global-network-01231231231231231 \
  --description "New York office device" \
  --type "office device" \
  --vendor "anycompany" \
  --model "abcabc" \
  --serial-number "1234" \
  --region us-west-2
```

Ausgabe:

```
{
  "Device": {
    "DeviceId": "device-07f6fd08867abc123",
    "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
    "GlobalNetworkId": "global-network-01231231231231231",
    "Description": "New York office device",
    "Type": "office device",
    "Vendor": "anycompany",
    "Model": "abcabc",
    "SerialNumber": "1234",
    "CreatedAt": 1575554005.0,
    "State": "PENDING"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Geräten](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [CreateDevice](#) unter AWS CLI Befehlsreferenz.

create-global-network

Das folgende Codebeispiel zeigt die Verwendung `create-global-network`.

AWS CLI

Um ein globales Netzwerk zu erstellen

Mit den folgenden `create-global-network` Beispielen wird ein neues globales Netzwerk erstellt. Der Anfangszustand bei der Erstellung ist `PENDING`.

```
aws networkmanager create-global-network
```

Ausgabe:

```
{
  "GlobalNetwork": {
    "GlobalNetworkId": "global-network-00a77fc0f722dae74",
    "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/global-network-00a77fc0f722dae74",
    "CreatedAt": "2022-03-14T20:31:56+00:00",
    "State": "PENDING"
  }
}
```

- API-Einheiten finden Sie [CreateGlobalNetwork](#) in der AWS CLI Befehlsreferenz.

create-link

Das folgende Codebeispiel zeigt die Verwendung `create-link`.

AWS CLI

Um einen Link zu erstellen

Im folgenden `create-link` Beispiel wird ein Link im angegebenen globalen Netzwerk erstellt. Der Link enthält eine Beschreibung und Details zu Linktyp, Bandbreite und Anbieter. Die Site-ID gibt die Site an, der der Link zugeordnet ist.

```
aws networkmanager create-link \
  --global-network-id global-network-01231231231231231 \
  --description "VPN Link" \
  --type "broadband" \
  --bandwidth UploadSpeed=10,DownloadSpeed=20 \
  --provider "AnyCompany" \
  --site-id site-444555aaabbb11223 \
```

```
--region us-west-2
```

Ausgabe:

```
{
  "Link": {
    "LinkId": "link-11112222aaaabbbb1",
    "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
    "GlobalNetworkId": "global-network-01231231231231231",
    "SiteId": "site-444555aaabbb11223",
    "Description": "VPN Link",
    "Type": "broadband",
    "Bandwidth": {
      "UploadSpeed": 10,
      "DownloadSpeed": 20
    },
    "Provider": "AnyCompany",
    "CreatedAt": 1575555811.0,
    "State": "PENDING"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Links](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [CreateLink](#) unter AWS CLI Befehlsreferenz.

create-site

Das folgende Codebeispiel zeigt die Verwendung `create-site`.

AWS CLI

Um eine Site zu erstellen

Im folgenden `create-site` Beispiel wird eine Site im angegebenen globalen Netzwerk erstellt. Die Standortdetails umfassen eine Beschreibung und die Standortinformationen.

```
aws networkmanager create-site \
  --global-network-id global-network-01231231231231231 \
  --description "New York head office" \
  --location Latitude=40.7128,Longitude=-74.0060 \
```

```
--region us-west-2
```

Ausgabe:

```
{
  "Site": {
    "SiteId": "site-444555aaabbb11223",
    "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
    "GlobalNetworkId": "global-network-01231231231231231",
    "Description": "New York head office",
    "Location": {
      "Latitude": "40.7128",
      "Longitude": "-74.0060"
    },
    "CreatedAt": 1575554300.0,
    "State": "PENDING"
  }
}
```

Weitere Informationen finden Sie im Transit Gateway Network Manager-Handbuch unter [Working with Sites](#).

- API-Einheiten finden Sie [CreateSite](#) unter AWS CLI Befehlsreferenz.

create-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `create-vpc-attachment`.

AWS CLI

Um einen VPC-Anhang zu erstellen

Im folgenden `create-vpc-attachment` Beispiel wird ein VPC-Anhang mit IPv6-Unterstützung in einem Kernnetzwerk erstellt.

```
aws networkmanager create-vpc-attachment \
  --core-network-id core-network-0fab62fe438d94db6 \
  --vpc-arn arn:aws:ec2:us-east-1:987654321012:vpc/vpc-09f37f69e2786eeb8 \
  --subnet-arns arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7 \
  --Ipv6Support=true
```

Ausgabe:

```
{
  "VpcAttachment": {
    "Attachment": {
      "CoreNetworkId": "core-network-0fab62fe438d94db6",
      "AttachmentId": "attachment-05e1da6eba87a06e6",
      "OwnerAccountId": "987654321012",
      "AttachmentType": "VPC",
      "State": "CREATING",
      "EdgeLocation": "us-east-1",
      "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/
vpc-09f37f69e2786eeb8",
      "Tags": [],
      "CreatedAt": "2022-03-10T20:59:14+00:00",
      "UpdatedAt": "2022-03-10T20:59:14+00:00"
    },
    "SubnetArns": [
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7"
    ],
    "Options": {
      "Ipv6Support": true
    }
  }
}
```

Weitere Informationen finden Sie im WANCloud-Benutzerhandbuch unter [Einen Anhang erstellen](#).

- API-Einheiten finden Sie [CreateVpcAttachment](#) in der AWS CLI Befehlsreferenz.

delete-attachment

Das folgende Codebeispiel zeigt die Verwendung `delete-attachment`.

AWS CLI

Um einen Anhang zu löschen

Im folgenden `delete-attachment` Beispiel wird eine Connect-Anlage gelöscht.

```
aws networkmanager delete-attachment \
  --attachment-id attachment-01feddaeae26ab68c
```

Ausgabe:

```
{
  "Attachment": {
    "CoreNetworkId": "core-network-0f4b0a9d5ee7761d1",
    "AttachmentId": "attachment-01feddaeeae26ab68c",
    "OwnerAccountId": "987654321012",
    "AttachmentType": "CONNECT",
    "State": "DELETING",
    "EdgeLocation": "us-east-1",
    "ResourceArn": "arn:aws:networkmanager::987654321012:attachment/attachment-02c3964448fedf5aa",
    "CreatedAt": "2022-03-15T19:18:41+00:00",
    "UpdatedAt": "2022-03-15T19:28:59+00:00"
  }
}
```

Weitere Informationen finden Sie unter [Löschen von Anhängen](#) im WANCloud-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAttachment](#) in der AWS CLI Befehlsreferenz.

delete-bucket-analytics-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-bucket-analytics-configuration`.

AWS CLI

Um eine Analytics-Konfiguration für einen Bucket zu löschen

Im folgenden `delete-bucket-analytics-configuration` Beispiel wird die Analytics-Konfiguration für den angegebenen Bucket und die angegebene ID entfernt.

```
aws s3api delete-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBucketAnalyticsConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-bucket-metrics-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-bucket-metrics-configuration`.

AWS CLI

Um eine Metrikkonfiguration für einen Bucket zu löschen

Im folgenden `delete-bucket-metrics-configuration` Beispiel wird die Metrikkonfiguration für den angegebenen Bucket und die angegebene ID entfernt.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

delete-core-network

Das folgende Codebeispiel zeigt die Verwendung `delete-core-network`.

AWS CLI

Um ein Kernnetzwerk zu löschen

Im folgenden `delete-core-network` Beispiel wird ein Kernnetzwerk aus einem WAN globalen Cloud-Netzwerk gelöscht.

```
aws networkmanager delete-core-network \  
  --core-network-id core-network-0fab62fe438d94db6
```

Ausgabe:

```
{  
  "CoreNetwork": {  
    "GlobalNetworkId": "global-network-0d59060f16a73bc41",  
    "CoreNetworkId": "core-network-0fab62fe438d94db6",  
    "Description": "Main headquarters location",  
    "CreatedAt": "2021-12-09T18:31:11+00:00",  
    "State": "DELETING",  
    "Segments": [  
      {  
        "Name": "dev",  
        "EdgeLocations": [  

```

```

        "us-east-1"
      ],
      "SharedSegments": []
    }
  ],
  "Edges": [
    {
      "EdgeLocation": "us-east-1",
      "Asn": 64512,
      "InsideCidrBlocks": []
    }
  ]
}

```

Weitere Informationen finden Sie unter [Kernnetzwerke](#) im WANCloud-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteCoreNetwork](#) in der AWS CLI Befehlsreferenz.

delete-device

Das folgende Codebeispiel zeigt die Verwendung `delete-device`.

AWS CLI

Um ein Gerät zu löschen

Im folgenden `delete-device` Beispiel wird das angegebene Gerät aus dem angegebenen globalen Netzwerk gelöscht.

```

aws networkmanager delete-device \
  --global-network-id global-network-01231231231231231 \
  --device-id device-07f6fd08867abc123 \
  --region us-west-2

```

Ausgabe:

```

{
  "Device": {
    "DeviceId": "device-07f6fd08867abc123",
    "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-network-01231231231231231/device-07f6fd08867abc123",
  }
}

```



```
"GlobalNetworkId": "global-network-01231231231231231",
  "Description": "New York office device",
  "Type": "office device",
  "Vendor": "anycompany",
  "Model": "abcabc",
  "SerialNumber": "1234",
  "SiteId": "site-444555aaabbb11223",
  "CreatedAt": 1575554005.0,
  "State": "DELETING"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Geräten](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [DeleteDevice](#) unter AWS CLI Befehlsreferenz.

delete-global-network

Das folgende Codebeispiel zeigt die Verwendung `delete-global-network`.

AWS CLI

Um ein globales Netzwerk zu löschen

Im folgenden `delete-global-network` Beispiel wird ein globales Netzwerk gelöscht.

```
aws networkmanager delete-global-network \
  --global-network-id global-network-052bedddccb193b6b
```

Ausgabe:

```
{
  "GlobalNetwork": {
    "GlobalNetworkId": "global-network-052bedddccb193b6b",
    "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/global-network-052bedddccb193b6b",
    "CreatedAt": "2021-12-09T18:19:12+00:00",
    "State": "DELETING"
  }
}
```

- API-Einheiten finden Sie unter [DeleteGlobalNetwork](#) AWS CLI-Befehlsreferenz.

delete-link

Das folgende Codebeispiel zeigt die Verwendung `delete-link`.

AWS CLI

Um einen Link zu löschen

Im folgenden `delete-link` Beispiel wird der angegebene Link aus dem angegebenen globalen Netzwerk gelöscht.

```
aws networkmanager delete-link \  
  --global-network-id global-network-01231231231231231 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Link": {  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkArn": "arn:aws:networkmanager::123456789012:link/global-  
network-01231231231231231/link-11112222aaaabbbb1",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "SiteId": "site-444555aaaabbb11223",  
    "Description": "VPN Link",  
    "Type": "broadband",  
    "Bandwidth": {  
      "UploadSpeed": 20,  
      "DownloadSpeed": 20  
    },  
    "Provider": "AnyCompany",  
    "CreatedAt": 1575555811.0,  
    "State": "DELETING"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Links](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [DeleteLink](#) unter AWS CLI Befehlsreferenz.

delete-public-access-block

Das folgende Codebeispiel zeigt die Verwendung `delete-public-access-block`.

AWS CLI

Um die Konfiguration „Öffentlichen Zugriff blockieren“ für einen Bucket zu löschen

Im folgenden `delete-public-access-block` Beispiel wird die Konfiguration „Öffentlicher Zugriff blockieren“ für den angegebenen Bucket entfernt.

```
aws s3api delete-public-access-block \  
  --bucket my-bucket
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

delete-site

Das folgende Codebeispiel zeigt die Verwendung `delete-site`.

AWS CLI

Um eine Site zu löschen

Im folgenden `delete-site` Beispiel wird die angegebene Site (`site-444555aaabbb11223`) im angegebenen globalen Netzwerk gelöscht.

```
aws networkmanager delete-site \  
  --global-network-id global-network-01231231231231231 \  
  --site-id site-444555aaabbb11223 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Site": {  
    "SiteId": "site-444555aaabbb11223",  
    "SiteArn": "arn:aws:networkmanager::123456789012:site/global-  
network-01231231231231231/site-444555aaabbb11223",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "Description": "New York head office",
```

```

    "Location": {
      "Latitude": "40.7128",
      "Longitude": "-74.0060"
    },
    "CreatedAt": 1575554300.0,
    "State": "DELETING"
  }
}

```

Weitere Informationen finden Sie im Transit Gateway Network Manager-Handbuch unter [Working with Sites](#).

- API-Einheiten finden Sie [DeleteSite](#) unter AWS CLI Befehlsreferenz.

deregister-transit-gateway

Das folgende Codebeispiel zeigt die Verwendung `deregister-transit-gateway`.

AWS CLI

Um ein Transit-Gateway von einem globalen Netzwerk abzumelden

Im folgenden `deregister-transit-gateway` Beispiel wird das angegebene Transit-Gateway vom angegebenen globalen Netzwerk abgemeldet.

```

aws networkmanager deregister-transit-gateway \
  --global-network-id global-network-01231231231231231 \
  --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc \
  --region us-west-2

```

Ausgabe:

```

{
  "TransitGatewayRegistration": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
```

```
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Registrierungen](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [Deregister Transit Gateway](#) unter AWS CLI Befehlsreferenz.

describe-global-networks

Das folgende Codebeispiel zeigt die Verwendung `describe-global-networks`.

AWS CLI

Um Ihre globalen Netzwerke zu beschreiben

Das folgende `describe-global-networks` Beispiel beschreibt alle Ihre globalen Netzwerke in Ihrem Konto.

```
aws networkmanager describe-global-networks \  
  --region us-west-2
```

Ausgabe:

```
{  
  "GlobalNetworks": [  
    {  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-  
network/global-network-01231231231231231",  
      "Description": "Company 1 global network",  
      "CreatedAt": 1575553525.0,  
      "State": "AVAILABLE"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribeGlobalNetworks](#) in der AWS CLI Befehlsreferenz.

disassociate-customer-gateway

Das folgende Codebeispiel zeigt die Verwendung `disassociate-customer-gateway`.

AWS CLI

Um die Zuordnung eines Kunden-Gateways aufzuheben

Im folgenden `disassociate-customer-gateway` Beispiel wird die Verbindung zwischen dem angegebenen Kunden-Gateway (`cgw-11223344556677889`) und dem angegebenen globalen Netzwerk getrennt.

```
aws networkmanager disassociate-customer-gateway \
  --global-network-id global-network-01231231231231 \
  --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
  --region us-west-2
```

Ausgabe:

```
{
  "CustomerGatewayAssociation": {
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "State": "DELETING"
  }
}
```

Weitere Informationen finden Sie unter [Customer Gateway Associations](#) im Transit Gateway Network Manager Guide.

- API-Einheiten finden Sie [DisassociateCustomerGateway](#) unter AWS CLI Befehlsreferenz.

disassociate-link

Das folgende Codebeispiel zeigt die Verwendung `disassociate-link`.

AWS CLI

Um die Verknüpfung eines Links aufzuheben

Im folgenden `disassociate-link` Beispiel wird die Verbindung zwischen dem angegebenen Link und dem Gerät `device-07f6fd08867abc123` im angegebenen globalen Netzwerk getrennt.

```
aws networkmanager disassociate-link \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "LinkAssociation": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkAssociationState": "DELETING"  
  }  
}
```

Weitere Informationen finden Sie unter [Geräte- und Verbindungszuordnungen](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [DisassociateLink](#) unter AWS CLI Befehlsreferenz.

get-bucket-analytics-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-bucket-analytics-configuration`.

AWS CLI

Um die Analytics-Konfiguration für einen Bucket mit einer bestimmten ID abzurufen

Im folgenden `get-bucket-analytics-configuration` Beispiel wird die Analytics-Konfiguration für den angegebenen Bucket und die angegebene ID angezeigt.

```
aws s3api get-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Ausgabe:

```
{  
  "AnalyticsConfiguration": {  
    "StorageClassAnalysis": {},  
  }  
}
```

```
    "Id": "1"
  }
}
```

- API-Einheiten finden Sie [GetBucketAnalyticsConfiguration](#) unter AWS CLI Befehlsreferenz.

get-bucket-metrics-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-bucket-metrics-configuration`.

AWS CLI

Um die Metrikkonfiguration für einen Bucket mit einer bestimmten ID abzurufen

Im folgenden `get-bucket-metrics-configuration` Beispiel wird die Metrikkonfiguration für den angegebenen Bucket und die angegebene ID angezeigt.

```
aws s3api get-bucket-metrics-configuration \
  --bucket my-bucket \
  --id 123
```

Ausgabe:

```
{
  "MetricsConfiguration": {
    "Filter": {
      "Prefix": "logs"
    },
    "Id": "123"
  }
}
```

- API-Einheiten finden Sie [GetBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

get-customer-gateway-associations

Das folgende Codebeispiel zeigt die Verwendung `get-customer-gateway-associations`.

AWS CLI

Um die Gateway-Verknüpfungen Ihrer Kunden zu erhalten

Im folgenden `get-customer-gateway-associations` Beispiel werden die Kunden-Gateway-Verknüpfungen für das angegebene globale Netzwerk abgerufen.

```
aws networkmanager get-customer-gateway-associations \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "CustomerGatewayAssociations": [  
    {  
      "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-  
gateway/cgw-11223344556677889",  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "DeviceId": "device-07f6fd08867abc123",  
      "State": "AVAILABLE"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetCustomerGatewayAssociations](#) unter AWS CLI Befehlsreferenz.

get-devices

Das folgende Codebeispiel zeigt die Verwendung `get-devices`.

AWS CLI

Um deine Geräte zu bekommen

Im folgenden `get-devices` Beispiel werden die Geräte im angegebenen globalen Netzwerk abgerufen.

```
aws networkmanager get-devices \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Devices": [  

```

```

    {
      "DeviceId": "device-07f6fd08867abc123",
      "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231231/device-07f6fd08867abc123",
      "GlobalNetworkId": "global-network-01231231231231231",
      "Description": "NY office device",
      "Type": "office device",
      "Vendor": "anycompany",
      "Model": "abcabc",
      "SerialNumber": "1234",
      "CreatedAt": 1575554005.0,
      "State": "AVAILABLE"
    }
  ]
}

```

- API-Einheiten finden Sie [GetDevices](#) in der AWS CLI Befehlsreferenz.

get-link-associations

Das folgende Codebeispiel zeigt die Verwendung `get-link-associations`.

AWS CLI

Um Ihre Linkzuordnungen zu erhalten

Im folgenden `get-link-associations` Beispiel werden die Linkzuordnungen im angegebenen globalen Netzwerk abgerufen.

```

aws networkmanager get-link-associations \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2

```

Ausgabe:

```

{
  "LinkAssociations": [
    {
      "GlobalNetworkId": "global-network-01231231231231231",
      "DeviceId": "device-07f6fd08867abc123",
      "LinkId": "link-11112222aaaabbbb1",
      "LinkAssociationState": "AVAILABLE"
    }
  ]
}

```

```
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetLinkAssociations](#) unter AWS CLI Befehlsreferenz.

get-links

Das folgende Codebeispiel zeigt die Verwendung `get-links`.

AWS CLI

Um deine Links zu bekommen

Im folgenden `get-links` Beispiel werden die Links im angegebenen globalen Netzwerk abgerufen.

```
aws networkmanager get-links \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Links": [  
    {  
      "LinkId": "link-11112222aaaabbbb1",  
      "LinkArn": "arn:aws:networkmanager::123456789012:link/global-  
network-01231231231231231/link-11112222aaaabbbb1",  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "SiteId": "site-444555aaaabbb11223",  
      "Description": "VPN Link",  
      "Type": "broadband",  
      "Bandwidth": {  
        "UploadSpeed": 10,  
        "DownloadSpeed": 20  
      },  
      "Provider": "AnyCompany",  
      "CreatedAt": 1575555811.0,  
      "State": "AVAILABLE"  
    }  
  ]  
}
```

```
}
```

- API-Einheiten finden Sie [GetLinks](#) unter AWS CLI Befehlsreferenz.

get-object-retention

Das folgende Codebeispiel zeigt die Verwendung `get-object-retention`.

AWS CLI

Um die Objektaufbewahrungskonfiguration für ein Objekt abzurufen

Im folgenden `get-object-retention` Beispiel wird die Objektaufbewahrungskonfiguration für das angegebene Objekt abgerufen.

```
aws s3api get-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

Ausgabe:

```
{  
  "Retention": {  
    "Mode": "GOVERNANCE",  
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"  
  }  
}
```

- API-Einheiten finden Sie unter [GetObjectRetention AWS CLI](#) Befehlsreferenz.

get-public-access-block

Das folgende Codebeispiel zeigt die Verwendung `get-public-access-block`.

AWS CLI

Um die Konfiguration für den öffentlichen Zugriff blockieren für einen Bucket festzulegen oder zu ändern

Das folgende `get-public-access-block` Beispiel zeigt die Konfiguration für den blockierten öffentlichen Zugriff für den angegebenen Bucket.

```
aws s3api get-public-access-block --bucket my-bucket
```

Ausgabe:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
    "RestrictPublicBuckets": true
  }
}
```

- API-Einheiten finden Sie [GetPublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

get-sites

Das folgende Codebeispiel zeigt die Verwendung `get-sites`.

AWS CLI

Um deine Websites zu bekommen

Im folgenden `get-sites` Beispiel werden die Websites im angegebenen globalen Netzwerk abgerufen.

```
aws networkmanager get-sites \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2
```

Ausgabe:

```
{
  "Sites": [
    {
      "SiteId": "site-444555aaabbb11223",
      "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
      "GlobalNetworkId": "global-network-01231231231231231",
      "Description": "NY head office",
    }
  ]
}
```

```

        "Location": {
            "Latitude": "40.7128",
            "Longitude": "-74.0060"
        },
        "CreatedAt": 1575554528.0,
        "State": "AVAILABLE"
    }
]
}

```

- API-Einheiten finden Sie [GetSites](#) unter AWS CLI Befehlsreferenz.

get-transit-gateway-registrations

Das folgende Codebeispiel zeigt die Verwendung `get-transit-gateway-registrations`.

AWS CLI

Um Ihre Transit-Gateway-Registrierungen zu erhalten

Im folgenden `get-transit-gateway-registrations` Beispiel werden die Transit-Gateways abgerufen, die im angegebenen globalen Netzwerk registriert sind.

```

aws networkmanager get-transit-gateway-registrations \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2

```

Ausgabe:

```

{
  "TransitGatewayRegistrations": [
    {
      "GlobalNetworkId": "global-network-01231231231231231",
      "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/tgw-123abc05e04123abc",
      "State": {
        "Code": "AVAILABLE"
      }
    }
  ]
}

```

- APIEinzelheiten finden Sie unter [GetTransitGatewayRegistrations AWS CLI Befehlsreferenz](#).

get-vpc-attachment

Das folgende Codebeispiel zeigt die Verwendung `get-vpc-attachment`.

AWS CLI

Um einen VPC Anhang zu erhalten

Im folgenden `get-vpc-attachment` Beispiel werden Informationen zu einer VPC Anlage zurückgegeben.

```
aws networkmanager get-vpc-attachment \  
  --attachment-id attachment-03b7ea450134787da
```

Ausgabe:

```
{  
  "VpcAttachment": {  
    "Attachment": {  
      "CoreNetworkId": "core-network-0522de1b226a5d7b3",  
      "AttachmentId": "attachment-03b7ea450134787da",  
      "OwnerAccountId": "987654321012",  
      "AttachmentType": "VPC",  
      "State": "CREATING",  
      "EdgeLocation": "us-east-1",  
      "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "DevVPC"  
        }  
      ],  
      "CreatedAt": "2022-03-11T17:48:58+00:00",  
      "UpdatedAt": "2022-03-11T17:48:58+00:00"  
    },  
    "SubnetArns": [  
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-202cde6c",  
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-e5022dba",  
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-2387ae02",  
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-cda9dfffc"  
    ]  
  }  
}
```

```
    ],
    "Options": {
      "Ipv6Support": false
    }
  }
}
```

Weitere Informationen finden Sie unter [Anlagen](#) im WANCloud-Benutzerhandbuch.

- API-Einheiten finden Sie [GetVpcAttachment](#) in der AWS CLI Befehlsreferenz.

list-bucket-analytics-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-bucket-analytics-configurations`.

AWS CLI

Um eine Liste von Analytics-Konfigurationen für einen Bucket abzurufen

Im Folgenden wird eine Liste der Analytics-Konfigurationen für den angegebenen Bucket `list-bucket-analytics-configurations` abgerufen.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Ausgabe:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ],
  "IsTruncated": false
}
```

- API-Einheiten finden Sie [ListBucketAnalyticsConfigurations](#) in der AWS CLI Befehlsreferenz.

list-bucket-metrics-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-bucket-metrics-configurations`.

AWS CLI

Um eine Liste von Metrikkonfigurationen für einen Bucket abzurufen

Im folgenden `list-bucket-metrics-configurations` Beispiel wird eine Liste von Metrikkonfigurationen für den angegebenen Bucket abgerufen.

```
aws s3api list-bucket-metrics-configurations \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "IsTruncated": false,  
  "MetricsConfigurationList": [  
    {  
      "Filter": {  
        "Prefix": "logs"  
      },  
      "Id": "123"  
    },  
    {  
      "Filter": {  
        "Prefix": "tmp"  
      },  
      "Id": "234"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [ListBucketMetricsConfigurations AWS CLI Befehlsreferenz](#).

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für die angegebene Gerätesressource (`device-07f6fd08867abc123`) auf.

```
aws networkmanager list-tags-for-resource \  
  --resource-arn arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "TagList": [  
    {  
      "Key": "Network",  
      "Value": "Northeast"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

put-bucket-metrics-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-metrics-configuration`.

AWS CLI

Um eine Metrikkonfiguration für einen Bucket festzulegen

Im folgenden `put-bucket-metrics-configuration` Beispiel wird eine Metrikkonfiguration mit der ID 123 für den angegebenen Bucket festgelegt.

```
aws s3api put-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123 \  
  --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

put-object-retention

Das folgende Codebeispiel zeigt die Verwendung `put-object-retention`.

AWS CLI

Um eine Objektaufbewahrungskonfiguration für ein Objekt festzulegen

Im folgenden `put-object-retention` Beispiel wird eine Objektaufbewahrungskonfiguration für das angegebene Objekt bis zum 01.01.2025 festgelegt.

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter Befehlsreferenz [PutObjectRetention](#).AWS CLI

`put-public-access-block`

Das folgende Codebeispiel zeigt die Verwendung `put-public-access-block`.

AWS CLI

So legen Sie die Konfiguration für den blockierten öffentlichen Zugriff für einen Bucket fest

Im folgenden `put-public-access-block` Beispiel wird eine restriktive Konfiguration für den öffentlichen Blockzugriff für den angegebenen Bucket festgelegt.

```
aws s3api put-public-access-block \  
  --bucket my-bucket \  
  --public-access-block-  
  configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutPublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

`register-transit-gateway`

Das folgende Codebeispiel zeigt die Verwendung `register-transit-gateway`.

AWS CLI

Um ein Transit-Gateway in einem globalen Netzwerk zu registrieren

Im folgenden `register-transit-gateway` Beispiel wird ein Transit-Gateway `tgw-123abc05e04123abc` im angegebenen globalen Netzwerk registriert.

```
aws networkmanager register-transit-gateway \  
  --global-network-id global-network-01231231231231231 \  
  --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc \  
  --region us-west-2
```

Ausgabe:

```
{  
  "TransitGatewayRegistration": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc",  
    "State": {  
      "Code": "PENDING"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Transit Gateway-Registrierungen](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [RegisterTransitGateway](#) unter AWS CLI Befehlsreferenz.

reject-attachment

Das folgende Codebeispiel zeigt die Verwendung `reject-attachment`.

AWS CLI

Um einen Anhang abzulehnen

Im folgenden `reject-attachment` Beispiel wird eine VPC Anhangsanforderung zurückgewiesen.

```
aws networkmanager reject-attachment \  
  --attachment-id attachment-03b7ea450134787da
```

Ausgabe:

```
{
  "Attachment": {
    "CoreNetworkId": "core-network-0522de1b226a5d7b3",
    "AttachmentId": "attachment-03b7ea450134787da",
    "OwnerAccountId": "987654321012",
    "AttachmentType": "VPC",
    "State": "AVAILABLE",
    "EdgeLocation": "us-east-1",
    "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",
    "CreatedAt": "2022-03-11T17:48:58+00:00",
    "UpdatedAt": "2022-03-11T17:51:25+00:00"
  }
}
```

Weitere Informationen finden Sie unter [Annahme von Anhängen](#) im WANCloud-Benutzerhandbuch.

- API-Einheiten finden Sie [RejectAttachment](#) in der AWS CLI Befehlsreferenz.

start-route-analysis

Das folgende Codebeispiel zeigt die Verwendung `start-route-analysis`.

AWS CLI

Um die Routenanalyse zu starten

Im folgenden `start-route-analysis` Beispiel wird die Analyse zwischen einer Quelle und einem Ziel gestartet, einschließlich der optionalen `include-return-path`.

```
aws networkmanager start-route-analysis \
  --global-network-id global-network-00aa0aaa0b0aaa000 \
  --source TransitGatewayAttachmentArn=arn:aws:ec2:us-east-1:503089527312:transit-gateway-attachment/tgw-attach-0d4a2d491bf68c093,IpAddress=10.0.0.0 \
  --destination TransitGatewayAttachmentArn=arn:aws:ec2:us-west-1:503089527312:transit-gateway-attachment/tgw-attach-002577f30bb181742,IpAddress=11.0.0.0 \
  --include-return-path
```

Ausgabe:

```
{
  "RouteAnalysis": {
    "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000",
    "OwnerAccountId": "1111222233333",
    "RouteAnalysisId": "a1873de1-273c-470c-1a2bc2345678",
    "StartTimestamp": 1695760154.0,
    "Status": "RUNNING",
    "Source": {
      "TransitGatewayAttachmentArn": "arn:aws:ec2:us-east-1:111122223333:transit-gateway-attachment/tgw-attach-1234567890abcdef0",
      "TransitGatewayArn": "arn:aws:ec2:us-east-1:111122223333:transit-gateway/tgw-abcdef01234567890",
      "IpAddress": "10.0.0.0"
    },
    "Destination": {
      "TransitGatewayAttachmentArn": "arn:aws:ec2:us-west-1:555555555555:transit-gateway-attachment/tgw-attach-021345abcdef6789",
      "TransitGatewayArn": "arn:aws:ec2:us-west-1:111122223333:transit-gateway/tgw-09876543210fedcba0",
      "IpAddress": "11.0.0.0"
    },
    "IncludeReturnPath": true,
    "UseMiddleboxes": false
  }
}
```

Weitere Informationen finden Sie unter [Route Analyzer](#) im Benutzerhandbuch für AWS globale Netzwerke für Transit Gateways.

- API-Einheiten finden Sie unter [StartRouteAnalysis AWS CLI](#) Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags auf eine Ressource anzuwenden

Im folgenden `tag-resource` Beispiel wird das Tag `Network=Northeast` auf das Gerät `device-07f6fd08867abc123` angewendet.

```
aws networkmanager tag-resource \
```

```
--resource-arn arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123 \  
--tags Key=Network,Value=Northeast \  
--region us-west-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem Schlüssel `Network` vom Gerät entfernt `device-07f6fd08867abc123`.

```
aws networkmanager untag-resource \  
--resource-arn arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123 ]  
--tag-keys Network \  
--region us-west-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-device

Das folgende Codebeispiel zeigt die Verwendung `update-device`.

AWS CLI

Um ein Gerät zu aktualisieren

Im folgenden `update-device` Beispiel `device-07f6fd08867abc123` wird das Gerät aktualisiert, indem eine Site-ID für das Gerät angegeben wird.

```
aws networkmanager update-device \  

```

```
--global-network-id global-network-01231231231231 \  
--device-id device-07f6fd08867abc123 \  
--site-id site-444555aaabbb11223 \  
--region us-west-2
```

Ausgabe:

```
{  
  "Device": {  
    "DeviceId": "device-07f6fd08867abc123",  
    "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "Description": "NY office device",  
    "Type": "Office device",  
    "Vendor": "anycompany",  
    "Model": "abcabc",  
    "SerialNumber": "1234",  
    "SiteId": "site-444555aaabbb11223",  
    "CreatedAt": 1575554005.0,  
    "State": "UPDATING"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Geräten](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [UpdateDevice](#) unter AWS CLI Befehlsreferenz.

update-global-network

Das folgende Codebeispiel zeigt die Verwendung `update-global-network`.

AWS CLI

Um ein globales Netzwerk zu aktualisieren

Im folgenden `update-global-network` Beispiel wird die Beschreibung für das globale Netzwerk aktualisiert `global-network-01231231231231231`.

```
aws networkmanager update-global-network \  
--global-network-id global-network-01231231231231231 \  
--device-id device-07f6fd08867abc123 \  
--site-id site-444555aaabbb11223 \  
--region us-west-2
```



```
--description "Head offices" \  
--region us-west-2
```

Ausgabe:

```
{  
  "GlobalNetwork": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/  
global-network-01231231231231231",  
    "Description": "Head offices",  
    "CreatedAt": 1575553525.0,  
    "State": "UPDATING"  
  }  
}
```

Weitere Informationen finden Sie unter [Global Networks](#) im Transit Gateway Network Manager Guide.

- API-Einheiten finden Sie [UpdateGlobalNetwork](#) in der AWS CLI Befehlsreferenz.

update-link

Das folgende Codebeispiel zeigt die Verwendung `update-link`.

AWS CLI

Um einen Link zu aktualisieren

Im folgenden `update-link` Beispiel werden die Bandbreiteninformationen für den Link `aktualisiertlink-11112222aaaabbbb1`.

```
aws networkmanager update-link \  
  --global-network-id global-network-01231231231231231 \  
  --link-id link-11112222aaaabbbb1 \  
  --bandwidth UploadSpeed=20,DownloadSpeed=20 \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Link": {
```

```

    "LinkId": "link-11112222aaaabbbb1",
    "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
    "GlobalNetworkId": "global-network-01231231231231231",
    "SiteId": "site-444555aaabbb11223",
    "Description": "VPN Link",
    "Type": "broadband",
    "Bandwidth": {
      "UploadSpeed": 20,
      "DownloadSpeed": 20
    },
    "Provider": "AnyCompany",
    "CreatedAt": 1575555811.0,
    "State": "UPDATING"
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Links](#) im Transit Gateway Network Manager-Handbuch.

- API-Einheiten finden Sie [UpdateLink](#) unter AWS CLI Befehlsreferenz.

update-site

Das folgende Codebeispiel zeigt die Verwendung `update-site`.

AWS CLI

Um eine Site zu aktualisieren

Im folgenden `update-site` Beispiel wird die Beschreibung der Site `site-444555aaabbb11223` im angegebenen globalen Netzwerk aktualisiert.

```

aws networkmanager update-site \
  --global-network-id global-network-01231231231231231 \
  --site-id site-444555aaabbb11223 \
  --description "New York Office site" \
  --region us-west-2

```

Ausgabe:

```
{
```

```
"Site": {
  "SiteId": "site-444555aaabbb11223",
  "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
  "GlobalNetworkId": "global-network-01231231231231231",
  "Description": "New York Office site",
  "Location": {
    "Latitude": "40.7128",
    "Longitude": "-74.0060"
  },
  "CreatedAt": 1575554528.0,
  "State": "UPDATING"
}
```

Weitere Informationen finden Sie im Transit Gateway Network Manager-Handbuch unter [Working with Sites](#).

- API-Einheiten finden Sie [UpdateSite](#) unter AWS CLI Befehlsreferenz.

Nimble Studio-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Nimble Studio Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-eula

Das folgende Codebeispiel zeigt die Verwendung `get-eula`.

AWS CLI

Um Informationen über Ihr Studio zu erhalten

Das folgende `get-eula` Beispiel listet die Informationen zu einem aufEULA.

```
aws nimble get-eula \  
  --eula-id "EULAid"
```

Ausgabe:

```
{  
  "eula": {  
    "content": "https://www.mozilla.org/en-US/MPL/2.0/",  
    "createdAt": "2021-04-20T16:45:23+00:00",  
    "eulaId": "gJZLygd-Srq_5NNbSfiaLg",  
    "name": "Mozilla-FireFox",  
    "updatedAt": "2021-04-20T16:45:23+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Accept the EULA](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [GetEula](#) in der AWS CLI Befehlsreferenz.

get-launch-profile-details

Das folgende Codebeispiel zeigt, wie man es benutzt `get-launch-profile-details`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `get-launch-profile-details` Beispiel listet die Details zu einem Startprofil auf.

```
aws nimble get-launch-profile-details \  
  --studio-id "StudioID" \  
  --launch-profile-id "LaunchProfileID"
```

Ausgabe:

```
{
  "launchProfile": {
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q",
    "createdAt": "2022-01-27T21:18:59+00:00",
    "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
    "description": "The Launch Profile for the Render workers created by
StudioBuilder.",
    "ec2SubnetIds": [
      "subnet-EXAMPLE11111"
    ],
    "launchProfileId": "yeG7lDwNQEiwNTRT7DrV7Q",
    "launchProfileProtocolVersions": [
      "2021-03-31"
    ],
    "name": "RenderWorker-Default",
    "state": "READY",
    "statusCode": "LAUNCH_PROFILE_CREATED",
    "statusMessage": "Launch Profile has been created",
    "streamConfiguration": {
      "clipboardMode": "ENABLED",
      "ec2InstanceTypes": [
        "g4dn.4xlarge",
        "g4dn.8xlarge"
      ],
      "maxSessionLengthInMinutes": 690,
      "maxStoppedSessionLengthInMinutes": 0,
      "streamingImageIds": [
        "Cw_jXnp1QcSSXhE2hkNRoQ",
        "YGXAqgoWTnCNSV8VP20sHQ"
      ]
    },
    "studioComponentIds": [
      "_hR_-RaAReS0jAnLakbX7Q",
      "vQ5w_TbIRayPkAZgcbyYRA",
      "ZQuMxN99Qfa_Js6ma9TwdA",
      "45Kj0SPPRzK20yvpCuQ6qw"
    ],
    "tags": {
      "resourceArn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q"
    }
  },
}
```

```
"updatedAt": "2022-01-27T21:19:13+00:00",
"updatedBy": "AROA3002NEHCCYRNDDIFT:i-00b98256b04d9e989",
"validationResults": [
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
  },
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_SUBNET_ASSOCIATION"
  },
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
  },
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
  }
]
},
"streamingImages": [
  {
    "arn": "arn:aws:nimble:us-west-2:123456789012:streaming-image/
Cw_jXnp1QcSSXhE2hkNRoQ",
    "description": "Base windows image for NimbleStudio",
    "ec2ImageId": "ami-EXAMPLE11111",
    "eulaIds": [
      "gJZLygd-Srq_5NNbSfiaLg",
      "ggK2eIw6RQyt8PIee0lD3g",
      "a-D9Wc0VQCKUfxAinCDxaw",
      "RvoNmVXiSrS4LhLTb6ybkw",
      "wtp85BcSTa2NZeNRnMKdjw",
      "Rl-J0fM5S12hyIiwWIV6hw"
    ],
    "name": "NimbleStudioWindowsStreamImage",
```

```

    "owner": "amazon",
    "platform": "WINDOWS",
    "state": "READY",
    "streamingImageId": "Cw_jXnp1QcSSXhE2hkNRoQ",
    "tags": {
      "resourceArn": "arn:aws:nimble:us-west-2:123456789012:streaming-
image/Cw_jXnp1QcSSXhE2hkNRoQ"
    }
  },
  {
    "arn": "arn:aws:nimble:us-west-2:123456789012:streaming-image/
YGXAqgoWTnCNSV8VP20sHQ",
    "description": "Base linux image for NimbleStudio",
    "ec2ImageId": "ami-EXAMPLE11111",
    "eulaIds": [
      "gJZLygd-Srq_5NNbSfiaLg",
      "ggK2eIw6RQyt8PIee01D3g",
      "a-D9Wc0VQCKUfxAinCDxaw",
      "RvoNmVXiSrS4LhLTb6ybkw",
      "wtp85BcSTa2NZeNRnMKdjw",
      "R1-J0fM5S12hyIiwWIV6hw"
    ],
    "name": "NimbleStudioLinuxStreamImage",
    "owner": "amazon",
    "platform": "LINUX",
    "state": "READY",
    "streamingImageId": "YGXAqgoWTnCNSV8VP20sHQ",
    "tags": {
      "resourceArn": "arn:aws:nimble:us-west-2:123456789012:streaming-
image/YGXAqgoWTnCNSV8VP20sHQ"
    }
  }
],
"studioComponentSummaries": [
  {
    "description": "FSx for Windows",
    "name": "FSxWindows",
    "studioComponentId": "ZQuMxN99Qfa Js6ma9TwdA",
    "subtype": "AMAZON_FSX_FOR_WINDOWS",
    "type": "SHARED_FILE_SYSTEM"
  },
  {
    "description": "Instance configuration studio component.",
    "name": "InstanceConfiguration",

```

```

        "studioComponentId": "vQ5w_TbIRayPkAZgcbyYRA",
        "subtype": "CUSTOM",
        "type": "CUSTOM"
    },
    {
        "name": "ActiveDirectory",
        "studioComponentId": "_hR_-RaAReS0jAnLakbX7Q",
        "subtype": "AWS_MANAGED_MICROSOFT_AD",
        "type": "ACTIVE_DIRECTORY"
    },
    {
        "description": "Render farm running Deadline",
        "name": "RenderFarm",
        "studioComponentId": "45Kj0SPPRzK20yvpCuQ6qw",
        "subtype": "CUSTOM",
        "type": "COMPUTE_FARM"
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen von Startprofilen](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [GetLaunchProfileDetails](#) in der AWS CLI Befehlsreferenz.

get-launch-profile

Das folgende Codebeispiel zeigt, wie man es benutzt `get-launch-profile`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `get-launch-profile` Beispiel listet Informationen zu einem Startprofil auf.

```

aws nimble get-launch-profile \
  --studio-id "StudioID" \
  --launch-profile-id "LaunchProfileID"

```

Ausgabe:

```

{
  "launchProfile": {

```



```
"arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q",
  "createdAt": "2022-01-27T21:18:59+00:00",
  "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
  "description": "The Launch Profile for the Render workers created by
StudioBuilder.",
  "ec2SubnetIds": [
    "subnet-EXAMPLE11111"
  ],
  "launchProfileId": "yeG7lDwNQEiwNTRT7DrV7Q",
  "launchProfileProtocolVersions": [
    "2021-03-31"
  ],
  "name": "RenderWorker-Default",
  "state": "READY",
  "statusCode": "LAUNCH_PROFILE_CREATED",
  "statusMessage": "Launch Profile has been created",
  "streamConfiguration": {
    "clipboardMode": "ENABLED",
    "ec2InstanceTypes": [
      "g4dn.4xlarge",
      "g4dn.8xlarge"
    ],
    "maxSessionLengthInMinutes": 690,
    "maxStoppedSessionLengthInMinutes": 0,
    "streamingImageIds": [
      "Cw_jXnp1QcSSXhE2hkNRoQ",
      "YGXAqgoWTnCNSV8VP20sHQ"
    ]
  },
  "studioComponentIds": [
    "_hR_-RaAReS0jAnLakbX7Q",
    "vQ5w_TbIRayPkAZgcbyYRA",
    "ZQuMxN99Qfa_Js6ma9TwdA",
    "45Kj0SPPrzK20yvpCuQ6qw"
  ],
  "tags": {},
  "updatedAt": "2022-01-27T21:19:13+00:00",
  "updatedBy": "AROA3002NEHCCYRNDDIFT:i-00b98256b04d9e989",
  "validationResults": [
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
    }
  ]
}
```

```

        "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
    },
    {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SUBNET_ASSOCIATION"
    },
    {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
    },
    {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
    }
]
}
}

```

Weitere Informationen finden Sie unter [Erstellen von Startprofilen](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [GetLaunchProfile](#) in der AWS CLI Befehlsreferenz.

get-studio

Das folgende Codebeispiel zeigt, wie man es benutzt `get-studio`.

AWS CLI

Um Informationen über Ihr Studio zu erhalten

Das folgende `get-studio` Beispiel listet die Studios in Ihrem AWS Konto auf.

```
aws nimble get-studio \
  --studio-id "StudioID"
```

Ausgabe:

```
{
  "studio": {
    "adminRoleArn": "arn:aws:iam::123456789012:role/studio-admin-role",
    "arn": "arn:aws:nimble:us-west-2:123456789012:studio/stid-EXAMPLE11111",
    "createdAt": "2022-01-27T20:29:35+00:00",
    "displayName": "studio-name",
    "homeRegion": "us-west-2",
    "ssoClientId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "state": "READY",
    "statusCode": "STUDIO_CREATED",
    "statusMessage": "The studio has been created successfully ",
    "studioEncryptionConfiguration": {
      "keyType": "AWS_OWNED_KEY"
    },
    "studioId": "us-west-2:stid-EXAMPLE11111",
    "studioName": "studio-name",
    "studioUrl": "https://studio-name.nimblestudio.us-west-2.amazonaws.com",
    "tags": {},
    "updatedAt": "2022-01-27T20:29:37+00:00",
    "userRoleArn": "arn:aws:iam::123456789012:role/studio-user-role"
  }
}
```

Weitere Informationen finden Sie unter [Was ist Amazon Nimble Studio?](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [GetStudio](#) in der AWS CLI Befehlsreferenz.

list-eula-acceptances

Das folgende Codebeispiel zeigt, wie man es benutzt `list-eula-acceptances`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-eula-acceptances` Beispiel listet die EULAs in Ihrem AWS Konto akzeptierten Dateien auf.

```
aws nimble list-eula-acceptances \
  --studio-id "StudioID"
```

Ausgabe:

```
{
  "eulaAcceptances": [
    {
      "acceptedAt": "2022-01-28T17:44:35+00:00",
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
      "eulaAcceptanceId": "V0JlpZQaSx6yHcUuX0qfQw",
      "eulaId": "R1-J0fM5S12hyIiwWIV6hw"
    },
    {
      "acceptedAt": "2022-01-28T17:44:35+00:00",
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
      "eulaAcceptanceId": "YY_uDFW-SVibc627qbug0Q",
      "eulaId": "RvoNmVXiSrS4LhLTb6ybkw"
    },
    {
      "acceptedAt": "2022-01-28T17:44:35+00:00",
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
      "eulaAcceptanceId": "ov087PnhQ4-MpttiL5uN6Q",
      "eulaId": "a-D9Wc0VQCKUfxAinCDxaw"
    },
    {
      "acceptedAt": "2022-01-28T17:44:35+00:00",
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
      "eulaAcceptanceId": "5YeXje4yR0amuTESGvqIAQ",
      "eulaId": "gJZLygd-Srq_5NNbSfiaLg"
    },
    {
      "acceptedAt": "2022-01-28T17:44:35+00:00",
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
      "eulaAcceptanceId": "W1sIn8PtScqeJEn8sxxhgw",
      "eulaId": "ggK2eIw6RQyt8PIee01D3g"
    },
    {
      "acceptedAt": "2022-01-28T17:44:35+00:00",
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
      "eulaAcceptanceId": "Zq9KNEQPRMWJ7Fo1SoQgUA",

```

```

        "eulaId": "wtp85BcSTa2NZeNRnMKdjw"
      }
    ]
  }

```

Weitere Informationen finden Sie unter [Accept the EULA](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [ListEulaAcceptances](#) in der AWS CLI Befehlsreferenz.

list-eulas

Das folgende Codebeispiel zeigt, wie man es benutzt `list-eulas`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-eulas` Beispiel listet die EULAs in Ihrem AWS Konto auf.

```
aws nimble list-eulas
```

Ausgabe:

```

{
  "eulas": [
    {
      "content": "https://www.mozilla.org/en-US/MPL/2.0/",
      "createdAt": "2021-04-20T16:45:23+00:00",
      "eulaId": "gJZLygd-Srq_5NNbSfiaLg",
      "name": "Mozilla-FireFox",
      "updatedAt": "2021-04-20T16:45:23+00:00"
    },
    {
      "content": "https://www.awsthinkbox.com/end-user-license-agreement",
      "createdAt": "2021-04-20T16:45:24+00:00",
      "eulaId": "RvoNmVXiSrS4LhLTb6ybkw",
      "name": "Thinkbox-Deadline",
      "updatedAt": "2021-04-20T16:45:24+00:00"
    },
    {
      "content": "https://www.videolan.org/legal.html",
      "createdAt": "2021-04-20T16:45:24+00:00",

```

```
    "eulaId": "R1-J0fM5S12hyIiwWIV6hw",
    "name": "Videolan-VLC",
    "updatedAt": "2021-04-20T16:45:24+00:00"
  },
  {
    "content": "https://code.visualstudio.com/license",
    "createdAt": "2021-04-20T16:45:23+00:00",
    "eulaId": "ggK2eIw6RQyt8PIee0lD3g",
    "name": "Microsoft-VSCode",
    "updatedAt": "2021-04-20T16:45:23+00:00"
  },
  {
    "content": "https://darbyjohnston.github.io/DJV/legal.html#License",
    "createdAt": "2021-04-20T16:45:23+00:00",
    "eulaId": "wtp85BcSTa2NZeNRnMKdjw",
    "name": "DJV-DJV",
    "updatedAt": "2021-04-20T16:45:23+00:00"
  },
  {
    "content": "https://www.sidefx.com/legal/license-agreement/",
    "createdAt": "2021-04-20T16:45:24+00:00",
    "eulaId": "uu2VDLo-QJeIGWwLBae_UA",
    "name": "SideFX-Houdini",
    "updatedAt": "2021-04-20T16:45:24+00:00"
  },
  {
    "content": "https://www.chaosgroup.com/eula",
    "createdAt": "2021-04-20T16:45:23+00:00",
    "eulaId": "L0HS4P3CRYKVXc2J2L07Vw",
    "name": "ChaosGroup-Vray",
    "updatedAt": "2021-04-20T16:45:23+00:00"
  },
  {
    "content": "https://www.foundry.com/eula",
    "createdAt": "2021-04-20T16:45:23+00:00",
    "eulaId": "SAuhfHmmsAeUuq3wsMiMlw",
    "name": "Foundry-Nuke",
    "updatedAt": "2021-04-20T16:45:23+00:00"
  },
  {
    "content": "https://download.blender.org/release/GPL3-license.txt",
    "createdAt": "2021-04-20T16:45:23+00:00",
    "eulaId": "a-D9Wc0VQCKUfxAinCDxaw",
    "name": "BlenderFoundation-Blender",
```

```

    "updatedAt": "2021-04-20T16:45:23+00:00"
  }
]
}

```

Weitere Informationen finden Sie unter [Accept the EULA](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [ListEulas](#) in der AWS CLI Befehlsreferenz.

list-launch-profiles

Das folgende Codebeispiel zeigt, wie man es benutzt `list-launch-profiles`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-launch-profiles` Beispiel listet die Startprofile in Ihrem AWS Konto auf.

```

aws nimble list-launch-profiles \
  --studio-id "StudioID"

```

Ausgabe:

```

{
  "launchProfiles": [
    {
      "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG71DwNQEiwNTRT7DrV7Q",
      "createdAt": "2022-01-27T21:18:59+00:00",
      "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
      "description": "The Launch Profile for the Render workers created by
StudioBuilder.",
      "ec2SubnetIds": [
        "subnet-EXAMPLE11111"
      ],
      "launchProfileId": "yeG71DwNQEiwNTRT7DrV7Q",
      "launchProfileProtocolVersions": [
        "2021-03-31"
      ],
      "name": "RenderWorker-Default",
      "state": "READY",
    }
  ]
}

```

```
"statusCode": "LAUNCH_PROFILE_CREATED",
"statusMessage": "Launch Profile has been created",
"streamConfiguration": {
  "clipboardMode": "ENABLED",
  "ec2InstanceTypes": [
    "g4dn.4xlarge",
    "g4dn.8xlarge"
  ],
  "maxSessionLengthInMinutes": 690,
  "maxStoppedSessionLengthInMinutes": 0,
  "streamingImageIds": [
    "Cw_jXnp1QcSSXhE2hkNRoQ",
    "YGXAqgoWTnCNSV8VP20sHQ"
  ]
},
"studioComponentIds": [
  "_hR_-RaAReS0jAnLakbX7Q",
  "vQ5w_TbIRayPkAZgcbyYRA",
  "ZQuMxN99Qfa_Js6ma9TwdA",
  "45Kj0SPPRzK20yvpCuQ6qw"
],
"tags": {},
"updatedAt": "2022-01-27T21:19:13+00:00",
"updatedBy": "AROA3002NEHCCYRNDIIFT:i-EXAMPLE11111",
"validationResults": [
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
  },
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_SUBNET_ASSOCIATION"
  },
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
  },
  {
```



```

        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
    }
]
},
{
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
jDCIm1jRSaa9e44PZ3w7gg",
    "createdAt": "2022-01-27T21:19:26+00:00",
    "createdBy": "AROA3002NEHCCYRNDIIFT:i-EXAMPLE11111",
    "description": "This Workstation Launch Profile was created by
StudioBuilder",
    "ec2SubnetIds": [
        "subnet-046f4205ae535b2cc"
    ],
    "launchProfileId": "jDCIm1jRSaa9e44PZ3w7gg",
    "launchProfileProtocolVersions": [
        "2021-03-31"
    ],
    "name": "Workstation-Default",
    "state": "READY",
    "statusCode": "LAUNCH_PROFILE_CREATED",
    "statusMessage": "Launch Profile has been created",
    "streamConfiguration": {
        "clipboardMode": "ENABLED",
        "ec2InstanceTypes": [
            "g4dn.4xlarge",
            "g4dn.8xlarge"
        ],
        "maxSessionLengthInMinutes": 690,
        "maxStoppedSessionLengthInMinutes": 0,
        "streamingImageIds": [
            "Cw_jXnp1QcSSXhE2hkNRoQ",
            "YGXAqgoWTnCNsv8VP20sHQ"
        ]
    },
    "studioComponentIds": [
        "_hR_-RaAReS0jAnLakbX7Q",
        "vQ5w_TbIRayPkAZgcbyYRA",
        "ZQuMxN99Qfa_Js6ma9TwdA",
        "yJSbsHXAQYwk9FXLNusX1Q",
        "45Kj0SPPrzK20yvpCuQ6qw"
    ]
}

```

```

    ],
    "tags": {},
    "updatedAt": "2022-01-27T21:19:40+00:00",
    "updatedBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
    "validationResults": [
      {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
      },
      {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SUBNET_ASSOCIATION"
      },
      {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
      },
      {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Erstellen von Startprofilen](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [ListLaunchProfiles](#) in der AWS CLI Befehlsreferenz.

list-studio-components

Das folgende Codebeispiel zeigt, wie man es benutzt `list-studio-components`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-studio-components` Beispiel listet die Studio-Komponenten in Ihrem AWS Konto auf.

```
aws nimble list-studio-components \
  --studio-id "StudioID"
```

Ausgabe:

```
{
  "studioComponents": [
    {
      "arn": "arn:aws:nimble:us-west-2:123456789012:studio-component/
ZQuMxN99Qfa_Js6ma9TwdA",
      "configuration": {
        "sharedFileSystemConfiguration": {
          "fileSystemId": "fs-EXAMPLE11111",
          "linuxMountPoint": "/mnt/fsxshare",
          "shareName": "share",
          "windowsMountDrive": "Z"
        }
      },
      "createdAt": "2022-01-27T21:15:34+00:00",
      "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
      "description": "FSx for Windows",
      "ec2SecurityGroupIds": [
        "sg-EXAMPLE11111"
      ],
      "name": "FSxWindows",
      "state": "READY",
      "statusCode": "STUDIO_COMPONENT_CREATED",
      "statusMessage": "Studio Component has been created",
      "studioComponentId": "ZQuMxN99Qfa_Js6ma9TwdA",
      "subtype": "AMAZON_FSX_FOR_WINDOWS",
      "tags": {},
      "type": "SHARED_FILE_SYSTEM",
      "updatedAt": "2022-01-27T21:15:35+00:00",
      "updatedBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111"
    },
    ...
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Wie StudioBuilder funktioniert mit Amazon Nimble Studio](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStudioComponents](#) in der AWS CLI Befehlsreferenz.

list-studio-members

Das folgende Codebeispiel zeigt, wie man es benutzt `list-studio-members`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-studio-members` Beispiel listet die verfügbaren Studio-Mitglieder in Ihrem AWS Konto auf.

```
aws nimble list-studio-members \  
  --studio-id "StudioID"
```

Ausgabe:

```
{  
  "members": [  
    {  
      "identityStoreId": "d-EXAMPLE11111",  
      "persona": "ADMINISTRATOR",  
      "principalId": "EXAMPLE11111-e9fd012a-94ad-4f16-9866-c69a63ab6486"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Hinzufügen von Studio-Benutzern](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStudioMembers](#) in der AWS CLI Befehlsreferenz.

list-studios

Das folgende Codebeispiel zeigt, wie man es benutzt `list-studios`.

AWS CLI

Um deine Studios aufzulisten

Das folgende `list-studios` Beispiel listet die Studios in Ihrem AWS Konto auf.

```
aws nimble list-studios
```

Ausgabe:

```
{
  "studios": [
    {
      "adminRoleArn": "arn:aws:iam::123456789012:role/studio-admin-role",
      "arn": "arn:aws:nimble:us-west-2:123456789012:studio/studio-id",
      "createdAt": "2022-01-27T20:29:35+00:00",
      "displayName": "studio-name",
      "homeRegion": "us-west-2",
      "ssoClientId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "state": "READY",
      "statusCode": "STUDIO_CREATED",
      "statusMessage": "The studio has been created successfully ",
      "studioEncryptionConfiguration": {
        "keyType": "AWS_OWNED_KEY"
      },
      "studioId": "us-west-2:studio-id",
      "studioName": "studio-name",
      "studioUrl": "https://studio-name.nimblestudio.us-west-2.amazonaws.com",
      "tags": {},
      "updatedAt": "2022-01-27T20:29:37+00:00",
      "userRoleArn": "arn:aws:iam::123456789012:role/studio-user-role"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Was ist Amazon Nimble Studio?](#) im Amazon Nimble Studio-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStudios](#) in der AWS CLI Befehlsreferenz.

OpenSearch Servicebeispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with OpenSearch Service Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-elasticsearch-domain

Das folgende Codebeispiel zeigt die Verwendung `create-elasticsearch-domain`.

AWS CLI

So erstellen Sie eine Amazon Elasticsearch Service Service-Domain

Der folgende `create-elasticsearch-domain` Befehl erstellt eine neue Amazon Elasticsearch Service Service-Domain innerhalb eines VPC und beschränkt den Zugriff auf einen einzelnen Benutzer. Amazon ES leitet die VPC ID aus dem angegebenen Subnetz und der angegebenen Sicherheitsgruppe ab. IDs

```
aws es create-elasticsearch-domain \
  --domain-name vpc-cli-example \
  --elasticsearch-version 6.2 \
  --elasticsearch-cluster-
config InstanceType=m4.large.elasticsearch,InstanceCount=1 \
  --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 \
  --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect":
"Allow", "Principal": { "AWS": "arn:aws:iam::123456789012:root" }, "Action": "es:*",
"Resource": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*" } ] }' \
  --vpc-options SubnetIds=subnet-1a2a3a4a,SecurityGroupIds=sg-2a3a4a5a
```

Ausgabe:

```
{
  "DomainStatus": {
    "ElasticsearchClusterConfig": {
      "DedicatedMasterEnabled": false,
      "InstanceCount": 1,
      "ZoneAwarenessEnabled": false,
      "InstanceType": "m4.large.elasticsearch"
    },
    "DomainId": "123456789012/vpc-cli-example",
    "CognitoOptions": {
      "Enabled": false
    },
    "VPCOptions": {
      "SubnetIds": [
        "subnet-1a2a3a4a"
      ],
      "VPCId": "vpc-3a4a5a6a",
      "SecurityGroupIds": [
        "sg-2a3a4a5a"
      ],
      "AvailabilityZones": [
        "us-west-1c"
      ]
    },
    "Created": true,
    "Deleted": false,
    "EBSOptions": {
      "VolumeSize": 10,
      "VolumeType": "standard",
      "EBSEnabled": true
    },
    "Processing": true,
    "DomainName": "vpc-cli-example",
    "SnapshotOptions": {
      "AutomatedSnapshotStartHour": 0
    },
    "ElasticsearchVersion": "6.2",
    "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*\"}]}",
    "AdvancedOptions": {
```

```
        "rest.action.multi.allow_explicit_index": "true"
    },
    "EncryptionAtRestOptions": {
        "Enabled": false
    },
    "ARN": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example"
}
}
```

Weitere Informationen finden Sie unter [Creating and Managing Amazon Elasticsearch Service Domains](#) im Amazon Elasticsearch Service Developer Guide.

- API-Einheiten finden Sie [CreateElasticsearchDomain](#) in der AWS CLI Befehlsreferenz.

describe-elasticsearch-domain-config

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-elasticsearch-domain-config`.

AWS CLI

Um Details zur Domain-Konfiguration abzurufen

Das folgende `describe-elasticsearch-domain-config` Beispiel enthält Konfigurationsdetails für eine bestimmte Domäne sowie Statusinformationen für jede einzelne Domänenkomponente.

```
aws es describe-elasticsearch-domain-config \
  --domain-name cli-example
```

Ausgabe:

```
{
  "DomainConfig": {
    "ElasticsearchVersion": {
      "Options": "7.4",
      "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
      }
    }
  }
}
```



```
    }
  },
  "ElasticsearchClusterConfig": {
    "Options": {
      "InstanceType": "c5.large.elasticsearch",
      "InstanceCount": 1,
      "DedicatedMasterEnabled": true,
      "ZoneAwarenessEnabled": false,
      "DedicatedMasterType": "c5.large.elasticsearch",
      "DedicatedMasterCount": 3,
      "WarmEnabled": true,
      "WarmType": "ultrawarm1.medium.elasticsearch",
      "WarmCount": 2
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "EBSOptions": {
    "Options": {
      "EBSEnabled": true,
      "VolumeType": "gp2",
      "VolumeSize": 10
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "AccessPolicies": {
    "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
```

```
        "State": "Active",
        "PendingDeletion": false
    }
},
"SnapshotOptions": {
    "Options": {
        "AutomatedSnapshotStartHour": 0
    },
    "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
    }
},
"VPCOptions": {
    "Options": {},
    "Status": {
        "CreationDate": 1591210426.162,
        "UpdateDate": 1591210426.162,
        "UpdateVersion": 18,
        "State": "Active",
        "PendingDeletion": false
    }
},
"CognitoOptions": {
    "Options": {
        "Enabled": false
    },
    "Status": {
        "CreationDate": 1591210426.163,
        "UpdateDate": 1591210426.163,
        "UpdateVersion": 18,
        "State": "Active",
        "PendingDeletion": false
    }
},
"EncryptionAtRestOptions": {
    "Options": {
        "Enabled": true,
        "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
    }
},
```

```
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "NodeToNodeEncryptionOptions": {
    "Options": {
      "Enabled": true
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "AdvancedOptions": {
    "Options": {
      "rest.action.multi.allow_explicit_index": "true"
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "LogPublishingOptions": {
    "Options": {},
    "Status": {
      "CreationDate": 1591210426.164,
      "UpdateDate": 1591210426.164,
      "UpdateVersion": 18,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "DomainEndpointOptions": {
    "Options": {
```

```

        "EnforceHTTPS": true,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
    }
},
"AdvancedSecurityOptions": {
    "Options": {
        "Enabled": true,
        "InternalUserDatabaseEnabled": true
    },
    "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589827485.577,
        "UpdateVersion": 14,
        "State": "Active",
        "PendingDeletion": false
    }
}
}
}
}

```

Weitere Informationen finden Sie unter [Creating and Managing Amazon Elasticsearch Service Domains](#) im Amazon Elasticsearch Service Developer Guide.

- API-Einheiten finden Sie [DescribeElasticsearchDomainConfig](#) in der AWS CLI Befehlsreferenz.

describe-elasticsearch-domain

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-elasticsearch-domain`.

AWS CLI

Um Details für eine einzelne Domain abzurufen

Das folgende `describe-elasticsearch-domain` Beispiel enthält Konfigurationsdetails für eine bestimmte Domäne.

```
aws es describe-elasticsearch-domain \  
  --domain-name cli-example
```

Ausgabe:

```
{  
  "DomainStatus": {  
    "DomainId": "123456789012/cli-example",  
    "DomainName": "cli-example",  
    "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example",  
    "Created": true,  
    "Deleted": false,  
    "Endpoint": "search-cli-example-1a2a3a4a5a6a7a8a9a0a.us-  
east-1.es.amazonaws.com",  
    "Processing": false,  
    "UpgradeProcessing": false,  
    "ElasticsearchVersion": "7.4",  
    "ElasticsearchClusterConfig": {  
      "InstanceType": "c5.large.elasticsearch",  
      "InstanceCount": 1,  
      "DedicatedMasterEnabled": true,  
      "ZoneAwarenessEnabled": false,  
      "DedicatedMasterType": "c5.large.elasticsearch",  
      "DedicatedMasterCount": 3,  
      "WarmEnabled": true,  
      "WarmType": "ultrawarm1.medium.elasticsearch",  
      "WarmCount": 2  
    },  
    "EBSOptions": {  
      "EBSEnabled": true,  
      "VolumeType": "gp2",  
      "VolumeSize": 10  
    },  
    "AccessPolicies": "{\n\"Version\": \"2012-10-17\", \"Statement\": [\n{\n\"Effect\": \"Allow\", \"Principal\": {\n\"AWS\": \"*\"}, \"Action\": \"es:*\", \"Resource\":\n\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]]}",  
    "SnapshotOptions": {  
      "AutomatedSnapshotStartHour": 0  
    },  
    "CognitoOptions": {  
      "Enabled": false  
    },  
    "EncryptionAtRestOptions": {
```

```
    "Enabled": true,
    "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "AdvancedOptions": {
    "rest.action.multi.allow_explicit_index": "true"
  },
  "ServiceSoftwareOptions": {
    "CurrentVersion": "R20200522",
    "NewVersion": "",
    "UpdateAvailable": false,
    "Cancelable": false,
    "UpdateStatus": "COMPLETED",
    "Description": "There is no software update available for this domain.",
    "AutomatedUpdateDate": 0.0
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
  },
  "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true
  }
}
```

Weitere Informationen finden Sie unter [Creating and Managing Amazon Elasticsearch Service Domains](#) im Amazon Elasticsearch Service Developer Guide.

- API-Einheiten finden Sie [DescribeElasticsearchDomain](#) in der AWS CLI Befehlsreferenz.

describe-elasticsearch-domains

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-elasticsearch-domains`.

AWS CLI

Um Details für eine oder mehrere Domains abzurufen

Das folgende `describe-elasticsearch-domains` Beispiel enthält Konfigurationsdetails für eine oder mehrere Domänen.

```
aws es describe-elasticsearch-domains \  
  --domain-names cli-example-1 cli-example-2
```

Ausgabe:

```
{  
  "DomainStatusList": [{  
    "DomainId": "123456789012/cli-example-1",  
    "DomainName": "cli-example-1",  
    "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-1",  
    "Created": true,  
    "Deleted": false,  
    "Endpoint": "search-cli-example-1-1a2a3a4a5a6a7a8a9a0a.us-  
east-1.es.amazonaws.com",  
    "Processing": false,  
    "UpgradeProcessing": false,  
    "ElasticsearchVersion": "7.4",  
    "ElasticsearchClusterConfig": {  
      "InstanceType": "c5.large.elasticsearch",  
      "InstanceCount": 1,  
      "DedicatedMasterEnabled": true,  
      "ZoneAwarenessEnabled": false,  
      "DedicatedMasterType": "c5.large.elasticsearch",  
      "DedicatedMasterCount": 3,  
      "WarmEnabled": true,  
      "WarmType": "ultrawarm1.medium.elasticsearch",  
      "WarmCount": 2  
    },  
    "EBSOptions": {  
      "EBSEnabled": true,  
      "VolumeType": "gp2",  
      "VolumeSize": 10  
    },  
    "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":"  
\": \"Allow\", \"Principal\": {\"AWS\": \"*\"}, \"Action\": \"es:*\", \"Resource\":  
\": \"arn:aws:es:us-east-1:123456789012:domain/cli-example-1/*\"}]}",  
    "SnapshotOptions": {  
      "AutomatedSnapshotStartHour": 0  
    },  
    "CognitoOptions": {
```

```
        "Enabled": false
    },
    "EncryptionAtRestOptions": {
        "Enabled": true,
        "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
    },
    "NodeToNodeEncryptionOptions": {
        "Enabled": true
    },
    "AdvancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
    },
    "ServiceSoftwareOptions": {
        "CurrentVersion": "R20200522",
        "NewVersion": "",
        "UpdateAvailable": false,
        "Cancellable": false,
        "UpdateStatus": "COMPLETED",
        "Description": "There is no software update available for this
domain.",
        "AutomatedUpdateDate": 0.0
    },
    "DomainEndpointOptions": {
        "EnforceHTTPS": true,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "AdvancedSecurityOptions": {
        "Enabled": true,
        "InternalUserDatabaseEnabled": true
    }
},
{
    "DomainId": "123456789012/cli-example-2",
    "DomainName": "cli-example-2",
    "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-2",
    "Created": true,
    "Deleted": false,
    "Processing": true,
    "UpgradeProcessing": false,
    "ElasticsearchVersion": "7.4",
    "ElasticsearchClusterConfig": {
        "InstanceType": "r5.large.elasticsearch",
        "InstanceCount": 1,
```



```
        "DedicatedMasterEnabled": false,
        "ZoneAwarenessEnabled": false,
        "WarmEnabled": false
    },
    "EBSOptions": {
        "EBSEnabled": true,
        "VolumeType": "gp2",
        "VolumeSize": 10
    },
    "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Deny\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/cli-example-2/*\"}]}",
    "SnapshotOptions": {
        "AutomatedSnapshotStartHour": 0
    },
    "CognitoOptions": {
        "Enabled": false
    },
    "EncryptionAtRestOptions": {
        "Enabled": false
    },
    "NodeToNodeEncryptionOptions": {
        "Enabled": false
    },
    "AdvancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
    },
    "ServiceSoftwareOptions": {
        "CurrentVersion": "",
        "NewVersion": "",
        "UpdateAvailable": false,
        "Cancellable": false,
        "UpdateStatus": "COMPLETED",
        "Description": "There is no software update available for this
domain.",
        "AutomatedUpdateDate": 0.0
    },
    "DomainEndpointOptions": {
        "EnforceHTTPS": false,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "AdvancedSecurityOptions": {
        "Enabled": false,
        "InternalUserDatabaseEnabled": false
    }
```

```

    }
  }
]
}

```

Weitere Informationen finden Sie unter [Creating and Managing Amazon Elasticsearch Service Domains](#) im Amazon Elasticsearch Service Developer Guide.

- API-Einheiten finden Sie [DescribeElasticsearchDomains](#) in der AWS CLI Befehlsreferenz.

describe-reserved-elasticsearch-instances

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-reserved-elasticsearch-instances`.

AWS CLI

Um alle Reserved Instances anzuzeigen

Das folgende `describe-elasticsearch-domains` Beispiel bietet eine Zusammenfassung aller Instances, die Sie in einer Region reserviert haben.

```
aws es describe-reserved-elasticsearch-instances
```

Ausgabe:

```

{
  "ReservedElasticsearchInstances": [{
    "FixedPrice": 100.0,
    "ReservedElasticsearchInstanceOfferingId":
"1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
    "ReservationName": "my-reservation",
    "PaymentOption": "PARTIAL_UPFRONT",
    "UsagePrice": 0.0,
    "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
    "RecurringCharges": [{
      "RecurringChargeAmount": 0.603,
      "RecurringChargeFrequency": "Hourly"
    }],
    "State": "payment-pending",
    "StartTime": 1522872571.229,
    "ElasticsearchInstanceCount": 3,
  }],
}

```

```
    "Duration": 31536000,  
    "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",  
    "CurrencyCode": "USD"  
  }  
}
```

Weitere Informationen finden Sie unter [Reserved Instances](#) im Amazon Elasticsearch Service Developer Guide.

- API-Einheiten finden Sie [DescribeReservedElasticsearchInstances](#) in der AWS CLI Befehlsreferenz.

list-domain-names

Das folgende Codebeispiel zeigt, wie man es benutzt `list-domain-names`.

AWS CLI

Um alle Domänen aufzulisten

Das folgende `list-domain-names` Beispiel bietet eine kurze Zusammenfassung aller Domänen in der Region.

```
aws es list-domain-names
```

Ausgabe:

```
{  
  "DomainNames": [{  
    "DomainName": "cli-example-1"  
  },  
  {  
    "DomainName": "cli-example-2"  
  }  
]  
}
```

Weitere Informationen finden Sie unter [Creating and Managing Amazon Elasticsearch Service Domains](#) im Amazon Elasticsearch Service Developer Guide.

- API-Einheiten finden Sie [ListDomainNames](#) in der AWS CLI Befehlsreferenz.

AWS OpsWorks Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS OpsWorks.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

assign-instance

Das folgende Codebeispiel zeigt die Verwendung `assign-instance`.

AWS CLI

Um einer Ebene eine registrierte Instanz zuzuweisen

Im folgenden Beispiel wird einer benutzerdefinierten Ebene eine registrierte Instanz zugewiesen.

```
aws opsworks --region us-east-1 assign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --layer-ids 26cf1d32-6876-42fa-bbf1-9cad0bfff938
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Zuweisen einer registrierten Instanz zu einem Layer im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie unter [AssignInstance AWS CLI](#) Befehlsreferenz.

assign-volume

Das folgende Codebeispiel zeigt die Verwendung `assign-volume`.

AWS CLI

Um einer Instance ein registriertes Volume zuzuweisen

Im folgenden Beispiel wird einer Instance ein registriertes Amazon Elastic Block Store (AmazonEBS) -Volume zugewiesen. Das Volume wird anhand seiner Volume-ID identifiziert, die AWS OpsWorks zugewiesen wird, wenn Sie GUID das Volume bei einem Stack registrieren, und nicht anhand der Volume-ID von Amazon Elastic Compute Cloud (AmazonEC2). Vor der Ausführung müssen Sie zunächst `assign-volume`, `update-volume` um dem Volume einen Bereitstellungspunkt zuzuweisen.

```
aws opsworks --region us-east-1 assign-volume --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --volume-id 26cf1d32-6876-42fa-bbf1-9cad0bff938
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Zuweisen von EBS Amazon-Volumes zu einer Instance im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [AssignVolume](#) in der AWS CLI Befehlsreferenz.

associate-elastic-ip

Das folgende Codebeispiel zeigt die Verwendung `associate-elastic-ip`.

AWS CLI

Um eine Elastic IP-Adresse mit einer Instance zu verknüpfen

Das folgende Beispiel verknüpft eine Elastic IP-Adresse mit einer angegebenen Instance.

```
aws opsworks --region us-east-1 associate-elastic-ip --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f --elastic-ip 54.148.130.96
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Ressourcenverwaltung.

- API-Einheiten finden Sie [AssociateElasticLoadBalancer](#) in der AWS CLI Befehlsreferenz.

attach-elastic-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `attach-elastic-load-balancer`.

AWS CLI

Um einen Load Balancer an eine Ebene anzuhängen

Im folgenden Beispiel wird ein Load Balancer, der anhand seines Namens identifiziert wird, einer angegebenen Ebene zugeordnet.

```
aws opsworks --region us-east-1 attach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Elastic Load Balancing im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [AttachElasticLoadBalancer](#) in der AWS CLI Befehlsreferenz.

create-app

Das folgende Codebeispiel zeigt die Verwendung `create-app`.

AWS CLI

Beispiel 1: Um eine App zu erstellen

Im folgenden Beispiel wird `implePHPApp` aus Code, der in einem GitHub Repository gespeichert ist, eine PHP App namens `S` erstellt. Der Befehl verwendet die Kurzform der Anwendungsquellendefinition.

```
aws opsworks create-app \  
  --region us-east-1 \  
  --stack-id f6673d70-32e6-4425-8999-265dd002fec7 \  
  --name SimplePHPApp \  
  --type php \  
  --source github:implePHPApp
```

```
--app-source Type=git,Url=git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git,Revision=version1
```

Ausgabe:

```
{  
  "AppId": "6cf5163c-a951-444f-a8f7-3716be75f2a2"  
}
```

Beispiel 2: Um eine App mit einer angehängten Datenbank zu erstellen

Im folgenden Beispiel wird eine JSP App aus Code erstellt, der in einem ZIP-Archiv in einem öffentlichen S3-Bucket gespeichert ist. Es hängt eine RDS DB-Instance an, die als Datenspeicher der App dient. Die Anwendungs- und Datenbankquellen sind in separaten JSON Dateien definiert, die sich in dem Verzeichnis befinden, von dem aus Sie den Befehl ausführen.

```
aws opsworks create-app \  
  --region us-east-1 \  
  --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8 \  
  --name SimpleJSP \  
  --type java \  
  --app-source file://appsource.json \  
  --data-sources file://datasource.json
```

Die Quellinformationen der Anwendung befinden sich in `appsource.json` und enthalten Folgendes.

```
{  
  "Type": "archive",  
  "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"  
}
```

Die Quellinformationen der Datenbank befinden sich in `datasource.json` und enthalten Folgendes.

```
[  
  {  
    "Type": "RdsDbInstance",  
    "Arn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",  
    "DatabaseName": "mydb"  }  
]
```

```
}  
]
```

Hinweis: Bei einer RDS DB-Instance müssen Sie zuerst die Instance mit dem Befehl `register-rds-db-instance` zum Registrieren beim Stack verwenden. Legen Sie für My SQL App Server-Instances die Einstellung `Type` auf `festOpsworksMySQLInstance`. Diese Instanzen wurden von AWS OpsWorks erstellt und müssen daher nicht registriert werden.

Ausgabe:

```
{  
  "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"  
}
```

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Hinzufügen von Apps](#).

- API-Einheiten finden Sie [CreateApp](#) in der AWS CLI Befehlsreferenz.

create-deployment

Das folgende Codebeispiel zeigt die Verwendung `create-deployment`.

AWS CLI

Beispiel 1: Um Apps bereitzustellen und Stack-Befehle auszuführen

Die folgenden Beispiele zeigen, wie Sie den `create-deployment` Befehl verwenden, um Apps bereitzustellen und Stack-Befehle auszuführen. Beachten Sie, dass den Anführungszeichen (") in dem JSON Objekt, das den Befehl spezifiziert, alle Escape-Zeichen (\) vorangestellt sind. Ohne die Escape-Zeichen gibt der Befehl möglicherweise einen ungültigen JSON Fehler zurück.

Im folgenden `create-deployment` Beispiel wird eine App auf einem angegebenen Stack bereitgestellt.

```
aws opsworks create-deployment \  
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \  
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \  
  --command "{\"Name\":\"deploy\"}"
```

Ausgabe:


```
{
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

Beispiel 2: Um eine Rails-App bereitzustellen und die Datenbank zu migrieren

Der folgende `create-deployment` Befehl stellt eine Ruby on Rails-App auf einem angegebenen Stack bereit und migriert die Datenbank.

```
aws opsworks create-deployment \
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \
  --command "{\"Name\": \"deploy\", \"Args\": {\"migrate\": [\"true\"]}}"
```

Ausgabe:

```
{
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

Weitere Informationen zur Bereitstellung finden Sie unter [Deployment Apps](#) im AWS OpsWorks Benutzerhandbuch.

Beispiel 3: Ein Rezept ausführen

Mit dem folgenden `create-deployment` Befehl wird ein benutzerdefiniertes Rezept, `phpapp::appsetup`, für die Instanzen in einem angegebenen Stack ausgeführt.

```
aws opsworks create-deployment \
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
  --command "{\"Name\": \"execute_recipes\", \"Args\": {\"recipes\": [\"phpapp::appsetup\"]}}"
```

Ausgabe:

```
{
  "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Ausführen von Stack-Befehlen](#).

Beispiel 4: Abhängigkeiten installieren

Der folgende `create-deployment` Befehl installiert Abhängigkeiten, wie Pakete oder Ruby-Gems, auf den Instanzen in einem angegebenen Stack.

```
aws opsworks create-deployment \  
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \  
  --command "{\"Name\": \"install_dependencies\"}"
```

Ausgabe:

```
{  
  "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"  
}
```

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Ausführen von Stack-Befehlen](#).

- API-Einheiten finden Sie [CreateDeployment](#) in der AWS CLI Befehlsreferenz.

`create-instance`

Das folgende Codebeispiel zeigt die Verwendung `create-instance`.

AWS CLI

Um eine Instanz zu erstellen

Der folgende `create-instance` Befehl erstellt eine `m1.large` Amazon Linux-Instanz mit dem Namen `myinstance1` in einem angegebenen Stack. Die Instanz ist einer Ebene zugewiesen.

```
aws opsworks --region us-east-1 create-instance --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --layer-ids 5c8c272a-f2d5-42e3-8245-5bf3927cb65b --hostname myinstance1 --instance-type m1.large --os "Amazon Linux"
```

Um einen automatisch generierten Namen zu verwenden, rufen Sie auf `get-hostname-suggestion`, der einen Hostnamen generiert, der auf dem Thema basiert, das Sie bei der Erstellung des Stacks angegeben haben. Übergeben Sie diesen Namen dann an das `hostname`-Argument.

Ausgabe:

```
{
  "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Hinzufügen einer Instanz zu einer Ebene im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [CreateInstance](#) in der AWS CLI Befehlsreferenz.

create-layer

Das folgende Codebeispiel zeigt die Verwendung `create-layer`.

AWS CLI

Um eine Ebene zu erstellen

Mit dem folgenden `create-layer` Befehl wird eine PHP App Server-Layer mit dem Namen `MyPHPLayer` in einem angegebenen Stapel erstellt.

```
aws opsworks create-layer --region us-east-1 --stack-
id f6673d70-32e6-4425-8999-265dd002fec7 --type php-app --name MyPHPLayer --
shortname myphpLayer
```

Ausgabe:

```
{
  "LayerId": "0b212672-6b4b-40e4-8a34-5a943cf2e07a"
}
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter So erstellen Sie eine Ebene.

- API-Einheiten finden Sie [CreateLayer](#) in der AWS CLI Befehlsreferenz.

create-server

Das folgende Codebeispiel zeigt die Verwendung `create-server`.

AWS CLI

Um einen Server zu erstellen

Das folgende `create-server` Beispiel erstellt einen neuen Chef Automate-Server mit `automate-06` dem Namen Ihrer Standardregion. Beachten Sie, dass Standardwerte für die meisten anderen Einstellungen verwendet werden, z. B. für die Anzahl der aufzubewahrenden Backups sowie für die Startzeiten für Wartung und Sicherung. Bevor Sie einen `create-server` Befehl ausführen, müssen Sie die Voraussetzungen unter [Erste Schritte mit AWS OpsWorks für Chef Automate](#) im AWS Opsworks for Chef Automate-Benutzerhandbuch erfüllen.

```
aws opsworks-cm create-server \
  --engine "ChefAutomate" \
  --instance-profile-arn "arn:aws:iam::012345678901:instance-profile/aws-opsworks-
cm-ec2-role" \
  --instance-type "t2.medium" \
  --server-name "automate-06" \
  --service-role-arn "arn:aws:iam::012345678901:role/aws-opsworks-cm-service-role"
```

Ausgabe:

```
{
  "Server": {
    "AssociatePublicIpAddress": true,
    "BackupRetentionCount": 10,
    "CreatedAt": 2019-12-29T13:38:47.520Z,
    "DisableAutomatedBackup": FALSE,
    "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
    "Engine": "ChefAutomate",
    "EngineAttributes": [
      {
        "Name": "CHEF_AUTOMATE_ADMIN_PASSWORD",
        "Value": "1Example1"
      }
    ],
    "EngineModel": "Single",
    "EngineVersion": "2019-08",
    "InstanceProfileArn": "arn:aws:iam::012345678901:instance-profile/aws-
opsworks-cm-ec2-role",
    "InstanceType": "t2.medium",
    "PreferredBackupWindow": "Sun:02:00",
    "PreferredMaintenanceWindow": "00:00",
```

```
"SecurityGroupIds": [ "sg-12345678" ],
"ServerArn": "arn:aws:iam::012345678901:instance/automate-06-1010V4UU2WRM2",
"ServerName": "automate-06",
"ServiceRoleArn": "arn:aws:iam::012345678901:role/aws-opsworks-cm-service-
role",
"Status": "CREATING",
"SubnetIds": [ "subnet-12345678" ]
}
}
```

Weitere Informationen finden Sie [CreateServer](#) in der Referenz AWS OpsWorks für Chef Automate API.

- API-Einheiten finden Sie [CreateServer](#) unter AWS CLI Befehlsreferenz.

create-stack

Das folgende Codebeispiel zeigt die Verwendung `create-stack`.

AWS CLI

Um einen Stapel zu erstellen

Der folgende `create-stack` Befehl erstellt einen Stack namens CLI Stack.

```
aws opsworks create-stack --name "CLI Stack" --stack-region "us-east-1" --service-
role-arn arn:aws:iam::123456789012:role/aws-opsworks-service-role --default-
instance-profile-arn arn:aws:iam::123456789012:instance-profile/aws-opsworks-ec2-
role --region us-east-1
```

Die Parameter `service-role-arn` und `default-instance-profile-arn` müssen angegeben werden. Normalerweise verwenden Sie diejenigen, die für Sie AWS OpsWorks erstellt werden, wenn Sie Ihren ersten Stack erstellen. Um die Amazon-Ressourcennamen (ARNs) für Ihr Konto abzurufen, gehen Sie zur IAM Konsole, wählen Sie Rollen im Navigationsbereich die Rolle oder das Profil aus und klicken Sie Summary auf die Registerkarte.

Ausgabe:

```
{
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"
}
```

Weitere Informationen

Weitere Informationen finden Sie unter [Create a New Stack](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [CreateStack](#) in der AWS CLI Befehlsreferenz.

create-user-profile

Das folgende Codebeispiel zeigt die Verwendung `create-user-profile`.

AWS CLI

Um ein Benutzerprofil zu erstellen

Sie importieren einen AWS Identity and Access Manager (IAM) -Benutzer in, AWS OpsWorks indem Sie ihn aufrufen `create-user-profile`, um ein Benutzerprofil zu erstellen. Das folgende Beispiel erstellt ein Benutzerprofil für den `cli-user-test` IAM Benutzer, der durch Amazon Resource Name (ARN) identifiziert wird. Das Beispiel weist dem Benutzer den SSH Benutzernamen von `myusername` zu und aktiviert die Selbstverwaltung, sodass der Benutzer einen SSH öffentlichen Schlüssel angeben kann.

```
aws opsworks --region us-east-1 create-user-profile --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test --ssh-username myusername --allow-self-management
```

Ausgabe:

```
{
  "IamUserArn": "arn:aws:iam::123456789102:user/cli-user-test"
}
```

Tipp: Dieser Befehl importiert einen IAM Benutzer in AWS OpsWorks, aber nur mit den Berechtigungen, die durch die angehängten Richtlinien gewährt werden. Mithilfe des `set-permissions` Befehls können Sie AWS OpsWorks Berechtigungen pro Stack gewähren.

Weitere Informationen

Weitere Informationen finden Sie im Benutzerhandbuch unter [AWS OpsWorks Benutzer importieren](#) AWS OpsWorks in.

- API-Einheiten finden Sie [CreateUserProfile](#) in der AWS CLI Befehlsreferenz.

delete-app

Das folgende Codebeispiel zeigt die Verwendung `delete-app`.

AWS CLI

Um eine App zu löschen

Im folgenden Beispiel wird eine angegebene App gelöscht, die anhand ihrer App-ID identifiziert wird. Sie können eine App-ID abrufen, indem Sie die Detailseite der App auf der AWS OpsWorks Konsole aufrufen oder den `describe-apps` Befehl ausführen.

```
aws opsworks delete-app --region us-east-1 --app-id 577943b9-2ec1-4baf-a7bf-1d347601edc5
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Apps.

- API-Einheiten finden Sie [DeleteApp](#) in der AWS CLI Befehlsreferenz.

delete-instance

Das folgende Codebeispiel zeigt die Verwendung `delete-instance`.

AWS CLI

Um eine Instanz zu löschen

Im folgenden `delete-instance` Beispiel wird eine angegebene Instanz gelöscht, die durch ihre Instanz-ID identifiziert wird. Sie finden eine Instanz-ID, indem Sie die Detailseite der Instanz in der AWS OpsWorks Konsole öffnen oder den `describe-instances` Befehl ausführen.

Wenn die Instanz online ist, müssen Sie die Instanz zuerst beenden, indem Sie sie aufrufen `stop-instance`, und dann warten, bis die Instanz gestoppt wurde. Führen Sie `describe-instances` den Befehl aus, um den Instanzstatus zu überprüfen.

Um die EBS Amazon-Volumes oder Elastic-IP-Adressen der Instanz zu entfernen, fügen Sie die `--delete-elastic-ip` Argumente `--delete-volumes` oder hinzu.

```
aws opsworks delete-instance \  
  --region us-east-1 \  
  --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Löschen von AWS OpsWorks Instanzen](#).

- APIEinzelheiten finden Sie [DeleteInstance](#) in der AWS CLI Befehlsreferenz.

delete-layer

Das folgende Codebeispiel zeigt die Verwendung `delete-layer`.

AWS CLI

Um eine Ebene zu löschen

Im folgenden Beispiel wird eine angegebene Ebene gelöscht, die durch ihre Layer-ID identifiziert wird. Sie können eine Layer-ID abrufen, indem Sie die Detailseite des Layers in der AWS OpsWorks Konsole aufrufen oder den `describe-layers` Befehl ausführen.

Hinweis: Bevor Sie eine Ebene löschen, müssen Sie `delete-instance` die Option zum Löschen aller Instanzen der Ebene verwenden.

```
aws opsworks delete-layer --region us-east-1 --layer-id a919454e-b816-4598-  
b29a-5796afb498ed
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Löschen von AWS OpsWorks Instanzen](#).

- APIEinzelheiten finden Sie [DeleteLayer](#) in der AWS CLI Befehlsreferenz.

delete-stack

Das folgende Codebeispiel zeigt die Verwendung `delete-stack`.

AWS CLI

Um einen Stapel zu löschen

Das folgende Beispiel löscht einen angegebenen Stapel, der durch seine Stack-ID identifiziert wird. Sie können eine Stack-ID abrufen, indem Sie in der AWS OpsWorks Konsole auf Stack-Einstellungen klicken oder den `describe-stacks` Befehl ausführen.

Hinweis: Bevor Sie eine Ebene löschen, müssen Sie `delete-appdelete-instance`, und verwenden, `delete-layer` um alle Apps, Instanzen und Ebenen des Stacks zu löschen.

```
aws opsworks delete-stack --region us-east-1 --stack-  
id 154a9d89-7e9e-433b-8de8-617e53756c84
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter [Einen Stack herunterfahren](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteStack](#) in der AWS CLI Befehlsreferenz.

delete-user-profile

Das folgende Codebeispiel zeigt die Verwendung `delete-user-profile`.

AWS CLI

Um ein Benutzerprofil zu löschen und einen IAM Benutzer zu entfernen AWS OpsWorks

Im folgenden Beispiel wird das Benutzerprofil für einen angegebenen AWS Identity and Access Management (IAM) -Benutzer gelöscht, der durch Amazon Resource Name (ARN) identifiziert wird. Der Vorgang entfernt den Benutzer von AWS OpsWorks, löscht ihn jedoch nicht. IAM Sie müssen die IAM Konsole, CLI, oder API für diese Aufgabe verwenden.

```
aws opsworks --region us-east-1 delete-user-profile --iam-user-  
arn arn:aws:iam::123456789102:user/cli-user-test
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im Benutzerhandbuch unter [AWS OpsWorks Benutzer importieren AWS OpsWorks in](#).

- API-Einheiten finden Sie [DeleteUserProfile](#) in der AWS CLI Befehlsreferenz.

deregister-elastic-ip

Das folgende Codebeispiel zeigt die Verwendung `deregister-elastic-ip`.

AWS CLI

Um eine Elastic IP-Adresse von einem Stack abzumelden

Im folgenden Beispiel wird die Registrierung einer Elastic IP-Adresse, die anhand ihrer IP-Adresse identifiziert wird, aus ihrem Stack aufgehoben.

```
aws opsworks deregister-elastic-ip --region us-east-1 --elastic-ip 54.148.130.96
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter [Deregistering Elastic IP Addresses](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterElasticIp](#) in der AWS CLI Befehlsreferenz.

deregister-instance

Das folgende Codebeispiel zeigt die Verwendung `deregister-instance`.

AWS CLI

Um eine registrierte Instance von einem Stack abzumelden

Mit dem folgenden `deregister-instance` Befehl wird die Registrierung einer registrierten Instance aus ihrem Stack aufgehoben.

```
aws opsworks --region us-east-1 deregister-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Deregistrierung einer registrierten Instance im AWS OpsWorks Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeregisterInstance](#) in der AWS CLI Befehlsreferenz.

deregister-rds-db-instance

Das folgende Codebeispiel zeigt die Verwendung `deregister-rds-db-instance`.

AWS CLI

Um eine RDS Amazon-DB-Instance von einem Stack abzumelden

Im folgenden Beispiel wird die Registrierung einer RDS DB-Instance, die durch ihre identifiziert wird, von ihrem Stack ARN aufgehoben.

```
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:123456789012:db:clitestdb
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Abmeldung von RDS Amazon-Instances im ASW OpsWorks Benutzerhandbuch.

Instanz-ID: clitestdb Master-Benutzername: cliuser Master: some23! PWD pwd Datenbankname: mydb
aws opsworks deregister-rds-db-instance --region us-east-1 -- arn:aws:rds:us-west-2:645732743964:db:clitestdb rds-db-instance-arn

- APIEinzelheiten [DeregisterRdsDbInstance](#) finden AWS CLI Sie in der Befehlsreferenz.

deregister-volume

Das folgende Codebeispiel zeigt die Verwendung `deregister-volume`.

AWS CLI

Um ein Amazon-Volume abzumelden EBS

Im folgenden Beispiel wird die Registrierung eines EBS Volumes aus seinem Stack aufgehoben. Das Volume wird anhand seiner Volume-ID identifiziert. Dabei handelt es sich um die Volume-IDGUID, die bei AWS OpsWorks der Registrierung des Volumes im Stack zugewiesen wurde, nicht anhand der EC2 Volume-ID.

```
aws opsworks deregister-volume --region us-east-1 --volume-id 5c48ef52-3144-4bf5-beaa-fda4deb23d4d
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Abmeldung von Amazon EBS Volumes im AWS OpsWorks Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeregisterVolume](#) in der AWS CLI Befehlsreferenz.

describe-apps

Das folgende Codebeispiel zeigt die Verwendung `describe-apps`.

AWS CLI

Um Apps zu beschreiben

Der folgende `describe-apps` Befehl beschreibt die Apps in einem angegebenen Stapel.

```
aws opsworks describe-apps \  
  --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a \  
  --region us-east-1
```

Ausgabe:

```
{  
  "Apps": [  
    {  
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",  
      "AppSource": {  
        "Url": "https://s3-us-west-2.amazonaws.com/opsworks-demo-assets/  
simplejsp.zip",
```

```

    "Type": "archive"
  },
  "Name": "SimpleJSP",
  "EnableSsl": false,
  "SslConfiguration": {},
  "AppId": "da1decc1-0dff-43ea-ad7c-bb667cd87c8b",
  "Attributes": {
    "RailsEnv": null,
    "AutoBundleOnDeploy": "true",
    "DocumentRoot": "ROOT"
  },
  "Shortname": "simplejsp",
  "Type": "other",
  "CreatedAt": "2013-08-01T21:46:54+00:00"
}
]
}

```

Weitere Informationen finden Sie unter Apps im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeApps](#) in der AWS CLI Befehlsreferenz.

describe-commands

Das folgende Codebeispiel zeigt die Verwendung des `describe-commands`.

AWS CLI

Um Befehle zu beschreiben

Der folgende `describe-commands` Befehl beschreibt die Befehle in einer angegebenen Instanz.

```

aws opsworks describe-commands \
  --instance-id 8c2673b9-3fe5-420d-9cfa-78d875ee7687 \
  --region us-east-1

```

Ausgabe:

```

{
  "Commands": [
    {
      "Status": "successful",

```

```

    "CompletedAt": "2013-07-25T18:57:47+00:00",
    "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
    "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
    "AcknowledgedAt": "2013-07-25T18:57:41+00:00",
    "LogUrl": "https://s3.amazonaws.com/<bucket-name>/logs/008c1a91-
ec59-4d51-971d-3adff54b00cc?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Expires=1375394373&Signature=HkXil6UuNfxTCC37EPQAa462E1E%3D&response-cache-
control=private&response-content-encoding=gzip&response-content-type=text%2Fplain",
    "Type": "undeploy",
    "CommandId": "008c1a91-ec59-4d51-971d-3adff54b00cc",
    "CreatedAt": "2013-07-25T18:57:34+00:00",
    "ExitCode": 0
  },
  {
    "Status": "successful",
    "CompletedAt": "2013-07-25T18:55:40+00:00",
    "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
    "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
    "AcknowledgedAt": "2013-07-25T18:55:32+00:00",
    "LogUrl": "https://s3.amazonaws.com/<bucket-name>/
logs/899d3d64-0384-47b6-a586-33433aad117c?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Expires=1375394373&Signature=xMsJvtLuUqWmsr8s%2FAjVru0BtRs%3D&response-cache-
control=private&response-content-encoding=gzip&response-content-type=text%2Fplain",
    "Type": "deploy",
    "CommandId": "899d3d64-0384-47b6-a586-33433aad117c",
    "CreatedAt": "2013-07-25T18:55:29+00:00",
    "ExitCode": 0
  }
]
}

```

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter AWS OpsWorks Lifecycle Events.

- API-Einheiten finden Sie [DescribeCommands](#) in der AWS CLI Befehlsreferenz.

describe-deployments

Das folgende Codebeispiel zeigt die Verwendung von `describe-deployments`.

AWS CLI

Um Bereitstellungen zu beschreiben

Der folgende `describe-deployments` Befehl beschreibt die Bereitstellungen in einem angegebenen Stack.

```
aws opsworks --region us-east-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

Ausgabe:

```
{
  "Deployments": [
    {
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
      "Status": "successful",
      "CompletedAt": "2013-07-25T18:57:49+00:00",
      "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
      "Command": {
        "Args": {},
        "Name": "undeploy"
      },
    },
    "CreatedAt": "2013-07-25T18:57:34+00:00",
    "Duration": 15,
    "InstanceIds": [
      "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
      "9e588a25-35b2-4804-bd43-488f85ebe5b7"
    ]
  },
  {
    "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
    "Status": "successful",
    "CompletedAt": "2013-07-25T18:56:41+00:00",
    "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
    "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
    "Command": {
      "Args": {},
      "Name": "deploy"
    },
    "InstanceIds": [
      "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
      "9e588a25-35b2-4804-bd43-488f85ebe5b7"
    ],
    "Duration": 72,
    "CreatedAt": "2013-07-25T18:55:29+00:00"
  }
}
```

```
]
}
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Bereitstellen von Apps.

- API-Einheiten finden Sie [DescribeDeployments](#) in der AWS CLI Befehlsreferenz.

describe-elastic-ips

Das folgende Codebeispiel zeigt die Verwendung `describe-elastic-ips`.

AWS CLI

Um Elastic IP-Instances zu beschreiben

Der folgende `describe-elastic-ips` Befehl beschreibt die Elastic IP-Adressen in einer angegebenen Instanz.

```
aws opsworks --region us-east-1 describe-elastic-ips --instance-id b62f3e04-  
e9eb-436c-a91f-d9e9a396b7b0
```

Ausgabe:

```
{
  "ElasticIps": [
    {
      "Ip": "192.0.2.0",
      "Domain": "standard",
      "Region": "us-west-2"
    }
  ]
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Instances im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeElasticIps](#) in der AWS CLI Befehlsreferenz.

describe-elastic-load-balancers

Das folgende Codebeispiel zeigt die Verwendung `describe-elastic-load-balancers`.

AWS CLI

Um die Elastic Load Balancer eines Stacks zu beschreiben

Der folgende `describe-elastic-load-balancers` Befehl beschreibt die Load Balancer eines angegebenen Stacks.

```
aws opsworks --region us-west-2 describe-elastic-load-balancers --stack-id 6f4660e5-37a6-4e42-bfa0-1358ebd9c182
```

Ausgabe: Dieser spezielle Stack hat einen Load Balancer.

```
{
  "ElasticLoadBalancers": [
    {
      "SubnetIds": [
        "subnet-60e4ea04",
        "subnet-66e1c110"
      ],
      "Ec2InstanceIds": [],
      "ElasticLoadBalancerName": "my-balancer",
      "Region": "us-west-2",
      "LayerId": "344973cb-bf2b-4cd0-8d93-51cd819bab04",
      "AvailabilityZones": [
        "us-west-2a",
        "us-west-2b"
      ],
      "VpcId": "vpc-b319f9d4",
      "StackId": "6f4660e5-37a6-4e42-bfa0-1358ebd9c182",
      "DnsName": "my-balancer-2094040179.us-west-2.elb.amazonaws.com"
    }
  ]
}
```

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Apps.

- API-Einheiten finden Sie [DescribeElasticLoadBalancers](#) in der AWS CLI Befehlsreferenz.

describe-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-instances`.

AWS CLI

Um Instanzen zu beschreiben

Der folgende `describe-instances` Befehl beschreibt die Instanzen in einem angegebenen Stack:

```
aws opsworks --region us-east-1 describe-instances --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Ausgabe: Das folgende Ausgabebeispiel bezieht sich auf einen Stack mit zwei Instanzen. Die erste ist eine registrierte EC2 Instanz, und die zweite wurde von erstellt AWS OpsWorks.

```
{
  "Instances": [
    {
      "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcdded3fa0f",
      "PrivateDns": "ip-10-31-39-66.us-west-2.compute.internal",
      "LayerIds": [
        "26cf1d32-6876-42fa-bbf1-9cadcb0bff938"
      ],
      "EbsOptimized": false,
      "ReportedOs": {
        "Version": "14.04",
        "Name": "ubuntu",
        "Family": "debian"
      },
      "Status": "online",
      "InstanceId": "4d6d1710-ded9-42a1-b08e-b043ad7af1e2",
      "SshKeyName": "US-West-2",
      "InfrastructureClass": "ec2",
      "RootDeviceVolumeId": "vol-d08ec6c1",
      "SubnetId": "subnet-b8de0ddd",
      "InstanceType": "t1.micro",
      "CreatedAt": "2015-02-24T20:52:49+00:00",
      "AmiId": "ami-35501205",
      "Hostname": "ip-192-0-2-0",
      "Ec2InstanceId": "i-5cd23551",
      "PublicDns": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com",
```

```
"SecurityGroupIds": [
  "sg-c4d3f0a1"
],
"Architecture": "x86_64",
"RootDeviceType": "ebs",
"InstallUpdatesOnBoot": true,
"Os": "Custom",
"VirtualizationType": "paravirtual",
"AvailabilityZone": "us-west-2a",
"PrivateIp": "10.31.39.66",
"PublicIp": "192.0.2.06",
"RegisteredBy": "arn:aws:iam::123456789102:user/AWS/OpsWorks/OpsWorks-
EC2Register-i-5cd23551"
},
{
  "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcdded3fa0f",
  "PrivateDns": "ip-10-31-39-158.us-west-2.compute.internal",
  "SshHostRsaKeyFingerprint": "69:6b:7b:8b:72:f3:ed:23:01:00:05:bc:9f:a4:60:c1",
  "LayerIds": [
    "26cf1d32-6876-42fa-bbf1-9cad0bfff938"
  ],
  "EbsOptimized": false,
  "ReportedOs": {},
  "Status": "booting",
  "InstanceId": "9b137a0d-2f5d-4cc0-9704-13da4b31fdcb",
  "SshKeyName": "US-West-2",
  "InfrastructureClass": "ec2",
  "RootDeviceVolumeId": "vol-e09dd5f1",
  "SubnetId": "subnet-b8de0ddd",
  "InstanceProfileArn": "arn:aws:iam::123456789102:instance-profile/aws-
opsworks-ec2-role",
  "InstanceType": "c3.large",
  "CreatedAt": "2015-02-24T21:29:33+00:00",
  "AmiId": "ami-9fc29baf",
  "SshHostDsaKeyFingerprint": "fc:87:95:c3:f5:e1:3b:9f:d2:06:6e:62:9a:35:27:e8",
  "Ec2InstanceId": "i-8d2dca80",
  "PublicDns": "ec2-192-0-2-1.us-west-2.compute.amazonaws.com",
  "SecurityGroupIds": [
    "sg-b022add5",
    "sg-b122add4"
  ],
  "Architecture": "x86_64",
  "RootDeviceType": "ebs",
  "InstallUpdatesOnBoot": true,
```

```
"Os": "Amazon Linux 2014.09",
"VirtualizationType": "paravirtual",
"AvailabilityZone": "us-west-2a",
"Hostname": "custom11",
"PrivateIp": "10.31.39.158",
"PublicIp": "192.0.2.0"
}
]
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Instances im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstances](#) in der AWS CLI Befehlsreferenz.

describe-layers

Das folgende Codebeispiel zeigt die Verwendung `describe-layers`.

AWS CLI

Um die Ebenen eines Stapels zu beschreiben

Der folgende `describe-layers` Befehl beschreibt die Ebenen in einem angegebenen Stapel:

```
aws opsworks --region us-east-1 describe-layers --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

Ausgabe:

```
{
  "Layers": [
    {
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
      "Type": "db-master",
      "DefaultSecurityGroupNames": [
        "AWS-OpsWorks-DB-Master-Server"
      ],
      "Name": "MySQL",
      "Packages": [],
      "DefaultRecipes": {
        "Undeploy": [],

```

```
    "Setup": [
      "opsworks_initial_setup",
      "ssh_host_keys",
      "ssh_users",
      "mysql::client",
      "dependencies",
      "ebs",
      "opsworks_ganglia::client",
      "mysql::server",
      "dependencies",
      "deploy::mysql"
    ],
    "Configure": [
      "opsworks_ganglia::configure-client",
      "ssh_users",
      "agent_version",
      "deploy::mysql"
    ],
    "Shutdown": [
      "opsworks_shutdown::default",
      "mysql::stop"
    ],
    "Deploy": [
      "deploy::default",
      "deploy::mysql"
    ]
  ],
  "CustomRecipes": {
    "Undeploy": [],
    "Setup": [],
    "Configure": [],
    "Shutdown": [],
    "Deploy": []
  },
  "EnableAutoHealing": false,
  "LayerId": "41a20847-d594-4325-8447-171821916b73",
  "Attributes": {
    "MysqlRootPasswordUbiquitous": "true",
    "RubygemsVersion": null,
    "RailsStack": null,
    "HaproxyHealthCheckMethod": null,
    "RubyVersion": null,
    "BundlerVersion": null,
    "HaproxyStatsPassword": null,
```

```
    "PassengerVersion": null,
    "MemcachedMemory": null,
    "EnableHaproxyStats": null,
    "ManageBundler": null,
    "NodejsVersion": null,
    "HaproxyHealthCheckUrl": null,
    "MysqlRootPassword": "*****FILTERED*****",
    "GangliaPassword": null,
    "GangliaUser": null,
    "HaproxyStatsUrl": null,
    "GangliaUrl": null,
    "HaproxyStatsUser": null
  },
  "Shortname": "db-master",
  "AutoAssignElasticIps": false,
  "CustomSecurityGroupIds": [],
  "CreatedAt": "2013-07-25T18:11:19+00:00",
  "VolumeConfigurations": [
    {
      "MountPoint": "/vol/mysql",
      "Size": 10,
      "NumberOfDisks": 1
    }
  ]
},
{
  "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
  "Type": "custom",
  "DefaultSecurityGroupNames": [
    "AWS-OpsWorks-Custom-Server"
  ],
  "Name": "TomCustom",
  "Packages": [],
  "DefaultRecipes": {
    "Undeploy": [],
    "Setup": [
      "opsworks_initial_setup",
      "ssh_host_keys",
      "ssh_users",
      "mysql::client",
      "dependencies",
      "ebs",
      "opsworks_ganglia::client"
    ]
  ],
}
```

```
"Configure": [
    "opsworks_ganglia::configure-client",
    "ssh_users",
    "agent_version"
],
"Shutdown": [
    "opsworks_shutdown::default"
],
"Deploy": [
    "deploy::default"
]
},
"CustomRecipes": {
    "Undeploy": [],
    "Setup": [
        "tomcat::setup"
    ],
    "Configure": [
        "tomcat::configure"
    ],
    "Shutdown": [],
    "Deploy": [
        "tomcat::deploy"
    ]
},
"EnableAutoHealing": true,
"LayerId": "e6cbcd29-d223-40fc-8243-2eb213377440",
"Attributes": {
    "MysqlRootPasswordUbiquitous": null,
    "RubygemsVersion": null,
    "RailsStack": null,
    "HaproxyHealthCheckMethod": null,
    "RubyVersion": null,
    "BundlerVersion": null,
    "HaproxyStatsPassword": null,
    "PassengerVersion": null,
    "MemcachedMemory": null,
    "EnableHaproxyStats": null,
    "ManageBundler": null,
    "NodejsVersion": null,
    "HaproxyHealthCheckUrl": null,
    "MysqlRootPassword": null,
    "GangliaPassword": null,
    "GangliaUser": null,
```

```

        "HaproxyStatsUrl": null,
        "GangliaUrl": null,
        "HaproxyStatsUser": null
    },
    "Shortname": "tomcustom",
    "AutoAssignElasticIps": false,
    "CustomSecurityGroupIds": [],
    "CreatedAt": "2013-07-25T18:12:53+00:00",
    "VolumeConfigurations": []
}
]
}

```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Ebenen.

- API-Einheiten finden Sie [DescribeLayers](#) in der AWS CLI Befehlsreferenz.

describe-load-based-auto-scaling

Das folgende Codebeispiel zeigt die Verwendung `describe-load-based-auto-scaling`.

AWS CLI

Um die lastbasierte Skalierungskonfiguration einer Ebene zu beschreiben

Das folgende Beispiel beschreibt die lastbasierte Skalierungskonfiguration einer bestimmten Ebene. Der Layer wird anhand seiner Layer-ID identifiziert, die Sie auf der Detailseite des Layers oder durch Ausführen `describe-layers` finden.

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

Ausgabe: Der Beispiel-Layer hat eine einzelne lastbasierte Instanz.

```

{
  "LoadBasedAutoScalingConfigurations": [
    {
      "DownScaling": {
        "IgnoreMetricsTime": 10,

```



```
    "ThresholdsWaitTime": 10,
    "InstanceCount": 1,
    "CpuThreshold": 30.0
  },
  "Enable": true,
  "UpScaling": {
    "IgnoreMetricsTime": 5,
    "ThresholdsWaitTime": 5,
    "InstanceCount": 1,
    "CpuThreshold": 80.0
  },
  "LayerId": "6bec29c9-c866-41a0-aba5-fa3e374ce2a1"
}
]
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [So funktioniert die automatische lastbasierte Skalierung](#).

- API-Einheiten finden Sie [DescribeLoadBasedAutoScaling](#) in der AWS CLI Befehlsreferenz.

describe-my-user-profile

Das folgende Codebeispiel zeigt die Verwendung `describe-my-user-profile`.

AWS CLI

Um ein Benutzerprofil abzurufen

Das folgende Beispiel zeigt, wie das Profil des AWS Identity and Access Management (IAM) - Benutzers abgerufen wird, der den Befehl ausführt.

```
aws opsworks --region us-east-1 describe-my-user-profile
```

Ausgabe: Der Kürze halber wird der größte Teil des SSH öffentlichen Schlüssels des Benutzers durch ein Auslassungszeichen (...) ersetzt.

```
{
  "UserProfile": {
```

```
"IamUserArn": "arn:aws:iam::123456789012:user/myusername",
"SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQ...3LQ4aX9jpxQw== rsa-
key-20141104",
"Name": "myusername",
"SshUsername": "myusername"
}
}
```

Weitere Informationen

Weitere Informationen finden Sie im Benutzerhandbuch unter [AWS OpsWorks Benutzer importieren AWS OpsWorks in](#).

- API-Einheiten finden Sie [DescribeMyUserProfile](#) in der AWS CLI Befehlsreferenz.

describe-permissions

Das folgende Codebeispiel zeigt die Verwendung `describe-permissions`.

AWS CLI

Um die AWS OpsWorks Berechtigungsstufe eines Benutzers pro Stack abzurufen

Das folgende Beispiel zeigt, wie Sie die Berechtigungsstufe eines AWS Identity and Access Management (IAM) -Benutzers für einen angegebenen Stack abrufen können.

```
aws opsworks --region us-east-1 describe-permissions --iam-user-
arn arn:aws:iam::123456789012:user/cli-user-test --stack-id d72553d4-8727-448c-9b00-
f024f0ba1b06
```

Ausgabe:

```
{
  "Permissions": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "Level": "manage",
      "AllowSudo": true,
      "AllowSsh": true
    }
  ]
}
```

```
]
}
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Gewähren von Zugriffsstufen pro Stack.

- API-Einheiten finden Sie [DescribePermissions](#) in der AWS CLI Befehlsreferenz.

describe-raid-arrays

Das folgende Codebeispiel zeigt die Verwendung `describe-raid-arrays`.

AWS CLI

Um RAID Arrays zu beschreiben

Das folgende Beispiel beschreibt die RAID Arrays, die an die Instanzen in einem bestimmten Stack angehängt sind.

```
aws opsworks --region us-east-1 describe-raid-arrays --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

Ausgabe: Das Folgende ist die Ausgabe für einen Stack mit einem RAID Array.

```
{
  "RaidArrays": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "AvailabilityZone": "us-west-2a",
      "Name": "Created for php-app1",
      "NumberOfDisks": 2,
      "InstanceId": "9f14adbc-ced5-43b6-bf01-e7d0db6cf2f7",
      "RaidLevel": 0,
      "VolumeType": "standard",
      "RaidArrayId": "f2d4e470-5972-4676-b1b8-bae41ec3e51c",
      "Device": "/dev/md0",
      "MountPoint": "/mnt/workspace",
      "CreatedAt": "2015-02-26T23:53:09+00:00",
      "Size": 100
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter EBS Volumes im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeRaidArrays](#) in der AWS CLI Befehlsreferenz.

describe-rds-db-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-rds-db-instances`.

AWS CLI

Um die registrierten RDS Amazon-Instances eines Stacks zu beschreiben

Das folgende Beispiel beschreibt die RDS Amazon-Instances, die für einen bestimmten Stack registriert sind.

```
aws opsworks --region us-east-1 describe-rds-db-instances --stack-  
id d72553d4-8727-448c-9b00-f024f0ba1b06
```

Ausgabe: Das Folgende ist die Ausgabe für einen Stack mit einer registrierten RDS Instance.

```
{  
  "RdsDbInstances": [  
    {  
      "Engine": "mysql",  
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",  
      "MissingOnRds": false,  
      "Region": "us-west-2",  
      "RdsDbInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",  
      "DbPassword": "*****FILTERED*****",  
      "Address": "clitestdb.cd1qlk5uwd0k.us-west-2.rds.amazonaws.com",  
      "DbUser": "cliuser",  
      "DbInstanceIdentifier": "clitestdb"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter Resource Management im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeRdsDbInstances](#) in der AWS CLI Befehlsreferenz.

describe-stack-provisioning-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-provisioning-parameters`.

AWS CLI

Um die Bereitstellungsparameter für einen Stack zurückzugeben

Das folgende `describe-stack-provisioning-parameters` Beispiel gibt die Bereitstellungsparameter für einen angegebenen Stack zurück. Zu den Bereitstellungsparametern gehören Einstellungen wie der Installationsort des Agenten und der öffentliche Schlüssel, OpsWorks mit dem der Agent auf Instanzen in einem Stack verwaltet wird.

```
aws opsworks describe-stack-provisioning-parameters \
  --stack-id 62744d97-6faf-4ecb-969b-a086fEXAMPLE
```

Ausgabe:

```
{
  "AgentInstallerUrl": "https://opsworks-instance-agent-us-west-2.s3.amazonaws.com/ID_number/opsworks-agent-installer.tgz",
  "Parameters": {
    "agent_installer_base_url": "https://opsworks-instance-agent-us-west-2.s3.amazonaws.com",
    "agent_installer_tgz": "opsworks-agent-installer.tgz",
    "assets_download_bucket": "opsworks-instance-assets-us-west-2.s3.amazonaws.com",
    "charlie_public_key": "-----BEGIN PUBLIC KEY-----PUBLIC_KEY_EXAMPLE\n-----END PUBLIC KEY-----",
    "instance_service_endpoint": "opsworks-instance-service.us-west-2.amazonaws.com",
    "instance_service_port": "443",
    "instance_service_region": "us-west-2",
    "instance_service_ssl_verify_peer": "true",
    "instance_service_use_ssl": "true",
    "ops_works_endpoint": "opsworks.us-west-2.amazonaws.com",
    "ops_works_port": "443",
    "ops_works_region": "us-west-2",
    "ops_works_ssl_verify_peer": "true",
    "ops_works_use_ssl": "true",
```

```
        "verbose": "false",
        "wait_between_runs": "30"
    }
}
```

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Stack-Befehle ausführen](#).

- API-Einheiten finden Sie [DescribeStackProvisioningParameters](#) in der AWS CLI Befehlsreferenz.

describe-stack-summary

Das folgende Codebeispiel zeigt die Verwendung `describe-stack-summary`.

AWS CLI

Um die Konfiguration eines Stacks zu beschreiben

Der folgende `describe-stack-summary` Befehl gibt eine Zusammenfassung der Konfiguration des angegebenen Stacks zurück.

```
aws opsworks --region us-east-1 describe-stack-summary --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Ausgabe:

```
{
  "StackSummary": {
    "StackId": "8c428b08-a1a1-46ce-a5f8-feddc43771b8",
    "InstancesCount": {
      "Booting": 1
    },
    "Name": "CLITest",
    "AppsCount": 1,
    "LayersCount": 1,
    "Arn": "arn:aws:opsworks:us-west-2:123456789012:stack/8c428b08-a1a1-46ce-a5f8-feddc43771b8/"
  }
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Stacks im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeStackSummary](#) in der AWS CLI Befehlsreferenz.

describe-stacks

Das folgende Codebeispiel zeigt die Verwendung `describe-stacks`.

AWS CLI

Um Stapel zu beschreiben

Der folgende `describe-stacks` Befehl beschreibt die Stacks eines Accounts.

```
aws opsworks --region us-east-1 describe-stacks
```

Ausgabe:

```
{
  "Stacks": [
    {
      "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
      "StackId": "aeb7523e-7c8b-49d4-b866-03aae9d4fbc",
      "DefaultRootDeviceType": "instance-store",
      "Name": "TomStack-sd",
      "ConfigurationManager": {
        "Version": "11.4",
        "Name": "Chef"
      },
      "UseCustomCookbooks": true,
      "CustomJson": "{\n  \"tomcat\": {\n    \"base_version\": 7,\n    \"java_opts\n\": \"-Djava.awt.headless=true -Xmx256m\"\n  },\n  \"datasources\": {\n    \"ROOT\":\n  \"jdbc/mydb\"\n  }\n}",
      "Region": "us-east-1",
      "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/aws-opsworks-ec2-role",
      "CustomCookbooksSource": {
        "Url": "git://github.com/example-repo/tomcustom.git",
        "Type": "git"
      },
      "DefaultAvailabilityZone": "us-east-1a",
      "HostnameTheme": "Layer_Dependent",
      "Attributes": {
```

```
    "Color": "rgb(45, 114, 184)"
  },
  "DefaultOs": "Amazon Linux",
  "CreatedAt": "2013-08-01T22:53:42+00:00"
},
{
  "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
  "StackId": "40738975-da59-4c5b-9789-3e422f2cf099",
  "DefaultRootDeviceType": "instance-store",
  "Name": "MyStack",
  "ConfigurationManager": {
    "Version": "11.4",
    "Name": "Chef"
  },
  "UseCustomCookbooks": false,
  "Region": "us-east-1",
  "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/aws-opsworks-ec2-role",
  "CustomCookbooksSource": {},
  "DefaultAvailabilityZone": "us-east-1a",
  "HostnameTheme": "Layer_Dependent",
  "Attributes": {
    "Color": "rgb(45, 114, 184)"
  },
  "DefaultOs": "Amazon Linux",
  "CreatedAt": "2013-10-25T19:24:30+00:00"
}
]
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Stacks im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeStacks](#) in der AWS CLI Befehlsreferenz.

describe-timebased-auto-scaling

Das folgende Codebeispiel zeigt die Verwendung `describe-timebased-auto-scaling`.

AWS CLI

Um die zeitbasierte Skalierungskonfiguration einer Instanz zu beschreiben

Das folgende Beispiel beschreibt die zeitbasierte Skalierungskonfiguration einer angegebenen Instanz. Die Instance wird anhand ihrer Instance-ID identifiziert, die Sie auf der Detailseite der Instances oder durch Ausführen `describe-instances` finden.

```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

Ausgabe: Das Beispiel hat eine einzelne zeitbasierte Instanz.

```
{
  "TimeBasedAutoScalingConfigurations": [
    {
      "InstanceId": "701f2ffe-5d8e-4187-b140-77b75f55de8d",
      "AutoScalingSchedule": {
        "Monday": {
          "11": "on",
          "10": "on",
          "13": "on",
          "12": "on"
        },
        "Tuesday": {
          "11": "on",
          "10": "on",
          "13": "on",
          "12": "on"
        }
      }
    }
  ]
}
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [So funktioniert die automatische zeitbasierte Skalierung](#).

- API-Einheiten finden Sie [DescribeTimebasedAutoScaling](#) in der AWS CLI Befehlsreferenz.

describe-user-profiles

Das folgende Codebeispiel zeigt die Verwendung `describe-user-profiles`.

AWS CLI

Um Benutzerprofile zu beschreiben

Der folgende `describe-user-profiles` Befehl beschreibt die Benutzerprofile des Kontos.

```
aws opsworks --region us-east-1 describe-user-profiles
```

Ausgabe:

```
{
  "UserProfiles": [
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
      "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEakOuP7i80q3Cko...",
      "AllowSelfManagement": true,
      "Name": "someuser",
      "SshUsername": "someuser"
    },
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "AllowSelfManagement": true,
      "Name": "cli-user-test",
      "SshUsername": "myusername"
    }
  ]
}
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [AWS OpsWorks Benutzer verwalten](#).

- API-Einheiten finden Sie [DescribeUserProfiles](#) in der AWS CLI Befehlsreferenz.

describe-volumes

Das folgende Codebeispiel zeigt die Verwendung `describe-volumes`.

AWS CLI

Um die Volumes eines Stacks zu beschreiben

Das folgende Beispiel beschreibt die EBS Volumes eines Stacks.

```
aws opsworks --region us-east-1 describe-volumes --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Ausgabe:

```
{
  "Volumes": [
    {
      "Status": "in-use",
      "AvailabilityZone": "us-west-2a",
      "Name": "CLITest",
      "InstanceId": "dfe18b02-5327-493d-91a4-c5c0c448927f",
      "VolumeType": "standard",
      "VolumeId": "56b66fbd-e1a1-4aff-9227-70f77118d4c5",
      "Device": "/dev/sdi",
      "Ec2VolumeId": "vol-295c1638",
      "MountPoint": "/mnt/myvolume",
      "Size": 1
    }
  ]
}
```

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Ressourcenverwaltung.

- API-Einheiten finden Sie [DescribeVolumes](#) in der AWS CLI Befehlsreferenz.

detach-elastic-load-balancer

Das folgende Codebeispiel zeigt die Verwendung `detach-elastic-load-balancer`.

AWS CLI

Um einen Load Balancer von seiner Ebene zu trennen

Im folgenden Beispiel wird ein Load Balancer, der anhand seines Namens identifiziert wird, von seiner Ebene getrennt.

```
aws opsworks --region us-east-1 detach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Elastic Load Balancing im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DetachElasticLoadBalancer](#) in der AWS CLI Befehlsreferenz.

disassociate-elastic-ip

Das folgende Codebeispiel zeigt die Verwendung `disassociate-elastic-ip`.

AWS CLI

Um die Zuordnung einer Elastic IP-Adresse zu einer Instance zu trennen

Im folgenden Beispiel wird die Zuordnung einer Elastic IP-Adresse zu einer angegebenen Instance aufgehoben.

```
aws opsworks --region us-east-1 disassociate-elastic-ip --elastic-ip 54.148.130.96
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Ressourcenverwaltung.

- API-Einheiten finden Sie [DisassociateElasticIp](#) in der AWS CLI Befehlsreferenz.

get-hostname-suggestion

Das folgende Codebeispiel zeigt die Verwendung `get-hostname-suggestion`.

AWS CLI

Um den nächsten Hostnamen für eine Ebene abzurufen

Im folgenden Beispiel wird der nächste generierte Hostname für eine angegebene Ebene abgerufen. Die für dieses Beispiel verwendete Schicht ist eine Java Application Server-Schicht mit einer Instanz. Das Hostnamen-Thema des Stacks ist standardmäßig Layer_Dependent.

```
aws opsworks --region us-east-1 get-hostname-suggestion --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Ausgabe:

```
{
  "Hostname": "java-app2",
  "LayerId": "888c5645-09a5-4d0e-95a8-812ef1db76a4"
}
```

Weitere Informationen

Weitere Informationen finden Sie unter [Create a New Stack](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [GetHostnameSuggestion](#) in der AWS CLI Befehlsreferenz.

reboot-instance

Das folgende Codebeispiel zeigt die Verwendung `reboot-instance`.

AWS CLI

Um eine Instanz neu zu starten

Im folgenden Beispiel wird eine Instanz neu gestartet.

```
aws opsworks --region us-east-1 reboot-instance --instance-id dfc18b02-5327-493d-91a4-c5c0c448927f
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter [Rebooting a Instance](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [RebootInstance](#) in der AWS CLI Befehlsreferenz.

register-elastic-ip

Das folgende Codebeispiel zeigt die Verwendung `register-elastic-ip`.

AWS CLI

Um eine Elastic IP-Adresse bei einem Stack zu registrieren

Im folgenden Beispiel wird eine Elastic IP-Adresse, die anhand ihrer IP-Adresse identifiziert wird, bei einem angegebenen Stack registriert.

Hinweis: Die Elastic IP-Adresse muss sich in derselben Region wie der Stack befinden.

```
aws opsworks register-elastic-ip --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --elastic-ip 54.148.130.96
```

Ausgabe

```
{
  "ElasticIp": "54.148.130.96"
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Registrierung von Elastic IP-Adressen mit einem Stack im OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterElasticIp](#) in der AWS CLI Befehlsreferenz.

register-rds-db-instance

Das folgende Codebeispiel zeigt die Verwendung `register-rds-db-instance`.

AWS CLI

Um eine RDS Amazon-Instance bei einem Stack zu registrieren

Das folgende Beispiel registriert eine RDS Amazon-DB-Instance, die durch ihren Amazon-Ressourcennamen (ARN) identifiziert wird, mit einem angegebenen Stack. Es gibt auch den Master-Benutzernamen und das Passwort der Instance an. Beachten Sie, dass AWS OpsWorks

keiner dieser Werte validiert wird. Wenn einer der beiden falsch ist, kann Ihre Anwendung keine Verbindung zur Datenbank herstellen.

```
aws opsworks register-rds-db-instance --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --rds-db-instance-arn arn:aws:rds:us-west-2:123456789012:db:clitestdb --db-user cliuser --db-password some23!pwd
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Registrierung von RDS Amazon-Instances mit einem Stack im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterRdsDbInstance](#) in der AWS CLI Befehlsreferenz.

register-volume

Das folgende Codebeispiel zeigt die Verwendung `register-volume`.

AWS CLI

Um ein EBS Amazon-Volume bei einem Stack zu registrieren

Im folgenden Beispiel wird ein EBS Amazon-Volume, das durch seine Volume-ID identifiziert wird, mit einem angegebenen Stack registriert.

```
aws opsworks register-volume --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --ec-2-volume-id vol-295c1638
```

Ausgabe:

```
{
  "VolumeId": "ee08039c-7cb7-469f-be10-40fb7f0c05e8"
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Registrierung von Amazon EBS Volumes mit einem Stack im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterVolume](#) in der AWS CLI Befehlsreferenz.

register

Das folgende Codebeispiel zeigt die Verwendung `register`.

AWS CLI

Um Instances mit einem Stack zu registrieren

Die folgenden Beispiele zeigen verschiedene Möglichkeiten, Instanzen mit einem Stack zu registrieren, die außerhalb von AWS Opsworks erstellt wurden. Sie können `register` von der Instanz aus, die registriert werden soll, oder von einer separaten Workstation aus ausführen. Weitere Informationen finden Sie unter Registrierung von Amazon EC2 - und On-Premises-Instances im AWS OpsWorks Benutzerhandbuch.

Hinweis: Der Kürze halber wird in den Beispielen das Argument `region` weggelassen.

Um eine EC2 Amazon-Instance zu registrieren

Um anzugeben, dass Sie eine EC2 Instance registrieren, setzen Sie das `--infrastructure-class` Argument auf `ec2`.

Im folgenden Beispiel wird eine EC2 Instanz mit dem angegebenen Stack von einer separaten Workstation aus registriert. Die Instanz wird durch ihre EC2 ID identifiziert, `i-12345678`. Das Beispiel verwendet den SSH Standardbenutzernamen der Workstation und versucht, sich mit Authentifizierungstechniken, für die kein Passwort erforderlich ist, wie z. B. einem privaten SSH Standardschlüssel, bei der Instanz anzumelden. Schlägt das fehl, `register` fragt das Passwort ab.

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb i-12345678
```

Im folgenden Beispiel wird eine EC2 Instanz mit dem angegebenen Stack von einer separaten Workstation aus registriert. Es verwendet die `--ssh-private-key` Argumente `--ssh-username` und, um explizit den SSH Benutzernamen und die private Schlüsseldatei anzugeben, mit denen sich der Befehl bei der Instanz anmeldet. `ec2-user` ist der Standardbenutzername für Amazon Linux-Instances. `ubuntu` für Ubuntu-Instances verwenden.

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --ssh-username ec2-user --ssh-private-key ssh_private_key i-12345678
```


Das folgende Beispiel registriert die EC2 Instanz, die den `register` Befehl ausführt. Melden Sie sich bei der Instanz an SSH und führen Sie sie `register` mit dem `--local` Argument statt mit einer Instanz-ID oder einem Hostnamen aus.

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

Um eine lokale Instanz zu registrieren

Um anzugeben, dass Sie eine lokale Instanz registrieren, setzen Sie das `--infrastructure-class` Argument auf `on-premises`

Im folgenden Beispiel wird eine vorhandene lokale Instanz mit einem angegebenen Stack von einer separaten Workstation aus registriert. Die Instanz wird anhand ihrer IP-Adresse identifiziert, `192.0.2.3`. Das Beispiel verwendet den SSH Standardbenutzernamen der Workstation und versucht, sich mit Authentifizierungstechniken, für die kein Passwort erforderlich ist, wie z. B. einem privaten SSH Standardschlüssel, bei der Instanz anzumelden. Schlägt das fehl, `register` fragt das Passwort ab.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

Im folgenden Beispiel wird eine lokale Instanz mit einem angegebenen Stack von einer separaten Workstation aus registriert. Die Instanz wird durch ihren Hostnamen identifiziert, `host1`. Die `--override-...` Argumente AWS OpsWorks werden direkt `webserver1` als Hostname bzw. `10.0.0.2` als öffentliche und private IP-Adressen der Instanz angezeigt. `192.0.2.3`

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --override-hostname webserver1 --override-public-ip 192.0.2.3 --override-private-ip 10.0.0.2 host1
```

Im folgenden Beispiel wird eine lokale Instanz mit einem angegebenen Stack von einer separaten Workstation aus registriert. Die Instanz wird anhand ihrer IP-Adresse identifiziert. `register` meldet sich mit dem angegebenen SSH Benutzernamen und der privaten Schlüsseldatei bei der Instanz an.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --ssh-username admin --ssh-private-key ssh_private_key 192.0.2.3
```

Im folgenden Beispiel wird eine bestehende lokale Instanz mit einem angegebenen Stack von einer separaten Workstation aus registriert. Der Befehl meldet sich mit einer benutzerdefinierten SSH Befehlszeichenfolge, die das SSH Passwort und die IP-Adresse der Instanz angibt, bei der Instanz an.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --override-ssh "sshpass -p 'mypassword' ssh your-user@192.0.2.3"
```

Im folgenden Beispiel wird die lokale Instanz registriert, die den `register` Befehl ausführt. Melden Sie sich bei der Instanz an SSH und führen Sie sie `register` mit dem `--local` Argument anstelle einer Instanz-ID oder eines Hostnamens aus.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

Ausgabe: Die folgende Ausgabe ist eine typische Ausgabe für die Registrierung einer EC2 Instanz.

```
Warning: Permanently added '52.11.41.206' (ECDSA) to the list of known hosts.
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload    Total   Spent    Left   Speed
100 6403k  100 6403k    0     0 2121k      0  0:00:03  0:00:03 --:--:-- 2121k
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Initializing AWS OpsWorks
environment
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Running on Ubuntu
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Checking if OS is supported
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Running on supported OS
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Setup motd
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing: ln -sf --backup /etc/
motd.opsworks-static /etc/motd
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Enabling multiverse repositories
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Customizing APT environment
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Installing system packages
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing: dpkg --configure -a
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing with retry: apt-get
update
[Tue, 24 Feb 2015 20:49:13 +0000] opsworks-init: Executing: apt-get install -y ruby
ruby-dev libicu-dev libssl-dev libxslt-dev libxml2-dev libyaml-dev monit
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Using assets bucket from
environment: 'opsworks-instance-assets-us-east-1.s3.amazonaws.com'.
```

```
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Installing Ruby for the agent
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Executing: /tmp/opsworks-
agent-installer.YgGq8wF3UUre6yDy/opsworks-agent-installer/opsworks-agent/bin/
installer_wrapper.sh -r -R opsworks-instance-assets-us-east-1.s3.amazonaws.com
[Tue, 24 Feb 2015 20:50:44 +0000] opsworks-init: Starting the installer
Instance successfully registered. Instance ID: 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
Connection to 52.11.41.206 closed.
```

Weitere Informationen

Weitere Informationen finden Sie unter [Registrierung einer Instance mit einem AWS OpsWorks Stack](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie unter [Registrierung](#) in der AWS CLI Befehlsreferenz.

set-load-based-auto-scaling

Das folgende Codebeispiel zeigt die Verwendung `set-load-based-auto-scaling`.

AWS CLI

So legen Sie die lastbasierte Skalierungskonfiguration für eine Ebene fest

Das folgende Beispiel aktiviert die lastbasierte Skalierung für einen angegebenen Layer und legt die Konfiguration für diesen Layer fest. Sie müssen verwenden `create-instance`, um dem Layer lastbasierte Instanzen hinzuzufügen.

```
aws opsworks --region us-east-1 set-load-based-auto-scaling --layer-
id 523569ae-2faf-47ac-b39e-f4c4b381f36d --enable --up-scaling file://upscale.json --
down-scaling file://downscale.json
```

In diesem Beispiel werden die Einstellungen für den Upscaling-Schwellenwert in einer separaten Datei im Arbeitsverzeichnis mit dem Namen `gespeichertupscale.json`, die Folgendes enthält.

```
{
  "InstanceCount": 2,
  "ThresholdsWaitTime": 3,
  "IgnoreMetricsTime": 3,
  "CpuThreshold": 85,
  "MemoryThreshold": 85,
  "LoadThreshold": 85
}
```

In dem Beispiel werden die Schwellenwerte für das Herunterskalieren in einer separaten Datei im Arbeitsverzeichnis mit dem Namen `gespeichertdownscale.json`, die Folgendes enthält.

```
{
  "InstanceCount": 2,
  "ThresholdsWaitTime": 3,
  "IgnoreMetricsTime": 3,
  "CpuThreshold": 35,
  "MemoryThreshold": 30,
  "LoadThreshold": 30
}
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter [Verwenden der automatischen lastbasierten Skalierung im AWS OpsWorks Benutzerhandbuch](#).

- API-Einheiten finden Sie [SetLoadBasedAutoScaling](#) in der AWS CLI Befehlsreferenz.

set-permission

Das folgende Codebeispiel zeigt die Verwendung `set-permission`.

AWS CLI

Um pro Stack AWS OpsWorks Berechtigungsstufen zu gewähren

Wenn Sie einen AWS Identity and Access Management (IAM) -Benutzer AWS OpsWorks per Anruf in `importierencreate-user-profile`, hat der Benutzer nur die Berechtigungen, die durch die angehängten IAM Richtlinien gewährt werden. Sie können AWS OpsWorks Berechtigungen gewähren, indem Sie die Richtlinien eines Benutzers ändern. Es ist jedoch oft einfacher, einen Benutzer zu importieren und dann den `set-permission` Befehl zu verwenden, um dem Benutzer eine der Standardberechtigungsstufen für jeden Stack zu gewähren, auf den der Benutzer Zugriff benötigt.

Das folgende Beispiel erteilt einem Benutzer, der durch Amazon Resource Name (ARN) identifiziert wird, die Erlaubnis für den angegebenen Stack. Das Beispiel gewährt dem Benutzer die Berechtigungsstufe „Verwalten“ mit Sudo und SSH Berechtigungen für die Instanzen des Stacks.

```
aws opsworks set-permission --region us-east-1 --stack-id 71c7ca72-55ae-4b6a-8ee1-a8dcded3fa0f --level manage --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test --allow-ssh --allow-sudo
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Erteilen von AWS OpsWorks Benutzerberechtigungen pro Stack im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [SetPermission](#) in der AWS CLI Befehlsreferenz.

set-time-based-auto-scaling

Das folgende Codebeispiel zeigt die Verwendung `set-time-based-auto-scaling`.

AWS CLI

Um die zeitbasierte Skalierungskonfiguration für eine Ebene festzulegen

Im folgenden Beispiel wird die zeitbasierte Konfiguration für eine angegebene Instanz festgelegt. Sie müssen zuerst verwenden `create-instance`, um die Instanz dem Layer hinzuzufügen.

```
aws opsworks --region us-east-1 set-time-based-auto-scaling --instance-id 69b6237c-08c0-4edb-a6af-78f3d01cedf2 --auto-scaling-schedule file://schedule.json
```

In dem Beispiel wird der Zeitplan in einer separaten Datei im Arbeitsverzeichnis mit dem Namen `schedule.json` gespeichert. In diesem Beispiel ist die Instanz am Montag und Dienstag gegen Mittag UTC (koordinierte Weltzeit) einige Stunden lang aktiv.

```
{
  "Monday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  },
  "Tuesday": {
    "10": "on",
    "11": "on",
  }
```

```
"12": "on",  
"13": "on"  
}  
}
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Verwenden der automatischen zeitbasierten Skalierung im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [SetTimeBasedAutoScaling](#) in der AWS CLI Befehlsreferenz.

start-instance

Das folgende Codebeispiel zeigt die Verwendung `start-instance`.

AWS CLI

Um eine Instanz zu starten

Der folgende `start-instance` Befehl startet eine angegebene 24/7-Instanz.

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

Ausgabe: Keine. Verwenden Sie `describe-instances`, um den Status der Instanz zu überprüfen.

Tipp: Sie können jede Offline-Instanz in einem Stack mit einem Befehl starten, indem Sie `start-stack` aufrufen.

Weitere Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Manuelles Starten, Stoppen und Neustarten von 24/7-Instances.

- API-Einheiten finden Sie [StartInstance](#) in der AWS CLI Befehlsreferenz.

start-stack

Das folgende Codebeispiel zeigt die Verwendung `start-stack`.

AWS CLI

Um die Instanzen eines Stacks zu starten

Im folgenden Beispiel werden alle 24/7-Instances eines Stacks gestartet. Um eine bestimmte Instanz zu starten, verwenden Sie `start-instance`.

```
aws opsworks --region us-east-1 start-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Eine Instanz starten](#).

- API-Einheiten finden Sie [StartStack](#) in der AWS CLI Befehlsreferenz.

stop-instance

Das folgende Codebeispiel zeigt die Verwendung `stop-instance`.

AWS CLI

Um eine Instanz zu stoppen

Im folgenden Beispiel wird eine angegebene Instanz gestoppt, die durch ihre Instanz-ID identifiziert wird. Sie können eine Instance-ID abrufen, indem Sie die Detailseite der Instanz auf der AWS OpsWorks Konsole aufrufen oder den `describe-instances` Befehl ausführen.

```
aws opsworks stop-instance --region us-east-1 --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

Sie können eine gestoppte Instanz neu starten, indem Sie sie aufrufen, `start-instance` oder indem Sie die Instanz durch einen Aufruf `delete-instance` löschen.

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Stoppen einer Instance.

- API-Einheiten finden Sie [StopInstance](#) in der AWS CLI Befehlsreferenz.

stop-stack

Das folgende Codebeispiel zeigt die Verwendung `stop-stack`.

AWS CLI

Um die Instanzen eines Stacks zu stoppen

Im folgenden Beispiel werden alle 24/7-Instances eines Stacks gestoppt. Um eine bestimmte Instanz zu stoppen, verwenden Sie `stop-instance`.

```
aws opsworks --region us-east-1 stop-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Ausgabe: Keine Ausgabe.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Stoppen einer Instance.

- API-Einheiten finden Sie [StopStack](#) in der AWS CLI Befehlsreferenz.

unassign-instance

Das folgende Codebeispiel zeigt die Verwendung `unassign-instance`.

AWS CLI

Um die Zuweisung einer registrierten Instanz zu ihren Layern aufzuheben

Mit dem folgenden `unassign-instance` Befehl wird die Zuweisung einer Instanz zu ihren angehängten Ebenen aufgehoben.

```
aws opsworks --region us-east-1 unassign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```


Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Aufheben der Zuweisung einer registrierten Instanz im AWS OpsWorks Benutzerhandbuch.

- APIEinzelheiten finden Sie [UnassignInstance](#) in der AWS CLI Befehlsreferenz.

unassign-volume

Das folgende Codebeispiel zeigt die Verwendung `unassign-volume`.

AWS CLI

Um die Zuweisung eines Volumes zu seiner Instanz aufzuheben

Im folgenden Beispiel wird die Zuweisung eines registrierten Amazon Elastic Block Store (AmazonEBS) -Volumes zu seiner Instance aufgehoben. Das Volume wird anhand seiner Volume-ID identifiziert, die AWS OpsWorks zugewiesen wird, wenn Sie GUID das Volume bei einem Stack registrieren, und nicht anhand der Volume-ID von Amazon Elastic Compute Cloud (AmazonEC2).

```
aws opsworks --region us-east-1 unassign-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Unassigning Amazon EBS Volumes im AWS OpsWorks Benutzerhandbuch.

- APIEinzelheiten finden Sie [UnassignVolume](#) in der AWS CLI Befehlsreferenz.

update-app

Das folgende Codebeispiel zeigt die Verwendung `update-app`.

AWS CLI

Um eine App zu aktualisieren

Im folgenden Beispiel wird eine angegebene App aktualisiert, um ihren Namen zu ändern.

```
aws opsworks --region us-east-1 update-app --app-id 26a61ead-d201-47e3-b55c-2a7c666942f8 --name NewAppName
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Bearbeiten von Apps.

- APIEinzelheiten finden Sie [UpdateApp](#) in der AWS CLI Befehlsreferenz.

update-elastic-ip

Das folgende Codebeispiel zeigt die Verwendung `update-elastic-ip`.

AWS CLI

Um den Namen einer Elastic IP-Adresse zu aktualisieren

Im folgenden Beispiel wird der Name einer angegebenen Elastic IP-Adresse aktualisiert.

```
aws opsworks --region us-east-1 update-elastic-ip --elastic-ip 54.148.130.96 --name NewIPName
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Ressourcenverwaltung.

- APIEinzelheiten finden Sie [UpdateElasticIp](#) in der AWS CLI Befehlsreferenz.

update-instance

Das folgende Codebeispiel zeigt die Verwendung `update-instance`.

AWS CLI

Um eine Instanz zu aktualisieren

Im folgenden Beispiel wird der Typ einer angegebenen Instanz aktualisiert.

```
aws opsworks --region us-east-1 update-instance --instance-  
id dfc18b02-5327-493d-91a4-c5c0c448927f --instance-type c3.xlarge
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Bearbeiten der Instanzkonfiguration im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateInstance](#) in der AWS CLI Befehlsreferenz.

update-layer

Das folgende Codebeispiel zeigt die Verwendung `update-layer`.

AWS CLI

Um eine Ebene zu aktualisieren

Im folgenden Beispiel wird eine angegebene Ebene aktualisiert, um EBS Amazon-optimierte Instanzen zu verwenden.

```
aws opsworks --region us-east-1 update-layer --layer-  
id 888c5645-09a5-4d0e-95a8-812ef1db76a4 --use-ebs-optimized-instances
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter Bearbeiten der OpsWorks Layer-Konfiguration.

- API-Einheiten finden Sie [UpdateLayer](#) in der AWS CLI Befehlsreferenz.

update-my-user-profile

Das folgende Codebeispiel zeigt die Verwendung `update-my-user-profile`.

AWS CLI

Um das Profil eines Benutzers zu aktualisieren

Im folgenden Beispiel wird das development Benutzerprofil aktualisiert, sodass es einen angegebenen SSH öffentlichen Schlüssel verwendet. Die AWS Anmeldeinformationen des Benutzers werden durch das development Profil in der credentials Datei (~\.aws\credentials) dargestellt, und der Schlüssel befindet sich in einer .pem Datei im Arbeitsverzeichnis.

```
aws opsworks --region us-east-1 --profile development update-my-user-profile --ssh-public-key file://development_key.pem
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter AWS OpsWorks Benutzereinstellungen bearbeiten.

- APIEinzelheiten finden Sie [UpdateMyUserProfile](#) unter AWS CLI Befehlsreferenz.

update-rds-db-instance

Das folgende Codebeispiel zeigt die Verwendung `update-rds-db-instance`.

AWS CLI

Um eine registrierte Amazon RDS DB-Instance zu aktualisieren

Im folgenden Beispiel wird der Master-Passwortwert einer RDS Amazon-Instance aktualisiert. Beachten Sie, dass dieser Befehl nicht das Master-Passwort der RDS Instance ändert, sondern nur das Passwort, das Sie angeben AWS OpsWorks. Wenn dieses Passwort nicht mit dem Passwort der RDS Instanz übereinstimmt, kann Ihre Anwendung keine Verbindung zur Datenbank herstellen.

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

Ausgabe: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Registrierung von RDS Amazon-Instances mit einem Stack im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRdsDbInstance](#) in der AWS CLI Befehlsreferenz.

update-volume

Das folgende Codebeispiel zeigt die Verwendung `update-volume`.

AWS CLI

Um ein registriertes Volume zu aktualisieren

Das folgende Beispiel aktualisiert den Mountpoint eines registrierten Amazon Elastic Block Store (AmazonEBS) -Volumes. Das Volume wird anhand seiner Volume-ID identifiziert, die dem Volume AWS OpsWorks zugewiesen wird, wenn Sie es bei einem Stack registrieren, und nicht anhand der Volume-ID von Amazon Elastic Compute Cloud (AmazonEC2). GUID :

```
aws opsworks --region us-east-1 update-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df --mount-point /mnt/myvol
```

Ausgang: Keine.

Mehr Informationen

Weitere Informationen finden Sie unter Zuweisen von EBS Amazon-Volumes zu einer Instance im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateVolume](#) in der AWS CLI Befehlsreferenz.

AWS OpsWorks CM Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS OpsWorks CM.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-node

Das folgende Codebeispiel zeigt die Verwendung `associate-node`.

AWS CLI

Um Knoten zuzuordnen

Der folgende `associate-node` Befehl ordnet einen Knoten `i-44de882p` mit einem Chef Automate-Server mit dem Namen `zuautomate-06`, was bedeutet, dass der `automate-06` Server den Knoten verwaltet und Rezeptbefehle über die `chef-client` Agentsoftware, die mit dem Befehl `associate-node` auf dem Knoten installiert wird, an den Knoten übermittelt. Gültige Knotennamen sind Instanz. EC2 IDs :

```
aws opsworks-cm associate-node --server-name "automate-06" --node-name "i-43de882p"
--engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'
Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

Die vom Befehl zurückgegebene Ausgabe ähnelt der folgenden. Ausgabe:

```
{
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"
}
```

Weitere Informationen

Weitere Informationen finden Sie unter [Automatisches Hinzufügen von Knoten in Chef Automate im AWS OpsWorks Benutzerhandbuch](#). AWS OpsWorks

- API-Einheiten finden Sie [AssociateNode](#) in der AWS CLI Befehlsreferenz.

create-backup

Das folgende Codebeispiel zeigt die Verwendung `create-backup`.

AWS CLI

Um Backups zu erstellen

Der folgende `create-backup` Befehl startet eine manuelle Sicherung eines Chef Automate-Servers, der `automate-06` in der `us-east-1` Region benannt ist. Der Befehl fügt dem Backup im `--description` Parameter eine beschreibende Nachricht hinzu.

```
aws opsworks-cm create-backup \  
  --server-name 'automate-06' \  
  --description "state of my infrastructure at launch"
```

In der Ausgabe werden Ihnen Informationen über das neue Backup angezeigt, die den folgenden ähneln.

Ausgabe:

```
{  
  "Backups": [  
    {  
      "BackupArn": "string",  
      "BackupId": "automate-06-20160729133847520",  
      "BackupType": "MANUAL",  
      "CreatedAt": "2016-07-29T13:38:47.520Z",  
      "Description": "state of my infrastructure at launch",  
      "Engine": "Chef",  
      "EngineModel": "Single",  
      "EngineVersion": "12",  
      "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/  
automate-06-1010V4UU2WRM2",  
      "InstanceType": "m4.large",  
      "KeyPair": "",  
      "PreferredBackupWindow": "",  
      "PreferredMaintenanceWindow": "",  
      "S3LogUrl": "https://s3.amazonaws.com/<bucket-name>/  
automate-06-20160729133847520",  
      "SecurityGroupIds": [ "sg-1a24c270" ],  
      "ServerName": "automate-06",  
      "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-  
service-role.1114810729735",  
      "Status": "OK",  
      "StatusDescription": "",  
      "SubnetIds": [ "subnet-49436a18" ],  
      "ToolsVersion": "string",  
      "UserArn": "arn:aws:iam::1019881987024:user/opsworks-user"  
    }  
  ],  
}
```

```
}
```

Weitere Informationen finden Sie unter [Sichern und Wiederherstellen eines AWS OpsWorks for Chef Automate-Servers](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [CreateBackup](#) unter AWS CLI Befehlsreferenz.

create-server

Das folgende Codebeispiel zeigt die Verwendung `create-server`.

AWS CLI

Um einen Server zu erstellen

Das folgende `create-server` Beispiel erstellt einen neuen Chef Automate-Server mit `automate-06` dem Namen Ihrer Standardregion. Beachten Sie, dass Standardwerte für die meisten anderen Einstellungen verwendet werden, z. B. für die Anzahl der aufzubewahrenden Backups sowie für die Startzeiten für Wartung und Sicherung. Bevor Sie einen `create-server` Befehl ausführen, müssen Sie die Voraussetzungen unter [Erste Schritte mit AWS OpsWorks für Chef Automate](#) im AWS Opsworks for Chef Automate-Benutzerhandbuch erfüllen.

```
aws opsworks-cm create-server \
  --engine "Chef" \
  --engine-model "Single" \
  --engine-version "12" \
  --server-name "automate-06" \
  --instance-profile-arn "arn:aws:iam::1019881987024:instance-profile/aws-opsworks-cm-ec2-role" \
  --instance-type "t2.medium" \
  --key-pair "amazon-test" \
  --service-role-arn "arn:aws:iam::044726508045:role/aws-opsworks-cm-service-role"
```

In der Ausgabe werden Ihnen Informationen über den neuen Server angezeigt, die den folgenden ähneln:

```
{
  "Server": {
    "BackupRetentionCount": 10,
    "CreatedAt": 2016-07-29T13:38:47.520Z,
    "DisableAutomatedBackup": FALSE,
    "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
```



```

    "Engine": "Chef",
    "EngineAttributes": [
      {
        "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
        "Value": "1Password1"
      }
    ],
    "EngineModel": "Single",
    "EngineVersion": "12",
    "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/aws-opsworks-cm-ec2-role",
    "InstanceType": "t2.medium",
    "KeyPair": "amazon-test",
    "MaintenanceStatus": "",
    "PreferredBackupWindow": "Sun:02:00",
    "PreferredMaintenanceWindow": "00:00",
    "SecurityGroupIds": [ "sg-1a24c270" ],
    "ServerArn": "arn:aws:iam::1019881987024:instance/automate-06-1010V4UU2WRM2",
    "ServerName": "automate-06",
    "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-role",
    "Status": "CREATING",
    "StatusReason": "",
    "SubnetIds": [ "subnet-49436a18" ]
  }
}

```

Weitere Informationen finden Sie [UpdateServer](#) in der AWS OpsWorks for Chef API Automate-Referenz.

- API-Einheiten finden Sie [CreateServer](#) unter AWS CLI Befehlsreferenz.

delete-backup

Das folgende Codebeispiel zeigt die Verwendung `delete-backup`.

AWS CLI

Um Backups zu löschen

Der folgende `delete-backup` Befehl löscht eine manuelle oder automatische Sicherung eines Chef Automate-Servers, der durch die Backup-ID identifiziert wird. Dieser Befehl ist nützlich, wenn

Sie sich der maximalen Anzahl von Backups nähern, die Sie speichern können, oder wenn Sie Ihre Amazon S3 S3-Speicherkosten minimieren möchten. :

```
aws opsworks-cm delete-backup --backup-id "automate-06-2016-11-19T23:42:40.240Z"
```

Die Ausgabe zeigt, ob das Löschen des Backups erfolgreich war.

Weitere Informationen

Weitere Informationen finden Sie unter Sichern und Wiederherstellen eines AWS OpsWorks für Chef Automate Server im AWS OpsWorks Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteBackup](#) unter AWS CLI Befehlsreferenz.

delete-server

Das folgende Codebeispiel zeigt die Verwendung `delete-server`.

AWS CLI

Um Server zu löschen

Der folgende `delete-server` Befehl löscht einen Chef Automate-Server, der durch den Namen des Servers identifiziert wird. Nachdem der Server gelöscht wurde, wird er nicht mehr durch `DescribeServer` Anfragen zurückgegeben. :

```
aws opsworks-cm delete-server --server-name "automate-06"
```

Die Ausgabe zeigt, ob das Löschen des Servers erfolgreich war.

Weitere Informationen

Weitere Informationen finden Sie unter Löschen eines AWS OpsWorks for Chef Automate Servers im AWS OpsWorks Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteServer](#) unter AWS CLI Befehlsreferenz.

describe-account-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-account-attributes`.

AWS CLI

Um Kontoattribute zu beschreiben

Der folgende `describe-account-attributes` Befehl gibt Informationen über die Nutzung von AWS OpsWorks For Chef Automate-Ressourcen durch Ihr Konto zurück. :

```
aws opsworks-cm describe-account-attributes
```

Die Ausgabe für jeden vom Befehl zurückgegebenen Kontoattributeintrag ähnelt der folgenden.
Ausgabe:

```
{
  "Attributes": [
    {
      "Maximum": 5,
      "Name": "ServerLimit",
      "Used": 2
    }
  ]
}
```

Weitere Informationen

Weitere Informationen finden Sie `DescribeAccountAttributes` in der AWS OpsWorks for Chef API Automate-Referenz.

- APIEinzelheiten finden Sie [DescribeAccountAttributes](#) unter AWS CLI Befehlsreferenz.

describe-backups

Das folgende Codebeispiel zeigt die Verwendung `describe-backups`.

AWS CLI

Um Backups zu beschreiben

Der folgende `describe-backups` Befehl gibt Informationen zu allen Backups zurück, die mit Ihrem Konto in Ihrer Standardregion verknüpft sind.

```
aws opsworks-cm describe-backups
```

Die Ausgabe für jeden vom Befehl zurückgegebenen Backup-Eintrag ähnelt der folgenden.

Ausgabe:

```
{
  "Backups": [
    {
      "BackupArn": "string",
      "BackupId": "automate-06-20160729133847520",
      "BackupType": "MANUAL",
      "CreatedAt": 2016-07-29T13:38:47.520Z,
      "Description": "state of my infrastructure at launch",
      "Engine": "Chef",
      "EngineModel": "Single",
      "EngineVersion": "12",
      "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
      "InstanceType": "m4.large",
      "KeyPair": "",
      "PreferredBackupWindow": "",
      "PreferredMaintenanceWindow": "",
      "S3LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
      "SecurityGroupIds": [ "sg-1a24c270" ],
      "ServerName": "automate-06",
      "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-
service-role.1114810729735",
      "Status": "Successful",
      "StatusDescription": "",
      "SubnetIds": [ "subnet-49436a18" ],
      "ToolsVersion": "string",
      "UserArn": "arn:aws:iam::1019881987024:user/opsworks-user"
    }
  ],
}
```

Weitere Informationen finden Sie unter [Sichern und Wiederherstellen eines AWS OpsWorks for Chef Automate-Servers](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeBackups](#) unter AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um Ereignisse zu beschreiben

Das folgende `describe-events` Beispiel gibt Informationen zu allen Ereignissen zurück, die dem angegebenen Chef Automate-Server zugeordnet sind.

```
aws opsworks-cm describe-events \
  --server-name 'automate-06'
```

Die Ausgabe für jeden vom Befehl zurückgegebenen Ereigniseintrag ähnelt dem folgenden Beispiel:

```
{
  "ServerEvents": [
    {
      "CreatedAt": 2016-07-29T13:38:47.520Z,
      "LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
      "Message": "Updates successfully installed.",
      "ServerName": "automate-06"
    }
  ]
}
```

Weitere Informationen finden Sie im AWS OpsWorks Benutzerhandbuch unter [Allgemeine Tipps zur Problembehandlung](#).

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

describe-node-association-status

Das folgende Codebeispiel zeigt die Verwendung `describe-node-association-status`.

AWS CLI

Um den Status der Knotenzuweisung zu beschreiben

Der folgende `describe-node-association-status` Befehl gibt den Status einer Anfrage zurück, einen Knoten einem Chef Automate-Server mit dem Namen zuzuordnen `automate-06`. :

```
aws opsworks-cm describe-node-association-status --server-  
name "automate-06" --node-association-status-token "AFLJKL+/  
GoKLZJBdDQEx0065CDi57b1Qe9nKM8joSok0pQ9xr8DqApBN9/106sLdSvlfDEKkEx+eoCHvj0WHa0s="
```

Die Ausgabe für jeden vom Befehl zurückgegebenen Kontoattributeintrag ähnelt der folgenden. Ausgabe:

```
{  
  "NodeAssociationStatus": "IN_PROGRESS"  
}
```

Weitere Informationen

Weitere Informationen finden Sie `DescribeNodeAssociationStatus` in der AWS OpsWorks for Chef API Automate-Referenz.

- API-Einheiten finden Sie [DescribeNodeAssociationStatus](#) unter AWS CLI Befehlsreferenz.

describe-servers

Das folgende Codebeispiel zeigt die Verwendung `describe-servers`.

AWS CLI

Um Server zu beschreiben

Der folgende `describe-servers` Befehl gibt Informationen zu allen Servern zurück, die Ihrem Konto zugeordnet sind, und zwar in Ihrer Standardregion. :

```
aws opsworks-cm describe-servers
```

Die Ausgabe für jeden vom Befehl zurückgegebenen Servereintrag ähnelt der folgenden. Ausgabe:

```
{  
  "Servers": [  
    {
```

```

{
  "BackupRetentionCount": 8,
  "CreatedAt": "2016-07-29T13:38:47.520Z",
  "DisableAutomatedBackup": FALSE,
  "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
  "Engine": "Chef",
  "EngineAttributes": [
    {
      "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
      "Value": "1Password1"
    }
  ],
  "EngineModel": "Single",
  "EngineVersion": "12",
  "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
  "InstanceType": "m4.large",
  "KeyPair": "",
  "MaintenanceStatus": "SUCCESS",
  "PreferredBackupWindow": "03:00",
  "PreferredMaintenanceWindow": "Mon:09:00",
  "SecurityGroupIds": [ "sg-1a24c270" ],
  "ServerArn": "arn:aws:iam::1019881987024:instance/automate-06-1010V4UU2WRM2",
  "ServerName": "automate-06",
  "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
  "Status": "HEALTHY",
  "StatusReason": "",
  "SubnetIds": [ "subnet-49436a18" ]
}
]
}

```

Weitere Informationen

Weitere Informationen finden Sie `DescribeServers` im AWS OpsWorks for Chef Automate API Guide.

- API-Einheiten finden Sie [DescribeServers](#) in der AWS CLI Befehlsreferenz.

disassociate-node

Das folgende Codebeispiel zeigt die Verwendung `disassociate-node`.

AWS CLI

Um Knoten zu trennen

Der folgende `disassociate-node` Befehl trennt einen Knoten mit dem Namen `i-44de882p` und entfernt den Knoten aus der Verwaltung durch einen Chef Automate-Server mit dem Namen `automate-06`. Gültige Knotennamen sind EC2 InstanzIDs. :

```
aws opsworks-cm disassociate-node --server-name "automate-06" --node-  
name "i-43de882p" --engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'  
Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

Die vom Befehl zurückgegebene Ausgabe ähnelt der folgenden. Ausgabe:

```
{  
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//  
rHRqHDWXxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"  
}
```

Weitere Informationen

Weitere Informationen finden Sie unter Löschen eines AWS OpsWorks for Chef Automate Servers im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateNode](#) unter AWS CLI Befehlsreferenz.

restore-server

Das folgende Codebeispiel zeigt die Verwendung `restore-server`.

AWS CLI

Um einen Server wiederherzustellen

Der folgende `restore-server` Befehl führt eine direkte Wiederherstellung eines Chef Automate-Servers mit dem Namen `automate-06` in Ihrer Standardregion aus einem Backup mit der ID von `Automate-06-2016-11-22T16:13:27.998Z`. Durch das Wiederherstellen eines Servers werden Verbindungen zu den Knoten wiederhergestellt, die der Chef Automate-Server zum Zeitpunkt der Durchführung der angegebenen Sicherung verwaltete.

```
aws opsworks-cm restore-server --backup-id „Automate-06-2016-11-22T 16:13:27.998 Z“ --  
servername „automate-06“
```


Die Ausgabe ist nur die Befehls-ID. Ausgabe:

```
(None)
```

Weitere Informationen

Weitere Informationen finden Sie unter [Restore a Failed AWS OpsWorks for Chef Automate Server](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [RestoreServer](#) in der AWS CLI Befehlsreferenz.

start-maintenance

Das folgende Codebeispiel zeigt die Verwendung `start-maintenance`.

AWS CLI

Um die Wartung zu starten

Im folgenden `start-maintenance` Beispiel wird die Wartung auf dem angegebenen Chef Automate- oder Puppet Enterprise-Server in Ihrer Standardregion manuell gestartet. Dieser Befehl ist nützlich, wenn ein früherer, automatisierter Wartungsversuch fehlgeschlagen ist und die zugrunde liegende Ursache des Wartungsfehlers behoben wurde.

```
aws opsworks-cm start-maintenance \  
  --server-name 'automate-06'
```

Ausgabe:

```
{  
  "Server": {  
    "AssociatePublicIpAddress": true,  
    "BackupRetentionCount": 10,  
    "ServerName": "automate-06",  
    "CreatedAt": 1569229584.842,  
    "CloudFormationStackArn": "arn:aws:cloudformation:us-  
west-2:123456789012:stack/aws-opsworks-cm-instance-automate-06-1606611794746/  
EXAMPLE0-31de-11eb-bdb0-0a5b0a1353b8",  
    "DisableAutomatedBackup": false,  
    "Endpoint": "automate-06-EXAMPLEv1r8gjfk5f.us-west-2.opsworks-cm.io",  
    "Engine": "ChefAutomate",
```

```

    "EngineModel": "Single",
    "EngineAttributes": [],
    "EngineVersion": "2020-07",
    "InstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/aws-opsworks-cm-ec2-role",
    "InstanceType": "m5.large",
    "PreferredMaintenanceWindow": "Sun:01:00",
    "PreferredBackupWindow": "Sun:15:00",
    "SecurityGroupIds": [
        "sg-EXAMPLE"
    ],
    "ServiceRoleArn": "arn:aws:iam::123456789012:role/service-role/aws-opsworks-cm-service-role",
    "Status": "UNDER_MAINTENANCE",
    "SubnetIds": [
        "subnet-EXAMPLE"
    ],
    "ServerArn": "arn:aws:opsworks-cm:us-west-2:123456789012:server/automate-06/0148382d-66b0-4196-8274-d1a2b6dff8d1"
}
}

```

Weitere Informationen finden Sie unter [Systemwartung \(Puppet Enterprise-Server\)](#) oder [Systemwartung \(Chef Automate-Server\)](#) im AWS OpsWorks Benutzerhandbuch.

- API-Einheiten finden Sie [StartMaintenance](#) in der AWS CLI Befehlsreferenz.

update-server-engine-attributes

Das folgende Codebeispiel zeigt die Verwendung `update-server-engine-attributes`.

AWS CLI

Um die Attribute der Server-Engine zu aktualisieren

Der folgende `update-server-engine-attributes` Befehl aktualisiert den Wert des `CHEF_PIVOTAL_KEY` Engine-Attributs für einen Chef Automate-Server mit dem Namen `automate-06`. Es ist derzeit nicht möglich, den Wert anderer Engine-Attribute zu ändern.

```

aws opsworks-cm update-server-engine-attributes \
  --attribute-name CHEF_PIVOTAL_KEY \
  --attribute-value "new key value" \

```

```
--server-name "automate-06"
```

In der Ausgabe werden Informationen über den aktualisierten Server angezeigt, die den folgenden ähneln.

```
{
  "Server": {
    "BackupRetentionCount": 2,
    "CreatedAt": 2016-07-29T13:38:47.520Z,
    "DisableAutomatedBackup": FALSE,
    "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
    "Engine": "Chef",
    "EngineAttributes": [
      {
        "Name": "CHEF_PIVOTAL_KEY",
        "Value": "new key value"
      }
    ],
    "EngineModel": "Single",
    "EngineVersion": "12",
    "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
    "InstanceType": "m4.large",
    "KeyPair": "",
    "MaintenanceStatus": "SUCCESS",
    "PreferredBackupWindow": "Mon:09:15",
    "PreferredMaintenanceWindow": "03:00",
    "SecurityGroupIds": [ "sg-1a24c270" ],
    "ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
    "ServerName": "automate-06",
    "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
    "Status": "HEALTHY",
    "StatusReason": "",
    "SubnetIds": [ "subnet-49436a18" ]
  }
}
```

Weitere Informationen finden Sie [UpdateServerEngineAttributes](#) in der AWS OpsWorks for Chef API Automate-Referenz.

- API-Einheiten finden Sie [UpdateServerEngineAttributes](#) unter AWS CLI Befehlsreferenz.

update-server

Das folgende Codebeispiel zeigt die Verwendung `update-server`.

AWS CLI

Um einen Server zu aktualisieren

Der folgende `update-server` Befehl aktualisiert die Startzeit der Wartung des angegebenen Chef Automate-Servers in Ihrer Standardregion. Der `--preferred-maintenance-window` Parameter wurde hinzugefügt, um den Starttag und die Uhrzeit der Serverwartung auf Montag um 9:15 Uhr zu ändern. UTC. :

```
aws opsworks-cm update-server \  
  --server-name "automate-06" \  
  --preferred-maintenance-window "Mon:09:15"
```

In der Ausgabe werden Informationen über den aktualisierten Server angezeigt, die den folgenden ähneln.

```
{  
  "Server": {  
    "BackupRetentionCount": 8,  
    "CreatedAt": "2016-07-29T13:38:47.520Z",  
    "DisableAutomatedBackup": TRUE,  
    "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",  
    "Engine": "Chef",  
    "EngineAttributes": [  
      {  
        "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",  
        "Value": "1Password1"  
      }  
    ],  
    "EngineModel": "Single",  
    "EngineVersion": "12",  
    "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/  
automate-06-1010V4UU2WRM2",  
    "InstanceType": "m4.large",  
    "KeyPair": "",  
    "MaintenanceStatus": "OK",  
    "PreferredBackupWindow": "Mon:09:15",  
    "PreferredMaintenanceWindow": "03:00",
```

```
    "SecurityGroupIds": [ "sg-1a24c270" ],
    "ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
    "ServerName": "automate-06",
    "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
    "Status": "HEALTHY",
    "StatusReason": "",
    "SubnetIds": [ "subnet-49436a18" ]
  }
}
```

Weitere Informationen finden Sie [UpdateServer](#) in der AWS OpsWorks for Chef API Automate-Referenz.

- API-Einheiten finden Sie [UpdateServer](#) unter AWS CLI Befehlsreferenz.

Beispiele für Organizations, die AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Organizations Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-handshake

Das folgende Codebeispiel zeigt die Verwendung `accept-handshake`.

AWS CLI

Um einen Handshake von einem anderen Konto anzunehmen

Bill, der Inhaber einer Organisation, hat Juans Account bereits früher eingeladen, seiner Organisation beizutreten. Das folgende Beispiel zeigt, wie Juans Konto den Handschlag akzeptiert und damit der Einladung zustimmt.

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

Die Ausgabe zeigt Folgendes:

```
{
  "Handshake": {
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "RequestedTimestamp": 1481656459.257,
    "ExpirationTimestamp": 1482952459.257,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "juan@example.com",
        "Type": "EMAIL"
      }
    ],
    "Resources": [
      {
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@amazon.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Org Master Account"
          },
          {
            "Type": "ORGANIZATION_FEATURE_SET",
            "Value": "ALL"
          }
        ],
        "Type": "ORGANIZATION",

```

```

    "Value": "o-exampleorgid"
  },
  {
    "Type": "EMAIL",
    "Value": "juan@example.com"
  }
],
"State": "ACCEPTED"
}
}

```

- API-Einheiten finden Sie [AcceptHandshake](#) in der AWS CLI Befehlsreferenz.

attach-policy

Das folgende Codebeispiel zeigt die Verwendung `attach-policy`.

AWS CLI

Um eine Richtlinie an ein Root-, OU- oder Konto anzuhängen

Beispiel 1

Das folgende Beispiel zeigt, wie eine Dienststeuerungsrichtlinie (SCP) an eine Organisationseinheit angehängt wird:

```

aws organizations attach-policy
  --policy-id p-examplepolicyid111
  --target-id ou-examplerootid111-exampleouid111

```

Beispiel 2

Das folgende Beispiel zeigt, wie eine Dienststeuerungsrichtlinie direkt an ein Konto angehängt wird:

```

aws organizations attach-policy
  --policy-id p-examplepolicyid111
  --target-id 333333333333

```

- API-Einheiten finden Sie [AttachPolicy](#) in der AWS CLI Befehlsreferenz.

cancel-handshake

Das folgende Codebeispiel zeigt die Verwendung `cancel-handshake`.

AWS CLI

Um einen Handshake zu stornieren, der von einem anderen Konto gesendet wurde

Bill hat zuvor eine Einladung an Susans Konto gesendet, seiner Organisation beizutreten. Er ändert seine Meinung und beschließt, die Einladung zu stornieren, bevor Susan sie annimmt. Das folgende Beispiel zeigt Bills Stornierung:

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

Die Ausgabe enthält ein Handshake-Objekt, das anzeigt, dass der Status jetzt CANCELED wie folgt lautet:

```
{
  "Handshake": {
    "Id": "h-examplehandshakeid111",
    "State": "CANCELED",
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "susan@example.com",
        "Type": "EMAIL"
      }
    ],
    "Resources": [
      {
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid",
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          }
        ]
      }
    ]
  }
}
```



```

        },
        {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
        },
        {
            "Type": "ORGANIZATION_FEATURE_SET",
            "Value": "CONSOLIDATED_BILLING"
        }
    ]
},
{
    "Type": "EMAIL",
    "Value": "anika@example.com"
},
{
    "Type": "NOTES",
    "Value": "This is a request for Susan's account to
join Bob's organization."
}
],
"RequestedTimestamp": 1.47008383521E9,
"ExpirationTimestamp": 1.47137983521E9
}
}

```

- API-Einheiten finden Sie [CancelHandshake](#) in der AWS CLI Befehlsreferenz.

create-account

Das folgende Codebeispiel zeigt die Verwendung `create-account`.

AWS CLI

Um ein Mitgliedskonto zu erstellen, das automatisch Teil der Organisation ist

Das folgende Beispiel zeigt, wie Sie ein Mitgliedskonto in einer Organisation erstellen.

Das Mitgliedskonto ist mit dem Namen `Production Account` und der E-Mail-Adresse `susan@example.com` konfiguriert. Organisationen erstellen automatisch eine IAM-Rolle mit dem Standardnamen `OrganizationAccountAccessRole`, weil der `roleName`-Parameter nicht angegeben ist. Außerdem ist die Einstellung, die IAM-Benutzern oder Rollen mit ausreichenden Berechtigungen den Zugriff auf Kontoabrechnungsdaten ermöglicht, auf den Standardwert von

gesetzt `ALLOW`, da der `IamUserAccessToBilling` Parameter nicht angegeben ist. Organizations sendet Susan automatisch eine „Willkommen bei AWS“-E-Mail:

```
aws organizations create-account --email susan@example.com --account-name "Production Account"
```

Die Ausgabe enthält ein Anforderungsobjekt, aus dem hervorgeht, dass der Status jetzt wie folgt lautet `IN_PROGRESS`:

```
{
  "CreateAccountStatus": {
    "State": "IN_PROGRESS",
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

Sie können später den aktuellen Status der Anforderung abfragen, indem Sie den Antwortwert ID für den `describe-create-account-status` Befehl als Wert für den `create-account-request-id` Parameter angeben.

Weitere Informationen finden Sie unter Erstellen eines AWS Kontos in Ihrer Organisation im Benutzerhandbuch für AWS Organizations.

- API-Einheiten finden Sie [CreateAccount](#) in der AWS CLI Befehlsreferenz.

create-organization

Das folgende Codebeispiel zeigt die Verwendung `create-organization`.

AWS CLI

Beispiel 1: Um eine neue Organisation zu erstellen

Bill möchte eine Organisation mit den Anmeldeinformationen des Kontos 111111111111 erstellen. Das folgende Beispiel zeigt, dass das Konto zum Hauptkonto in der neuen Organisation wird. Da er keinen Funktionsumfang angibt, sind in der neuen Organisation standardmäßig alle Funktionen aktiviert, und die Richtlinien zur Dienststeuerung sind im Stammverzeichnis aktiviert.

```
aws organizations create-organization
```

Die Ausgabe umfasst ein Organisationsobjekt mit Details zur neuen Organisation:

```
{
  "Organization": {
    "AvailablePolicyTypes": [
      {
        "Status": "ENABLED",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    ],
    "MasterAccountId": "111111111111",
    "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
    "MasterAccountEmail": "bill@example.com",
    "FeatureSet": "ALL",
    "Id": "o-exampleorgid",
    "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid"
  }
}
```

Beispiel 2: Um eine neue Organisation zu erstellen, für die nur konsolidierte Fakturierungsfunktionen aktiviert sind

Im folgenden Beispiel wird eine Organisation erstellt, die nur die Funktionen für die konsolidierte Fakturierung unterstützt:

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

Die Ausgabe enthält ein Organisationsobjekt mit Details zur neuen Organisation:

```
{
  "Organization": {
    "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
    "AvailablePolicyTypes": [],
    "Id": "o-exampleorgid",
    "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
    "MasterAccountEmail": "bill@example.com",
    "MasterAccountId": "111111111111",
    "FeatureSet": "CONSOLIDATED_BILLING"
  }
}
```

Weitere Informationen finden Sie unter [Creating a Organization](#) im [AWS Organizations Users Guide](#).

- API-Einheiten finden Sie [CreateOrganization](#) in der AWS CLI Befehlsreferenz.

create-organizational-unit

Das folgende Codebeispiel zeigt die Verwendung `create-organizational-unit`.

AWS CLI

Um eine Organisationseinheit in einer Stamm- oder übergeordneten Organisationseinheit zu erstellen

Das folgende Beispiel zeigt, wie eine Organisationseinheit mit dem Namen `AccountingOU` erstellt wird:

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --  
name AccountingOU
```

Die Ausgabe enthält ein `organizationalUnit` Objekt mit Details zur neuen Organisationseinheit:

```
{  
  "OrganizationalUnit": {  
    "Id": "ou-examplerootid111-exampleoid111",  
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-  
examplerootid111-exampleoid111",  
    "Name": "AccountingOU"  
  }  
}
```

- API-Einheiten finden Sie [CreateOrganizationalUnit](#) in der AWS CLI Befehlsreferenz.

create-policy

Das folgende Codebeispiel zeigt die Verwendung `create-policy`.

AWS CLI

Beispiel 1: Um eine Richtlinie mit einer Text Quelldatei für die JSON Richtlinie zu erstellen

Das folgende Beispiel zeigt Ihnen, wie Sie eine Dienststeuerungsrichtlinie (SCP) mit dem Namen `AllowAllS3Actions` erstellen. Der Richtlinieninhalt stammt aus einer Datei auf dem lokalen Computer mit dem Namen `policy.json`.

```
aws organizations create-policy --content file://policy.json --
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
delegation of all S3 actions"
```

Die Ausgabe enthält ein Richtlinienobjekt mit Details zur neuen Richtlinie:

```
{
  "Policy": {
    "Content": "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\":
\\\"Allow\\\", \"Action\": [\\\"s3:*\\\"], \"Resource\": [\\\"*\\\"]}]}",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid111",
      "Description": "Allows delegation of all S3 actions",
      "Name": "AllowAllS3Actions",
      "Type": "SERVICE_CONTROL_POLICY"
    }
  }
}
```

Beispiel 2: Um eine Richtlinie mit einer JSON Richtlinie als Parameter zu erstellen

Das folgende Beispiel zeigt Ihnen, wie Sie dasselbe erstellen SCP, diesmal indem Sie den Richtlinieninhalt als JSON Zeichenfolge in den Parameter einbetten. Die Zeichenfolge muss mit umgekehrten Schrägstrichen vor den doppelten Anführungszeichen maskiert werden, um sicherzustellen, dass sie im Parameter, der selbst von doppelten Anführungszeichen umgeben ist, als Literale behandelt werden:

```
aws organizations create-policy --content "{\"Version\":\"2012-10-17\", \"Statement
\": [{\"Effect\": \"Allow\", \"Action\": [\\\"s3:*\\\"], \"Resource\": [\\\"*\\\"]}]}" --
name AllowAllS3Actions --type SERVICE_CONTROL_POLICY --description "Allows
delegation of all S3 actions"
```

Weitere Informationen zum Erstellen und Verwenden von Richtlinien in Ihrer Organisation finden Sie unter Verwaltung von Organisationsrichtlinien im AWS Organizations User Guide.

- API-Einheiten finden Sie [CreatePolicy](#) in der AWS CLI Befehlsreferenz.

decline-handshake

Das folgende Codebeispiel zeigt die Verwendung `decline-handshake`.

AWS CLI

Um einen Handshake abzulehnen, der von einem anderen Konto gesendet wurde

Das folgende Beispiel zeigt, dass Susan, eine Administratorin, die Eigentümerin des Kontos 222222222222 ist, eine Einladung ablehnt, Bills Organisation beizutreten. Der `DeclineHandshake` Vorgang gibt ein Handshake-Objekt zurück, das anzeigt, dass der Status jetzt wie folgt lautet: `DECLINED`

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

Die Ausgabe enthält ein Handshake-Objekt, das den neuen Status von `DECLINED` anzeigt:

```
{
  "Handshake": {
    "Id": "h-examplehandshakeid111",
    "State": "DECLINED",
    "Resources": [
      {
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid",
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
          }
        ]
      },
      {
        "Type": "EMAIL",
        "Value": "susan@example.com"
      },
      {
        "Type": "NOTES",
```

```

        "Value": "This is an invitation to Susan's account
to join the Bill's organization."
    },
    "Parties": [
        {
            "Type": "EMAIL",
            "Id": "susan@example.com"
        },
        {
            "Type": "ORGANIZATION",
            "Id": "o-exampleorgid"
        }
    ],
    "Action": "INVITE",
    "RequestedTimestamp": 1470684478.687,
    "ExpirationTimestamp": 1471980478.687,
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
}
}

```

- API-Einheiten finden Sie [DeclineHandshake](#) in der AWS CLI Befehlsreferenz.

delete-organization

Das folgende Codebeispiel zeigt die Verwendung `delete-organization`.

AWS CLI

Um eine Organisation zu löschen

Das folgende Beispiel zeigt, wie eine Organisation gelöscht wird. Um diesen Vorgang ausführen zu können, müssen Sie Administrator des Hauptkontos in der Organisation sein. Das Beispiel geht davon aus, dass Sie zuvor alle Mitgliedskonten OUs und Richtlinien aus der Organisation entfernt haben:

```
aws organizations delete-organization
```

- API-Einheiten finden Sie [DeleteOrganization](#) unter AWS CLI Befehlsreferenz.

delete-organizational-unit

Das folgende Codebeispiel zeigt die Verwendung `delete-organizational-unit`.

AWS CLI

Um eine Organisationseinheit zu löschen

Im folgenden Beispiel wird gezeigt, wie eine Organisationseinheit gelöscht wird. Das Beispiel geht davon aus, dass Sie zuvor alle Konten und andere Konten OUs aus der Organisationseinheit entfernt haben:

```
aws organizations delete-organizational-unit --organizational-unit-id ou-examplerootid111-exampleouid111
```

- API-Einheiten finden Sie [DeleteOrganizationalUnit](#) in der AWS CLI Befehlsreferenz.

delete-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-policy`.

AWS CLI

Um eine Richtlinie zu löschen

Das folgende Beispiel zeigt, wie eine Richtlinie aus einer Organisation gelöscht wird. Das Beispiel geht davon aus, dass Sie die Richtlinie zuvor von allen Entitäten getrennt haben:

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API-Einheiten finden Sie [DeletePolicy](#) in der AWS CLI Befehlsreferenz.

describe-account

Das folgende Codebeispiel zeigt die Verwendung `describe-account`.

AWS CLI

Um die Details zu einem Konto abzurufen

Das folgende Beispiel zeigt Ihnen, wie Sie Details zu einem Konto anfordern können:


```
aws organizations describe-account --account-id 555555555555
```

Die Ausgabe zeigt ein Kontoobjekt mit den Details zum Konto:

```
{
  "Account": {
    "Id": "555555555555",
    "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/555555555555",
    "Name": "Beta account",
    "Email": "anika@example.com",
    "JoinedMethod": "INVITED",
    "JoinedTimeStamp": 1481756563.134,
    "Status": "ACTIVE"
  }
}
```

- API-Einheiten finden Sie [DescribeAccount](#) in der AWS CLI Befehlsreferenz.

describe-create-account-status

Das folgende Codebeispiel zeigt die Verwendung `describe-create-account-status`.

AWS CLI

Um den neuesten Status einer Anfrage zur Kontoerstellung abzurufen

Das folgende Beispiel zeigt, wie der aktuelle Status einer früheren Anfrage zur Kontoerstellung in einer Organisation abgefragt wird. Die angegebene `--request-id` stammt aus der Antwort auf den ursprünglichen Aufruf von `create-account`. Die Anfrage zur Kontoerstellung zeigt anhand des Statusfeldes, dass Organizations die Erstellung des Kontos erfolgreich abgeschlossen haben.

Befehl:

```
aws organizations describe-create-account-status --create-account-request-id car-
examplecreateaccountrequestid111
```

Ausgabe:

```
{
  "CreateAccountStatus": {
```

```
"State": "SUCCEEDED",
"AccountId": "555555555555",
"AccountName": "Beta account",
"RequestedTimestamp": 1470684478.687,
"CompletedTimestamp": 1470684532.472,
"Id": "car-examplecreateaccountrequestid111"
}
}
```

- API-Einheiten finden Sie [DescribeCreateAccountStatus](#) in der AWS CLI Befehlsreferenz.

describe-handshake

Das folgende Codebeispiel zeigt die Verwendung des `describe-handshake`.

AWS CLI

Um Informationen über einen Handschlag zu erhalten

Das folgende Beispiel zeigt Ihnen, wie Sie Details zu einem Handschlag anfordern können. Die Handshake-ID stammt entweder aus dem ursprünglichen Anruf an `InviteAccountToOrganization` oder aus einem Anruf an `ListHandshakesForAccount` oder: `ListHandshakesForOrganization`

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

Die Ausgabe enthält ein Handshake-Objekt, das alle Details zum angeforderten Handshake enthält:

```
{
  "Handshake": {
    "Id": "h-examplehandshakeid111",
    "State": "OPEN",
    "Resources": [
      {
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid",
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          }
        ]
      }
    ]
  }
}
```

```

        {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
        }
    ],
    },
    {
        "Type": "EMAIL",
        "Value": "anika@example.com"
    }
],
"Parties": [
    {
        "Type": "ORGANIZATION",
        "Id": "o-exampleorgid"
    },
    {
        "Type": "EMAIL",
        "Id": "anika@example.com"
    }
],
"Action": "INVITE",
"RequestedTimestamp": 1470158698.046,
"ExpirationTimestamp": 1471454698.046,
"Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
}
}

```

- API-Einheiten finden Sie unter [DescribeHandshake AWS CLI](#) Befehlsreferenz.

describe-organization

Das folgende Codebeispiel zeigt die Verwendung `describe-organization`.

AWS CLI

Um Informationen über die aktuelle Organisation zu erhalten

Das folgende Beispiel zeigt Ihnen, wie Sie Details zu einer Organisation anfordern können:

```
aws organizations describe-organization
```

Die Ausgabe umfasst ein Organisationsobjekt mit den Details zur Organisation:

```
{
  "Organization": {
    "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
    "MasterAccountEmail": "bill@example.com",
    "MasterAccountId": "111111111111",
    "Id": "o-exampleorgid",
    "FeatureSet": "ALL",
    "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
    "AvailablePolicyTypes": [
      {
        "Status": "ENABLED",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    ]
  }
}
```

- API-Einheiten finden Sie [DescribeOrganization](#) unter AWS CLI Befehlsreferenz.

describe-organizational-unit

Das folgende Codebeispiel zeigt die Verwendung `describe-organizational-unit`.

AWS CLI

Um Informationen über eine Organisationseinheit zu erhalten

Im folgenden `describe-organizational-unit` Beispiel werden Details zu einer Organisationseinheit abgefragt.

```
aws organizations describe-organizational-unit \
  --organizational-unit-id ou-examplerootid111-exampleoid111
```

Ausgabe:

```
{
  "OrganizationalUnit": {
```

```

    "Name": "Accounting Group",
    "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-
    exempleroottid111-exampleouid111",
    "Id": "ou-examplerootid111-exampleouid111"
  }
}

```

- API-Einheiten finden Sie [DescribeOrganizationalUnit](#) in der AWS CLI Befehlsreferenz.

describe-policy

Das folgende Codebeispiel zeigt die Verwendung `describe-policy`.

AWS CLI

Um Informationen über eine Richtlinie zu erhalten

Das folgende Beispiel zeigt, wie Sie Informationen zu einer Richtlinie anfordern können:

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

Die Ausgabe enthält ein Richtlinienobjekt, das Details zur Richtlinie enthält:

```

{
  "Policy": {
    "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n
    {\n    \"Effect\": \"Allow\",\n    \"Action\": \"*\",\n    \"Resource\":
    \"*\"\n  }]\n}",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::111111111111:policy/o-
      exampleorgid/service_control_policy/p-examplepolicyid111",
      "Type": "SERVICE_CONTROL_POLICY",
      "Id": "p-examplepolicyid111",
      "AwsManaged": false,
      "Name": "AllowAllS3Actions",
      "Description": "Enables admins to delegate S3 permissions"
    }
  }
}

```

- API-Einheiten finden Sie [DescribePolicy](#) in der AWS CLI Befehlsreferenz.

detach-policy

Das folgende Codebeispiel zeigt die Verwendung `detach-policy`.

AWS CLI

Um eine Richtlinie von einem Root-, OU- oder Konto zu trennen

Das folgende Beispiel zeigt, wie eine Richtlinie von einer Organisationseinheit getrennt wird:

```
aws organizations detach-policy --target-id ou-examplerootid111-exampleouid111 --  
policy-id p-examplepolicyid111
```

- API-Einheiten finden Sie [DetachPolicy](#) in der AWS CLI Befehlsreferenz.

disable-policy-type

Das folgende Codebeispiel zeigt die Verwendung `disable-policy-type`.

AWS CLI

Um einen Richtlinientyp in einem Root-Verzeichnis zu deaktivieren

Das folgende Beispiel zeigt, wie der Richtlinientyp Service Control Policy (SCP) in einem Root-Verzeichnis deaktiviert wird:

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-  
type SERVICE_CONTROL_POLICY
```

Die Ausgabe zeigt, dass das PolicyTypes Antwortelement `SERVICE _ CONTROL _` nicht mehr enthält `POLICY`:

```
{  
  "Root": {  
    "PolicyTypes": [],  
    "Name": "Root",  
    "Id": "r-examplerootid111",  
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-  
examplerootid111"  
  }  
}
```

```
}
```

- API-Einheiten finden Sie [DisablePolicyType](#) in der AWS CLI Befehlsreferenz.

enable-all-features

Das folgende Codebeispiel zeigt die Verwendung von `enable-all-features`.

AWS CLI

Um alle Funktionen in einer Organisation zu aktivieren

Dieses Beispiel zeigt, wie der Administrator alle eingeladenen Konten in der Organisation auffordert, alle aktivierten Funktionen in der Organisation zu genehmigen. AWS Organizations senden eine E-Mail an die Adresse, die für jedes Konto eines eingeladenen Mitglieds registriert ist, und bitten den Inhaber, die Änderung aller Funktionen zu genehmigen, indem er den gesendeten Handschlag akzeptiert. Nachdem alle Konten eingeladener Mitglieder den Handshake akzeptiert haben, kann der Organisationsadministrator die Änderung an allen Funktionen abschließen. Benutzer mit den entsprechenden Berechtigungen können Richtlinien erstellen und diese auf Stammkonten und Konten anwenden: OUs

```
aws organizations enable-all-features
```

Die Ausgabe ist ein Handshake-Objekt, das zur Genehmigung an alle Konten eingeladener Mitglieder gesendet wird:

```
{
  "Handshake": {
    "Action": "ENABLE_ALL_FEATURES",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/enable_all_features/h-examplehandshakeid111",
    "ExpirationTimestamp": 1.483127868609E9,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "id": "o-exampleorgid",
        "type": "ORGANIZATION"
      }
    ],
    "requestedTimestamp": 1.481831868609E9,
  }
}
```

```

        "resources": [
            {
                "type": "ORGANIZATION",
                "value": "o-exampleorgid"
            }
        ],
        "state": "REQUESTED"
    }
}

```

- API-Einheiten finden Sie unter [EnableAllFeatures AWS CLI](#) Befehlsreferenz.

enable-policy-type

Das folgende Codebeispiel zeigt die Verwendung von `enable-policy-type`.

AWS CLI

Um die Verwendung eines Richtlinien Typs in einem Stammverzeichnis zu aktivieren

Das folgende Beispiel zeigt, wie der Richtlinien Typ Service Control Policy (SCP) in einem Stammverzeichnis aktiviert wird:

```

aws organizations enable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY

```

Die Ausgabe zeigt ein Stammobjekt mit einem `policyTypes` Antwortelement, das angibt, welche jetzt aktiviert SCPs sind:

```

{
  "Root": {
    "PolicyTypes": [
      {
        "Status": "ENABLED",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    ],
    "Id": "r-examplerootid111",
    "Name": "Root",
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
  }
}

```



```
}

```

- API-Einheiten finden Sie [EnablePolicyType](#) in der AWS CLI Befehlsreferenz.

invite-account-to-organization

Das folgende Codebeispiel zeigt die Verwendung `invite-account-to-organization`.

AWS CLI

Um ein Konto einzuladen, einer Organisation beizutreten

Das folgende Beispiel zeigt, wie das Hauptkonto `bill@example.com` das Konto `juan@example.com` einlädt, einer Organisation beizutreten:

```
aws organizations invite-account-to-organization --target '{"Type": "EMAIL", "Id":
"juan@example.com"}' --notes "This is a request for Juan's account to join Bill's
organization."
```

Die Ausgabe enthält eine Handshake-Struktur, die zeigt, was an das eingeladene Konto gesendet wird:

```
{
  "Handshake": {
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1482952459.257,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "juan@example.com",
        "Type": "EMAIL"
      }
    ],
    "RequestedTimestamp": 1481656459.257,
    "Resources": [
      {
```

```

        "Resources": [
            {
                "Type": "MASTER_EMAIL",
                "Value": "bill@amazon.com"
            },
            {
                "Type": "MASTER_NAME",
                "Value": "Org Master Account"
            },
            {
                "Type": "ORGANIZATION_FEATURE_SET",
                "Value": "FULL"
            }
        ],
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid"
    },
    {
        "Type": "EMAIL",
        "Value": "juan@example.com"
    }
],
"State": "OPEN"
}
}

```

- API-Einheiten finden Sie [InviteAccountToOrganization](#) in der AWS CLI Befehlsreferenz.

leave-organization

Das folgende Codebeispiel zeigt die Verwendung `leave-organization`.

AWS CLI

Um eine Organisation als Mitgliedskonto zu verlassen

Das folgende Beispiel zeigt, wie der Administrator eines Mitgliedskontos darum bittet, die Organisation zu verlassen, der er derzeit angehört:

```
aws organizations leave-organization
```

- API-Einheiten finden Sie [LeaveOrganization](#) unter AWS CLI Befehlsreferenz.

list-accounts-for-parent

Das folgende Codebeispiel zeigt die Verwendung `list-accounts-for-parent`.

AWS CLI

Um eine Liste aller Konten in einem angegebenen übergeordneten Stammverzeichnis oder einer Organisationseinheit abzurufen

Das folgende Beispiel zeigt, wie eine Liste der Konten in einer Organisationseinheit angefordert wird:

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-exampleouid111
```

Die Ausgabe enthält eine Liste von Objekten mit einer Kontoübersicht.

```
{
  "Accounts": [
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-exampleorgid/333333333333",
      "JoinedMethod": "INVITED",
      "JoinedTimestamp": 1481835795.536,
      "Id": "333333333333",
      "Name": "Development Account",
      "Email": "juan@example.com",
      "Status": "ACTIVE"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-exampleorgid/444444444444",
      "JoinedMethod": "INVITED",
      "JoinedTimestamp": 1481835812.143,
      "Id": "444444444444",
      "Name": "Test Account",
      "Email": "anika@example.com",
      "Status": "ACTIVE"
    }
  ]
}
```

- API-Einheiten finden Sie [ListAccountsForParent](#) in der AWS CLI Befehlsreferenz.

list-accounts

Das folgende Codebeispiel zeigt die Verwendung `list-accounts`.

AWS CLI

Um eine Liste aller Konten in einer Organisation abzurufen

Das folgende Beispiel zeigt Ihnen, wie Sie eine Liste der Konten in einer Organisation anfordern können:

```
aws organizations list-accounts
```

Die Ausgabe umfasst eine Liste von Objekten mit einer Kontoübersicht.

```
{
  "Accounts": [
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
      "JoinedMethod": "INVITED",
      "JoinedTimestamp": 1481830215.45,
      "Id": "111111111111",
      "Name": "Master Account",
      "Email": "bill@example.com",
      "Status": "ACTIVE"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/222222222222",
      "JoinedMethod": "INVITED",
      "JoinedTimestamp": 1481835741.044,
      "Id": "222222222222",
      "Name": "Production Account",
      "Email": "alice@example.com",
      "Status": "ACTIVE"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
      "JoinedMethod": "INVITED",
      "JoinedTimestamp": 1481835795.536,
      "Id": "333333333333",
```

```

        "Name": "Development Account",
        "Email": "juan@example.com",
        "Status": "ACTIVE"
    },
    {
        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/4444444444444444",
        "JoinedMethod": "INVITED",
        "JoinedTimestamp": 1481835812.143,
        "Id": "4444444444444444",
        "Name": "Test Account",
        "Email": "anika@example.com",
        "Status": "ACTIVE"
    }
]
}

```

- API-Einheiten finden Sie [ListAccounts](#) in der AWS CLI Befehlsreferenz.

list-children

Das folgende Codebeispiel zeigt die Verwendung `list-children`.

AWS CLI

Um die Konten der Kinder und einer übergeordneten Organisationseinheit oder OUs eines Stammkontos abzurufen

Im folgenden Beispiel wird gezeigt, wie Sie das Stammkonto oder die Organisationseinheit auflisten, die dieses Konto 4444444444 enthält:

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-
examplerootid111-exampleoid111
```

Die Ausgabe zeigt die beiden untergeordneten Elemente, die im Elternteil OUs enthalten sind:

```

{
  "Children": [
    {
      "Id": "ou-examplerootid111-exampleoid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}

```

```
    },
    {
      "Id": "ou-examplerootid111-exampleouid222",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```

- API-Einheiten finden Sie [ListChildren](#) in der AWS CLI Befehlsreferenz.

list-create-account-status

Das folgende Codebeispiel zeigt die Verwendung `list-create-account-status`.

AWS CLI

Beispiel 1: Um eine Liste der Anfragen zur Kontoerstellung abzurufen, die in der aktuellen Organisation gestellt wurden

Das folgende Beispiel zeigt, wie Sie eine Liste von Anfragen zur Kontoerstellung für eine Organisation anfordern, die erfolgreich abgeschlossen wurden:

```
aws organizations list-create-account-status --states SUCCEEDED
```

Die Ausgabe umfasst eine Reihe von Objekten mit Informationen zu jeder Anfrage.

```
{
  "CreateAccountStatuses": [
    {
      "AccountId": "4444444444444444",
      "AccountName": "Developer Test Account",
      "CompletedTimeStamp": 1481835812.143,
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "State": "SUCCEEDED"
    }
  ]
}
```

Beispiel 2: Um eine Liste der laufenden Anfragen zur Kontoerstellung abzurufen, die in der aktuellen Organisation gestellt wurden

Im folgenden Beispiel wird eine Liste der laufenden Anfragen zur Kontoerstellung für eine Organisation abgerufen:

```
aws organizations list-create-account-status --states IN_PROGRESS
```

Die Ausgabe umfasst eine Reihe von Objekten mit Informationen zu jeder Anfrage.

```
{
  "CreateAccountStatuses": [
    {
      "State": "IN_PROGRESS",
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "AccountName": "Production Account"
    }
  ]
}
```

- API-Einheiten finden Sie [ListCreateAccountStatus](#) in der AWS CLI Befehlsreferenz.

list-handshakes-for-account

Das folgende Codebeispiel zeigt die Verwendung `list-handshakes-for-account`.

AWS CLI

Um eine Liste der Handshakes abzurufen, die an ein Konto gesendet wurden

Das folgende Beispiel zeigt, wie Sie eine Liste aller Handshakes abrufen können, die dem Konto der Anmeldeinformationen zugeordnet sind, die zum Aufrufen des Vorgangs verwendet wurden:

```
aws organizations list-handshakes-for-account
```

Die Ausgabe enthält eine Liste von Handshake-Strukturen mit Informationen zu jedem Handshake, einschließlich seines aktuellen Status:

```
{
  "Handshake": {
    "Action": "INVITE",
```

```

    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1482952459.257,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "juan@example.com",
        "Type": "EMAIL"
      }
    ],
    "RequestedTimestamp": 1481656459.257,
    "Resources": [
      {
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@amazon.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Org Master Account"
          },
          {
            "Type": "ORGANIZATION_FEATURE_SET",
            "Value": "FULL"
          }
        ],
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid"
      },
      {
        "Type": "EMAIL",
        "Value": "juan@example.com"
      }
    ],
    "State": "OPEN"
  }
}

```

- API-Einheiten finden Sie unter [ListHandshakesForAccount AWS CLI Befehlsreferenz](#).

list-handshakes-for-organization

Das folgende Codebeispiel zeigt die Verwendung `list-handshakes-for-organization`.

AWS CLI

Um eine Liste der Handshakes abzurufen, die einer Organisation zugeordnet sind

Das folgende Beispiel zeigt, wie Sie eine Liste von Handshakes abrufen können, die der aktuellen Organisation zugeordnet sind:

```
aws organizations list-handshakes-for-organization
```

Die Ausgabe zeigt zwei Handshakes. Die erste ist eine Einladung zu Juans Konto und zeigt den Status von. OPEN Die zweite ist eine Einladung zu Anikas Konto und zeigt folgenden Status anACCEPTED:

```
{
  "Handshakes": [
    {
      "Action": "INVITE",
      "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
      "ExpirationTimestamp": 1482952459.257,
      "Id": "h-examplehandshakeid111",
      "Parties": [
        {
          "Id": "o-exampleorgid",
          "Type": "ORGANIZATION"
        },
        {
          "Id": "juan@example.com",
          "Type": "EMAIL"
        }
      ],
      "RequestedTimestamp": 1481656459.257,
      "Resources": [
        {
          "Resources": [
            {
              "Type": "MASTER_EMAIL",
              "Value": "bill@amazon.com"
            }
          ],

```

```

        {
            "Type": "MASTER_NAME",
            "Value": "Org Master
Account"
        },
        {
            "Type":
"ORGANIZATION_FEATURE_SET",
            "Value": "FULL"
        }
    ],
    "Type": "ORGANIZATION",
    "Value": "o-exampleorgid"
},
{
    "Type": "EMAIL",
    "Value": "juan@example.com"
},
{
    "Type": "NOTES",
    "Value": "This is an invitation to Juan's
account to join Bill's organization."
}
],
"State": "OPEN"
},
{
    "Action": "INVITE",
    "State": "ACCEPTED",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1.471797437427E9,
    "Id": "h-examplehandshakeid222",
    "Parties": [
        {
            "Id": "o-exampleorgid",
            "Type": "ORGANIZATION"
        },
        {
            "Id": "anika@example.com",
            "Type": "EMAIL"
        }
    ]
},
"RequestedTimestamp": 1.469205437427E9,

```

```

    "Resources": [
      {
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
          }
        ],
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid"
      },
      {
        "Type": "EMAIL",
        "Value": "anika@example.com"
      },
      {
        "Type": "NOTES",
        "Value": "This is an invitation to Anika's
account to join Bill's organization."
      }
    ]
  }
]
}

```

- API-Einheiten finden Sie [ListHandshakesForOrganization](#) in der AWS CLI Befehlsreferenz.

list-organizational-units-for-parent

Das folgende Codebeispiel zeigt die Verwendung `list-organizational-units-for-parent`.

AWS CLI

Um eine Liste der OUs in einer übergeordneten Organisationseinheit oder einem Stammverzeichnis abzurufen

Das folgende Beispiel zeigt Ihnen, wie Sie eine Liste von OUs in einem angegebenen Stammverzeichnis abrufen:

```
aws organizations list-organizational-units-for-parent --parent-id r-examplerootid111
```

Die Ausgabe zeigt, dass der angegebene Stamm zwei enthält, OUs und es werden Details zu jedem Stamm angezeigt:

```
{
  "OrganizationalUnits": [
    {
      "Name": "AccountingDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-examplerootid111/ou-examplerootid111-exampleoid111"
    },
    {
      "Name": "ProductionDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-examplerootid111/ou-examplerootid111-exampleoid222"
    }
  ]
}
```

- API-Einheiten finden Sie [ListOrganizationalUnitsForParent](#) in der AWS CLI Befehlsreferenz.

list-parents

Das folgende Codebeispiel zeigt die Verwendung `list-parents`.

AWS CLI

Um das Elternteil OUs oder die Stammorganisationen für ein Konto oder eine untergeordnete Organisationseinheit aufzulisten

Im folgenden Beispiel wird gezeigt, wie Sie die Stamm- oder übergeordnete Organisationseinheit auflisten, die das Konto 444444444444 enthält:

```
aws organizations list-parents --child-id 444444444444
```

Die Ausgabe zeigt, dass sich das angegebene Konto in der Organisationseinheit mit der angegebenen ID befindet:

```
{
```

```

  "Parents": [
    {
      "Id": "ou-examplerootid111-exampleoid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}

```

- API-Einheiten finden Sie [ListParents](#) in der AWS CLI Befehlsreferenz.

list-policies-for-target

Das folgende Codebeispiel zeigt die Verwendung `list-policies-for-target`.

AWS CLI

Um eine Liste der direkt SCPs an ein Konto angehängten Dateien abzurufen

Das folgende Beispiel zeigt, wie eine Liste aller Dienststeuerungsrichtlinien (SCPs) abgerufen wird, wie im Filter-Parameter angegeben, die direkt mit einem Konto verknüpft sind:

```

aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-id 444444444444

```

Die Ausgabe umfasst eine Liste von Richtlinienstrukturen mit zusammenfassenden Informationen zu den Richtlinien. Die Liste enthält keine Richtlinien, die aufgrund der Vererbung von seinem Standort in einer OU-Hierarchie für das Konto gelten:

```

{
  "Policies": [
    {
      "Type": "SERVICE_CONTROL_POLICY",
      "Name": "AllowAllEC2Actions",
      "AwsManaged": false,
      "Id": "p-examplepolicyid222",
      "Arn": "arn:aws:organizations::o-exampleorgid:policy/service_control_policy/p-examplepolicyid222",
      "Description": "Enables account admins to delegate permissions for any EC2 actions to users and roles in their accounts."
    }
  ]
}

```

```
}
```

- API-Einheiten finden Sie [ListPoliciesForTarget](#) in der AWS CLI Befehlsreferenz.

list-policies

Das folgende Codebeispiel zeigt die Verwendung `list-policies`.

AWS CLI

Um eine Liste aller Richtlinien in einer Organisation eines bestimmten Typs abzurufen

Das folgende Beispiel zeigt Ihnen, wie Sie eine Liste von abzurufen SCPs, wie im Filterparameter angegeben:

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

Die Ausgabe enthält eine Liste von Richtlinien mit zusammenfassenden Informationen:

```
{
  "Policies": [
    {
      "Type": "SERVICE_CONTROL_POLICY",
      "Name": "AllowAllS3Actions",
      "AwsManaged": false,
      "Id": "p-examplepolicyid111",
      "Arn": "arn:aws:organizations::111111111111:policy/service_control_policy/p-examplepolicyid111",
      "Description": "Enables account admins to delegate permissions for any S3 actions to users and roles in their accounts."
    },
    {
      "Type": "SERVICE_CONTROL_POLICY",
      "Name": "AllowAllEC2Actions",
      "AwsManaged": false,
      "Id": "p-examplepolicyid222",
      "Arn": "arn:aws:organizations::111111111111:policy/service_control_policy/p-examplepolicyid222",
      "Description": "Enables account admins to delegate permissions for any EC2 actions to users and roles in their accounts."
    },
    {
```

```

        "AwsManaged": true,
        "Description": "Allows access to every operation",
        "Type": "SERVICE_CONTROL_POLICY",
        "Id": "p-FullAWSAccess",
        "Arn": "arn:aws:organizations::aws:policy/
service_control_policy/p-FullAWSAccess",
        "Name": "FullAWSAccess"
    }
]
}

```

- API-Einheiten finden Sie [ListPolicies](#) in der AWS CLI Befehlsreferenz.

list-roots

Das folgende Codebeispiel zeigt die Verwendung `list-roots`.

AWS CLI

Um eine Liste der Wurzeln in einer Organisation abzurufen

Dieses Beispiel zeigt Ihnen, wie Sie die Stammliste für eine Organisation abrufen können:

```
aws organizations list-roots
```

Die Ausgabe enthält eine Liste von Stammstrukturen mit zusammenfassenden Informationen:

```

{
  "Roots": [
    {
      "Name": "Root",
      "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
      "Id": "r-examplerootid111",
      "PolicyTypes": [
        {
          "Status": "ENABLED",
          "Type": "SERVICE_CONTROL_POLICY"
        }
      ]
    }
  ]
}

```

```
}

```

- API-Einheiten finden Sie [ListRoots](#) in der AWS CLI Befehlsreferenz.

list-targets-for-policy

Das folgende Codebeispiel zeigt die Verwendung `list-targets-for-policy`.

AWS CLI

Um eine Liste der Roots und Konten abzurufen OUs, an die eine Richtlinie angehängt ist

Das folgende Beispiel zeigt, wie Sie eine Liste der Roots und Konten abrufen OUs, denen die angegebene Richtlinie zugeordnet ist:

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

Die Ausgabe umfasst eine Liste von Anhangsobjekten mit zusammenfassenden Informationen zu den Wurzeln und Konten OUs, an die die Richtlinie angehängt ist:

```
{
  "Targets": [
    {
      "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
      "Name": "Root",
      "TargetId": "r-examplerootid111",
      "Type": "ROOT"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333;",
      "Name": "Developer Test Account",
      "TargetId": "333333333333",
      "Type": "ACCOUNT"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:ou/o-
exampleorgid/ou-examplerootid111-exampleoid111",
      "Name": "Accounting",
      "TargetId": "ou-examplerootid111-exampleoid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```



```

    ]
}
}

```

- API-Einheiten finden Sie [ListTargetsForPolicy](#) in der AWS CLI Befehlsreferenz.

move-account

Das folgende Codebeispiel zeigt die Verwendung `move-account`.

AWS CLI

Um ein Konto zwischen Roots zu verschieben oder OUs

Das folgende Beispiel zeigt Ihnen, wie Sie das Hauptkonto in der Organisation vom Stammkonto in eine Organisationseinheit verschieben:

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-
examplerootid111 --destination-parent-id ou-examplerootid111-exampleouid111
```

- API-Einheiten finden Sie [MoveAccount](#) unter AWS CLI Befehlsreferenz.

remove-account-from-organization

Das folgende Codebeispiel zeigt die Verwendung `remove-account-from-organization`.

AWS CLI

Um ein Konto als Hauptkonto aus einer Organisation zu entfernen

Das folgende Beispiel zeigt Ihnen, wie Sie ein Konto aus einer Organisation entfernen:

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- API-Einheiten finden Sie [RemoveAccountFromOrganization](#) unter AWS CLI Befehlsreferenz.

update-organizational-unit

Das folgende Codebeispiel zeigt die Verwendung `update-organizational-unit`.

AWS CLI

Um eine Organisationseinheit umzubenennen

Dieses Beispiel zeigt Ihnen, wie Sie eine Organisationseinheit umbenennen: In diesem Beispiel wird die Organisationseinheit in „AccountingOU“ umbenannt:

```
aws organizations update-organizational-unit --organizational-unit-id ou-examplerootid111-exampleoid111 --name AccountingOU
```

Die Ausgabe zeigt den neuen Namen:

```
{
  "OrganizationalUnit": {
    "Id": "ou-examplerootid111-exampleoid111"
    "Name": "AccountingOU",
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-examplerootid111-exampleoid111"
  }
}
```

- API-Einheiten finden Sie [UpdateOrganizationalUnit](#) in der AWS CLI Befehlsreferenz.

update-policy

Das folgende Codebeispiel zeigt die Verwendung `update-policy`.

AWS CLI

Beispiel 1: Um eine Richtlinie umzubenennen

Das folgende `update-policy` Beispiel benennt eine Richtlinie um und gibt ihr eine neue Beschreibung.

```
aws organizations update-policy \
  --policy-id p-examplepolicyid111 \
  --name Renamed-Policy \
  --description "This description replaces the original."
```

Die Ausgabe zeigt den neuen Namen und die neue Beschreibung.

```
{
```

```

    "Policy": {
      "Content": "{\n  \"Version\": \"2012-10-17\", \n  \"Statement\": {\n\n\n    \"Effect\": \"Allow\", \n    \"Action\": \"ec2:*\", \n    \"Resource\": \"*\"\n  }\n}\n",
      "PolicySummary": {
        "Id": "p-examplepolicyid111",
        "AwsManaged": false,
        "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/service_control_policy/p-examplepolicyid111",
        "Description": "This description replaces the original.",
        "Name": "Renamed-Policy",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    }
  }
}

```

Beispiel 2: Um den JSON Textinhalt einer Richtlinie zu ersetzen

Das folgende Beispiel zeigt Ihnen, wie Sie den JSON Text des SCP vorherigen Beispiels durch eine neue JSON Richtlinientextzeichenfolge ersetzen, die S3 anstelle von erlaubtEC2:

```

aws organizations update-policy \
  --policy-id p-examplepolicyid111 \
  --content "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\": \"Allow\", \"Action\": \"s3:*\", \"Resource\": \"*\" } }"

```

Die Ausgabe zeigt den neuen Inhalt:

```

{
  "Policy": {
    "Content": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\": \"Allow\", \"Action\": \"s3:*\", \"Resource\": \"*\" } }",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/service_control_policy/p-examplepolicyid111",
      "AwsManaged": false;
      "Description": "This description replaces the original.",
      "Id": "p-examplepolicyid111",
      "Name": "Renamed-Policy",
      "Type": "SERVICE_CONTROL_POLICY"
    }
  }
}

```

```
}
```

- API-Einheiten finden Sie [UpdatePolicy](#) in der AWS CLI Befehlsreferenz.

AWS Outposts Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Outposts.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-outpost-instance-types

Das folgende Codebeispiel zeigt die Verwendung `get-outpost-instance-types`.

AWS CLI

Um die Instance-Typen auf Ihrem Outpost abzurufen

Im folgenden `get-outpost-instance-types` Beispiel werden die Instance-Typen für den angegebenen Outpost abgerufen.

```
aws outposts get-outpost-instance-types \  
  --outpost-id op-0ab23c4567EXAMPLE
```

Ausgabe:

```
{  
  "InstanceTypes": [  
    {
```

```

        "InstanceType": "c5d.large"
      },
      {
        "InstanceType": "i3en.24xlarge"
      },
      {
        "InstanceType": "m5d.large"
      },
      {
        "InstanceType": "r5d.large"
      }
    ],
    "OutpostId": "op-0ab23c4567EXAMPLE",
    "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE"
  }
}

```

Weitere Informationen finden Sie im Outposts-Benutzerhandbuch unter [Eine Instance auf Ihrem AWS Outpost starten](#).

- API-Einheiten finden Sie [GetOutpostInstanceTypes](#) in der AWS CLI Befehlsreferenz.

get-outpost

Das folgende Codebeispiel zeigt die Verwendung `get-outpost`.

AWS CLI

Um Outpost-Details zu erhalten

Im folgenden `get-outpost` Beispiel werden die Details für den angegebenen Outpost angezeigt.

```

aws outposts get-outpost \
  --outpost-id op-0ab23c4567EXAMPLE

```

Ausgabe:

```

{
  "Outpost": {
    "OutpostId": "op-0ab23c4567EXAMPLE",
    "OwnerId": "123456789012",
    "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",

```

```
    "SiteId": "os-0ab12c3456EXAMPLE",
    "Name": "EXAMPLE",
    "LifecycleStatus": "ACTIVE",
    "AvailabilityZone": "us-west-2a",
    "AvailabilityZoneId": "usw2-az1",
    "Tags": {}
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Outposts](#) im AWS Outposts User Guide.

- API-Einheiten finden Sie [GetOutpost](#) in der AWS CLI Befehlsreferenz.

list-outposts

Das folgende Codebeispiel zeigt die Verwendung `list-outposts`.

AWS CLI

Um Outposts aufzulisten

Das folgende `list-outposts` Beispiel listet die Outposts in Ihrem AWS Konto auf.

```
aws outposts list-outposts
```

Ausgabe:

```
{
  "Outposts": [
    {
      "OutpostId": "op-0ab23c4567EXAMPLE",
      "OwnerId": "123456789012",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",
      "SiteId": "os-0ab12c3456EXAMPLE",
      "Name": "EXAMPLE",
      "Description": "example",
      "LifecycleStatus": "ACTIVE",
      "AvailabilityZone": "us-west-2a",
      "AvailabilityZoneId": "usw2-az1",
      "Tags": {
        "Name": "EXAMPLE"
      }
    }
  ]
}
```

```
    },
    {
      "OutpostId": "op-4fe3dc21baEXAMPLE",
      "OwnerId": "123456789012",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-4fe3dc21baEXAMPLE",
      "SiteId": "os-0ab12c3456EXAMPLE",
      "Name": "EXAMPLE2",
      "LifecycleStatus": "ACTIVE",
      "AvailabilityZone": "us-west-2a",
      "AvailabilityZoneId": "usw2-az1",
      "Tags": {}
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Outposts](#) im AWS Outposts User Guide.

- API-Einheiten finden Sie [ListOutposts](#) in der AWS CLI Befehlsreferenz.

list-sites

Das folgende Codebeispiel zeigt die Verwendung `list-sites`.

AWS CLI

Um Websites aufzulisten

Das folgende `list-sites` Beispiel listet die verfügbaren Outpost-Sites in Ihrem AWS Konto auf.

```
aws outposts list-sites
```

Ausgabe:

```
{
  "Sites": [
    {
      "SiteId": "os-0ab12c3456EXAMPLE",
      "AccountId": "123456789012",
      "Name": "EXAMPLE",
      "Description": "example",
      "Tags": {}
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Outposts](#) im AWS Outposts User Guide.

- API-Einheiten finden Sie [ListSites](#) in der AWS CLI Befehlsreferenz.

AWS Payment Cryptography Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Payment Cryptography.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-alias

Das folgende Codebeispiel zeigt die Verwendung `create-alias`.

AWS CLI

Um einen Alias für einen Schlüssel zu erstellen

Im folgenden `create-alias` Beispiel wird ein Alias für einen Schlüssel erstellt.

```
aws payment-cryptography create-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h
```

Ausgabe:


```
{
  "Alias": {
    "AliasName": "alias/sampleAlias1",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
  }
}
```

Weitere Informationen finden Sie unter [About Aliases](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [CreateAlias](#) in der AWS CLI Befehlsreferenz.

create-key

Das folgende Codebeispiel zeigt die Verwendung `create-key`.

AWS CLI

Um einen Schlüssel zu erstellen

Das folgende `create-key` Beispiel generiert einen KEY TDES 2-Schlüssel, den Sie verwenden können, um CVV CVV2 /-Werte zu generieren und zu überprüfen.

```
aws payment-cryptography create-key \
  --exportable \
  --key-
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY, KeyClass=SYMMETRIC
```

Ausgabe:

```
{
  "Key": {
    "CreateTimestamp": "1686800690",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_2KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": false,
```

```
        "DeriveKey": false,
        "Encrypt": false,
        "Generate": true,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": false,
        "Verify": true,
        "Wrap": false
    },
    "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
},
"KeyCheckValue": "F2E50F",
"KeyCheckValueAlgorithm": "ANSI_X9_24",
"KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
"KeyState": "CREATE_COMPLETE",
"UsageStartTimestamp": "1686800690"
}
}
```

Weitere Informationen finden Sie unter [Generieren von Schlüsseln](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [CreateKey](#) in der AWS CLI Befehlsreferenz.

delete-alias

Das folgende Codebeispiel zeigt die Verwendung `delete-alias`.

AWS CLI

Um einen Alias zu löschen

Im folgenden `delete-alias` Beispiel wird ein Alias gelöscht. Es hat keinen Einfluss auf den Schlüssel.

```
aws payment-cryptography delete-alias \  
  --alias-name alias/sampleAlias1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [About Aliases](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [DeleteAlias](#) in der AWS CLI Befehlsreferenz.

delete-key

Das folgende Codebeispiel zeigt die Verwendung `delete-key`.

AWS CLI

Um einen Schlüssel zu löschen

Im folgenden `delete-key` Beispiel wird die Löschung eines Schlüssels nach 7 Tagen geplant. Dies ist die standardmäßige Wartezeit.

```
aws payment-cryptography delete-key \  
  --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/  
  kwapwa6qaifl1w2h
```

Ausgabe:

```
{  
  "Key": {  
    "CreateTimestamp": "1686801198",  
    "DeletePendingTimestamp": "1687405998",  
    "Enabled": true,  
    "Exportable": true,  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaifl1w2h",  
    "KeyAttributes": {  
      "KeyAlgorithm": "TDES_2KEY",  
      "KeyClass": "SYMMETRIC_KEY",  
      "KeyModesOfUse": {  
        "Decrypt": false,  
        "DeriveKey": false,  
        "Encrypt": false,  
        "Generate": true,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
      },  
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"  
    },  
    "KeyCheckValue": "F2E50F",  
    "KeyCheckValueAlgorithm": "ANSI_X9_24",  
  },  
}
```

```

    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "DELETE_PENDING",
    "UsageStartTimestamp": "1686801190"
  }
}

```

Weitere Informationen finden Sie unter [Löschen von Schlüsseln](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [DeleteKey](#) in der AWS CLI Befehlsreferenz.

export-key

Das folgende Codebeispiel zeigt die Verwendung `export-key`.

AWS CLI

Um einen Schlüssel zu exportieren

Das folgende `export-key` Beispiel exportiert einen Schlüssel.

```

aws payment-cryptography export-key \
  --export-key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/
lco3w6agsk7zgu2l \
  --key-material '{"Tr34KeyBlock": { \
    "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-  

west-2:123456789012:key/ftobshq7pvioc5fx", \
    "ExportToken": "export-token-cu4lg26ofcziixny", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "WrappingKeyCertificate": file://wrapping-key-certificate.pem } }'

```

Inhalt von `wrapping-key-certificate.pem`:

```

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSU0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ01SQU1ZZS8xMXFUK2svVz1RUDJQOE1V

```

Ausgabe:

```

{
  "WrappedKey": {
    "KeyMaterial":
    "308205A106092A864886F70D010702A08205923082058E020101310D300B06096086480165030402013082031F
    "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
  }
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Schlüssel exportieren](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [ExportKey](#) in der AWS CLI Befehlsreferenz.

get-alias

Das folgende Codebeispiel zeigt die Verwendung `get-alias`.

AWS CLI

Um einen Alias zu erhalten

Das folgende `get-alias` Beispiel gibt den Schlüssel zurück, ARN der dem Alias zugeordnet ist.

```
aws payment-cryptography get-alias \  
  --alias-name alias/sampleAlias1
```

Ausgabe:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaifllw2h"  
  }  
}
```

Weitere Informationen finden Sie unter [About Aliases](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [GetAlias](#) in der AWS CLI Befehlsreferenz.

get-key

Das folgende Codebeispiel zeigt die Verwendung `get-key`.

AWS CLI

Um die Metadaten eines Schlüssels abzurufen

Das folgende `get-key` Beispiel gibt die Metadaten des Schlüssels zurück, der dem Alias zugeordnet ist. Dieser Vorgang gibt kein kryptografisches Material zurück.

```
aws payment-cryptography get-key \  
--key-identifier alias/sampleAlias1
```

Ausgabe:

```
{  
  "Key": {  
    "CreateTimestamp": "1686800690",  
    "DeletePendingTimestamp": "1687405998",  
    "Enabled": true,  
    "Exportable": true,  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaifllw2h",  
    "KeyAttributes": {  
      "KeyAlgorithm": "TDES_2KEY",  
      "KeyClass": "SYMMETRIC_KEY",  
      "KeyModesOfUse": {  
        "Decrypt": false,  
        "DeriveKey": false,  
        "Encrypt": false,  
        "Generate": true,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
      },  
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"  
    },  
    "KeyCheckValue": "F2E50F",  
    "KeyCheckValueAlgorithm": "ANSI_X9_24",  
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",  
    "KeyState": "DELETE_PENDING",  
    "UsageStartTimestamp": "1686801190"  
  }  
}
```

Weitere Informationen finden Sie unter [Get Keys](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [GetKey](#) in der AWS CLI Befehlsreferenz.

get-parameters-for-export

Das folgende Codebeispiel zeigt die Verwendung `get-parameters-for-export`.

AWS CLI

Um den Exportvorgang zu initialisieren

Das folgende `get-parameters-for-export` Beispiel generiert ein key pair, signiert den Schlüssel und gibt dann das Zertifikat und den Zertifikatsstamm zurück.

```
aws payment-cryptography get-parameters-for-export \
  --signing-key-algorithm RSA_2048 \
  --key-material-type TR34_KEY_BLOCK
```

Ausgabe:

```
{
  "ExportToken": "export-token-ep5cwyzone7oya53",
  "ParametersValidUntilTimestamp": "1687415640",
  "SigningKeyAlgorithm": "RSA_2048",
  "SigningKeyCertificate":

  "MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wZGZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "SigningKeyCertificateChain":
  "MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
```

```
VQQHEwdTZWF0dGx1MQ8wDQYDVQKQEWZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvJx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

Weitere Informationen finden Sie im AWS Payment Cryptography User Guide unter [Schlüssel exportieren](#).

- API-Einheiten finden Sie [GetParametersForExport](#) in der AWS CLI Befehlsreferenz.

get-parameters-for-import

Das folgende Codebeispiel zeigt die Verwendung `get-parameters-for-import`.

AWS CLI

Um den Importvorgang zu initialisieren

Das folgende `get-parameters-for-import` Beispiel generiert ein key pair, signiert den Schlüssel und gibt dann das Zertifikat und den Zertifikatsstamm zurück.

```
aws payment-cryptography get-parameters-for-import \
  --key-material-type TR34_KEY_BLOCK \
  --wrapping-key-algorithm RSA_2048
```

Ausgabe:

```
{
  "ImportToken": "import-token-qgmafpa7nt2kfbb",
  "ParametersValidUntilTimestamp": "1687415640",
  "WrappingKeyAlgorithm": "RSA_2048",
  "WrappingKeyCertificate":
  "MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMC
  VVMxZzA1BjBGNVBAgTALdBMRAwDgYDVQKQHEwdTZWF0dGx1MQ8wDQYDVQKQEWZBbWF6
  b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
```



```

BgmqkqkiG9w0BCQEWEG5vb251QGFtYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAgTAldBMRAdDgYD
VQHEwDZWF0dGx1MQ8wDQYDVQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQDEwLUZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGFt
YXpvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
"WrappingKeyCertificateChain":
"NIICiCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAdDgYDVQHEwDZWF0dGx1MQ8wDQYDVQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQDEwLUZXR0Q21sYWx1ZAd
BgkqhkiG9w0BCQEWEG5vb251QGFtYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAgTAldBMRAdDgYD
VQHEwDZWF0dGx1MQ8wDQYDVQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQDEwLUZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGFt
YXpvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}

```

Weitere Informationen finden Sie im AWS Payment Cryptography User Guide unter [Schlüssel importieren](#).

- API-Einheiten finden Sie [GetParametersForImport](#) in der AWS CLI Befehlsreferenz.

get-public-key-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-public-key-certificate`.

AWS CLI

Um den öffentlichen Schlüssel zurückzugeben

Das folgende `get-public-key-certificate` Beispiel gibt den öffentlichen Schlüsselteil eines `key pair` zurück.

```
aws payment-cryptography get-public-key-certificate \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Ausgabe:

```
{
  "KeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySwTC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStbNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "KeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySwTC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStbNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

Weitere Informationen finden [Sie unter Abrufen des mit einem key pair verknüpften öffentlichen Schlüssels/Zertifikats](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [GetPublicKeyCertificate](#) in AWS CLI der Befehlsreferenz.

import-key

Das folgende Codebeispiel zeigt die Verwendung `import-key`.

AWS CLI

Um einen TR-34-Schlüssel zu importieren

Das folgende `import-key` Beispiel importiert einen TR-34-Schlüssel.

```
aws payment-cryptography import-key \
  --key-material='{ "Tr34KeyBlock": {" \
    CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564njnjm", \
    "ImportToken": "import-token-5ott6ho5nts7bbc9", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "SigningKeyCertificate": file://signing-key-certificate.pem, \
    "WrappedKeyBlock": file://wrapped-key-block.pem }}}
```

Inhalt von `signing-key-certificate.pem`:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ01RYWVCK25IbE1WZU1PR1ZiNjU1Q2Jz
```

Inhalt von `wrapped-key-block.pem`:

```
3082059806092A864886F70D010702A082058930820585020101310D300B06096086480165030402013082031606
```

Ausgabe:

```
{
  "Key": {
    "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
bzmvgyx dg3skt wxd",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_2KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": false,
        "DeriveKey": false,
        "Encrypt": false,
```

```
        "Generate": true,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
    },  
    "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"  
},  
"KeyCheckValue": "D9B20E",  
"KeyCheckValueAlgorithm": "ANSI_X9_24",  
"KeyOrigin": "EXTERNAL",  
"KeyState": "CREATE_COMPLETE",  
"UsageStartTimestamp": "2023-06-09T16:56:27.621000-07:00"  
}  
}
```

Weitere Informationen finden Sie unter [Schlüssel importieren](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [ImportKey](#) in der AWS CLI Befehlsreferenz.

list-aliases

Das folgende Codebeispiel zeigt die Verwendung `list-aliases`.

AWS CLI

Um eine Liste von Aliasnamen zu erhalten

Das folgende `list-aliases` Beispiel zeigt alle Aliase in Ihrem Konto in dieser Region.

```
aws payment-cryptography list-aliases
```

Ausgabe:

```
{  
  "Aliases": [  
    {  
      "AliasName": "alias/sampleAlias1",  
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qai1lw2h"  
    }  
  ]  
}
```

```
    },
    {
      "AliasName": "alias/sampleAlias2",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaif1lw2h"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Über Aliase](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [ListAliases](#) in der AWS CLI Befehlsreferenz.

list-keys

Das folgende Codebeispiel zeigt die Verwendung `list-keys`.

AWS CLI

Um eine Liste von Schlüsseln zu erhalten

Das folgende `list-keys` Beispiel zeigt alle Schlüssel in Ihrem Konto in dieser Region.

```
aws payment-cryptography list-keys
```

Ausgabe:

```
{
  "Keys": [
    {
      "CreateTimestamp": "1666506840",
      "Enabled": false,
      "Exportable": true,
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaif1lw2h",
      "KeyAttributes": {
        "KeyAlgorithm": "TDES_3KEY",
        "KeyClass": "SYMMETRIC_KEY",
        "KeyModesOfUse": {
          "Decrypt": true,
          "DeriveKey": false,
          "Encrypt": true,

```

```

        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": true,
        "Verify": false,
        "Wrap": true
    },
    "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
},
"KeyCheckValue": "369D",
"KeyCheckValueAlgorithm": "ANSI_X9_24",
"KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
"KeyState": "CREATE_COMPLETE",
"UsageStopTimestamp": "1666938840"
}
]
}

```

Weitere Informationen finden Sie im AWS Payment Cryptography User Guide unter [Schlüssel auflisten](#).

- API-Einheiten finden Sie [ListKeys](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Liste der Tags für einen Schlüssel abzurufen

Im folgenden `list-tags-for-resource` Beispiel werden die Tags für einen Schlüssel abgerufen.

```

aws payment-cryptography list-tags-for-resource \
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlw2h

```

Ausgabe:

```
{
```

```
"Tags": [
  {
    "Key": "BIN",
    "Value": "20151120"
  },
  {
    "Key": "Project",
    "Value": "Production"
  }
]
```

Weitere Informationen finden Sie unter [Verwaltung von Schlüssel-Tags mit API Operationen](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie unter [ListTagsForResource AWS CLI Befehlsreferenz](#).

restore-key

Das folgende Codebeispiel zeigt die Verwendung `restore-key`.

AWS CLI

Um einen Schlüssel wiederherzustellen, dessen Löschung geplant ist

Im folgenden `restore-key` Beispiel wird das Löschen eines Schlüssels abgebrochen.

```
aws payment-cryptography restore-key \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Ausgabe:

```
{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyAttributes": {
      "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyAlgorithm": "TDES_3KEY",
      "KeyModesOfUse": {
```

```
        "Encrypt": false,
        "Decrypt": false,
        "Wrap": false,
        "Unwrap": false,
        "Generate": true,
        "Sign": false,
        "Verify": true,
        "DeriveKey": false,
        "NoRestrictions": false
    }
},
"KeyCheckValue": "",
"KeyCheckValueAlgorithm": "ANSI_X9_24",
"Enabled": false,
"Exportable": true,
"KeyState": "CREATE_COMPLETE",
"KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
"CreateTimestamp": "1686800690",
"UsageStopTimestamp": "1687405998"
}
}
```

Weitere Informationen finden Sie unter [Löschen von Schlüsseln](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [RestoreKey](#) in der AWS CLI Befehlsreferenz.

start-key-usage

Das folgende Codebeispiel zeigt die Verwendung `start-key-usage`.

AWS CLI

Um einen Schlüssel zu aktivieren

Das folgende `start-key-usage` Beispiel ermöglicht die Verwendung eines Schlüssels.

```
aws payment-cryptography start-key-usage \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlLw2h
```

Ausgabe:


```

{
  "Key": {
    "CreateTimestamp": "1686800690",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_3KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": true,
        "DeriveKey": false,
        "Encrypt": true,
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": true,
        "Verify": false,
        "Wrap": true
      },
      "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
    },
    "KeyCheckValue": "369D",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "CREATE_COMPLETE",
    "UsageStartTimestamp": "1686800690"
  }
}

```

Weitere Informationen finden Sie unter [Schlüssel aktivieren und deaktivieren](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie unter [StartKeyUsage AWS CLI](#) Befehlsreferenz.

stop-key-usage

Das folgende Codebeispiel zeigt die Verwendung stop-key-usage.

AWS CLI

Um einen Schlüssel zu deaktivieren

Das folgende stop-key-usage Beispiel deaktiviert einen Schlüssel.

```
aws payment-cryptography stop-key-usage \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlLw2h
```

Ausgabe:

```
{
  "Key": {
    "CreateTimestamp": "1686800690",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
alsuwfxug3pgy6xh",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_3KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": true,
        "DeriveKey": false,
        "Encrypt": true,
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": true,
        "Verify": false,
        "Wrap": true
      },
      "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
    },
    "KeyCheckValue": "369D",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "CREATE_COMPLETE",
    "UsageStartTimestamp": "1686800690"
  }
}
```

Weitere Informationen finden Sie unter [Schlüssel aktivieren und deaktivieren](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie unter [StopKeyUsage AWS CLI Befehlsreferenz](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einen Schlüssel zu taggen

Im folgenden `tag-resource` Beispiel wird ein Schlüssel markiert.

```
aws payment-cryptography tag-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlw2h \  
  --tags Key=sampleTag,Value=sampleValue
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schlüssel-Tags verwalten](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einem Schlüssel zu entfernen

Im folgenden `untag-resource` Beispiel wird ein Tag aus einem Schlüssel entfernt.

```
aws payment-cryptography untag-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlw2h \  
  --tag-keys sampleTag
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schlüssel-Tags verwalten](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-alias

Das folgende Codebeispiel zeigt die Verwendung `update-alias`.

AWS CLI

Um einen Alias zu aktualisieren

Im folgenden `update-alias` Beispiel wird der Alias einem anderen Schlüssel zugeordnet.

```
aws payment-cryptography update-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
tqv5yij6wtxx64pi
```

Ausgabe:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
tqv5yij6wtxx64pi "  
  }  
}
```

Weitere Informationen finden Sie unter [About Aliases](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [UpdateAlias](#) in der AWS CLI Befehlsreferenz.

AWS Payment Cryptography Beispiele für Datenebene mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with AWS Payment Cryptography Data Plane Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

decrypt-data

Das folgende Codebeispiel zeigt die Verwendung `decrypt-data`.

AWS CLI

Um Chiffretext zu entschlüsseln

Im folgenden `decrypt-data` Beispiel werden Chiffretextdaten mithilfe eines symmetrischen Schlüssels entschlüsselt. Für diesen Vorgang muss der Schlüssel auf und auf `KeyModesOfUse` gesetzt sein `Decrypt`. `KeyUsage` `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`

```
aws payment-cryptography-data decrypt-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifllw2h \  
  --cipher-text 33612AB9D6929C3A828EB6030082B2BD \  
  --decryption-attributes 'Symmetric={Mode=CBC}'
```

Ausgabe:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifllw2h",  
  "KeyCheckValue": "71D7AE",  
  "PlainText": "31323334313233343132333431323334"  
}
```

Weitere Informationen finden Sie unter [Daten entschlüsseln](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [DecryptData](#) in der AWS CLI Befehlsreferenz.

encrypt-data

Das folgende Codebeispiel zeigt die Verwendung `encrypt-data`.

AWS CLI

Um Daten zu verschlüsseln

Im folgenden `encrypt-data` Beispiel werden Klartextdaten mit einem symmetrischen Schlüssel verschlüsselt. Für diesen Vorgang muss der Schlüssel auf `Encrypt` und `KeyUsage` auf `KeyModesOfUse` gesetzt sein. `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`

```
aws payment-cryptography-data encrypt-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifllw2h \  
  --plain-text 31323334313233343132333431323334 \  
  --encryption-attributes 'Symmetric={Mode=CBC}'
```

Ausgabe:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifllw2h",  
  "KeyCheckValue": "71D7AE",  
  "CipherText": "33612AB9D6929C3A828EB6030082B2BD"  
}
```

Weitere Informationen finden Sie unter [Daten verschlüsseln](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [EncryptData](#) in der AWS CLI Befehlsreferenz.

`generate-card-validation-data`

Das folgende Codebeispiel zeigt die Verwendung `generate-card-validation-data`.

AWS CLI

Um eine zu generieren CVV

Das folgende `generate-card-validation-data` Beispiel generiert ein CVV/CVV2.

```
aws payment-cryptography-data generate-card-validation-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifllw2h \  
  --plain-text 31323334313233343132333431323334 \  
  --encryption-attributes 'Symmetric={Mode=CBC}'
```

```
--primary-account-number=171234567890123 \
--generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

Ausgabe:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
  "KeyCheckValue": "CADD1",
  "ValidationData": "801"
}
```

Weitere Informationen finden Sie unter [Generieren von Kartendaten](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie unter [GenerateCardValidationData AWS CLI](#) Befehlsreferenz.

generate-mac

Das folgende Codebeispiel zeigt die Verwendung `generate-mac`.

AWS CLI

Um eine zu generieren MAC

Im folgenden `generate-card-validation-data` Beispiel wird ein Hash-Based Message Authentication Code (HMAC) für die Kartendatenauthentifizierung mithilfe des Algorithmus HMAC_SHA256 und eines HMAC Verschlüsselungsschlüssels generiert. Der Schlüssel muss auf TR31_M7_HMAC_KEY und auf KeyUsage gesetzt KeyModesOfUse sein `generate`.

```
aws payment-cryptography-data generate-mac \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
  --message-
data "3b313038383439303031303733393431353d32343038323236303030373030303f33" \
  --generation-attributes Algorithm=HMAC_SHA256
```

Ausgabe:

```
{
```

```

    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
    kwapwa6qaifllw2h,
    "KeyCheckValue": "2976E7",
    "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
  }

```

Weitere Informationen finden Sie unter [Generate MAC](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [GenerateMac](#) in der AWS CLI Befehlsreferenz.

generate-pin-data

Das folgende Codebeispiel zeigt die Verwendung `generate-pin-data`.

AWS CLI

Um eine zu generieren PIN

Im folgenden `generate-card-validation-data` Beispiel wird PIN mithilfe des PIN Visa-Schemas ein neues Zufallsprinzip generiert.

```

aws payment-cryptography-data generate-pin-data \
  --generation-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2tsl45p5zjbh2 \
  --encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
  --primary-account-number 171234567890123 \
  --pin-block-format ISO_FORMAT_0 \
  --generation-attributes VisaPin={PinVerificationKeyIndex=1}

```

Ausgabe:

```

{
  "GenerationKeyArn": "arn:aws:payment-cryptography:us-
  east-2:111122223333:key/37y2tsl45p5zjbh2",
  "GenerationKeyCheckValue": "7F2363",
  "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
  ivi5ksfsuplneuyt",
  "EncryptionKeyCheckValue": "7CC9E2",
  "EncryptedPinBlock": "AC17DC148BDA645E",
  "PinData": {

```



```

    "VerificationValue": "5507"
  }
}

```

Weitere Informationen finden Sie unter [Generieren von PIN Daten](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie unter [GeneratePinData AWS CLI Befehlsreferenz](#).

re-encrypt-data

Das folgende Codebeispiel zeigt die Verwendung `re-encrypt-data`.

AWS CLI

Um Daten mit einem anderen Schlüssel erneut zu verschlüsseln

Im folgenden `re-encrypt-data` Beispiel wird Chiffriertext, der mit einem AES symmetrischen Schlüssel verschlüsselt wurde, entschlüsselt und mit einem Derived Unique Key Per Transaction (DUKPT) -Schlüssel erneut verschlüsselt.

```

aws payment-cryptography-data re-encrypt-data \
  --incoming-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/hyv7ymboitd4vfy \
  --outgoing-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/jl6ythkcvzesbxen \
  --cipher-
text 4D2B0BDBA192D5AEFEAA5B3EC28E4A65383C313FFA25140101560F75FE1B99F27192A90980AB9334
 \
  --incoming-encryption-
attributes "Dukpt={Mode=ECB,KeySerialNumber=012345678911111}" \
  --outgoing-encryption-attributes '{"Symmetric": {"Mode": "ECB"}}'

```

Ausgabe:

```

{
  "CipherText":
"F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
  "KeyCheckValue": "2E8CD9"
}

```

```
}

```

Weitere Informationen finden Sie unter Daten [verschlüsseln und entschlüsseln](#) im Payment Cryptography User Guide.AWS

- APIEinzelheiten finden Sie unter Befehlsreferenz [ReEncryptData](#).AWS CLI

translate-pin-data

Das folgende Codebeispiel zeigt die Verwendung translate-pin-data.

AWS CLI

Um PIN Daten zu übersetzen

Im folgenden translate-pin-data Beispiel wird mithilfe des DUKPT Algorithmus eine PIN PEK TDES Von-Verschlüsselung mit einem ISO PIN 0-Block in eine AES ISO PIN 4-Block-Verschlüsselung übersetzt.

```
aws payment-cryptography-data translate-pin-data \
  --encrypted-pin-block "AC17DC148BDA645E" \
  --incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}' \
  --incoming-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
  --outgoing-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/4pmyquwjs3yj4vwe \
  --outgoing-translation-attributes
IsoFormat4='{PrimaryAccountNumber=171234567890123}' \
  --outgoing-dukpt-attributes KeySerialNumber="FFFF9876543210E00008"
```

Ausgabe:

```
{
  "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt
  "KeyCheckValue": "7CC9E2"
}
```

Weitere Informationen finden Sie unter [PINDaten Translate](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [TranslatePinData](#) in der AWS CLI Befehlsreferenz.

verify-auth-request-cryptogram

Das folgende Codebeispiel zeigt die Verwendung `verify-auth-request-cryptogram`.

AWS CLI

Um eine Authentifizierungsanfrage zu überprüfen

Im folgenden `verify-auth-request-cryptogram` Beispiel wird ein Autorisierungsanforderungs-Kryptogramm () verifiziert. ARQC

```
aws payment-cryptography-data verify-auth-request-cryptogram \
  --auth-request-cryptogram F6E1BD1E6037FB3E \
  --auth-response-attributes '{"ArpcMethod1": {"AuthResponseCode": "1111"}}' \
  --key-identifizier arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdk1ya \
  --major-key-derivation-mode "EMV_OPTION_A" \
  --session-key-derivation-attributes '{"EmvCommon":
{"ApplicationTransactionCounter": "1234", "PanSequenceNumber":
"01", "PrimaryAccountNumber": "471234567890123"}}' \
  --transaction-data "123456789ABCDEF"
```

Ausgabe:

```
{
  "AuthResponseValue": "D899B8C6FBF971AA",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdk1ya",
  "KeyCheckValue": "985792"
}
```

Weitere Informationen finden Sie unter [Verify Auth Request \(ARQC\) -Kryptogramm im AWS Payment Cryptography User Guide](#).

- API-Einheiten finden Sie unter Befehlsreferenz [VerifyAuthRequestCryptogram](#).AWS CLI

verify-card-validation-data

Das folgende Codebeispiel zeigt die Verwendung `verify-card-validation-data`.

AWS CLI

Um eine zu validieren CVV

Im folgenden `verify-card-validation-data` Beispiel wird ein CVV/CVV2 für ein PAN validiert.

```
aws payment-cryptography-data verify-card-validation-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/  
tqv5yij6wtxx64pi \  
  --primary-account-number=171234567890123 \  
  --verification-attributes CardVerificationValue2={CardExpiryDate=0123} \  
  --validation-data 801
```

Ausgabe:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
tqv5yij6wtxx64pi",  
  "KeyCheckValue": "CADD1"  
}
```

Weitere Informationen finden Sie unter [Kartendaten verifizieren](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [VerifyCardValidationData](#) in der AWS CLI Befehlsreferenz.

verify-mac

Das folgende Codebeispiel zeigt die Verwendung `verify-mac`.

AWS CLI

Um ein zu verifizieren MAC

Im folgenden `verify-mac` Beispiel wird ein Hash-Based Message Authentication Code (HMAC) für die Kartendatenauthentifizierung mithilfe des Algorithmus HMAC_SHA256 und eines HMAC Verschlüsselungsschlüssels verifiziert.

```
aws payment-cryptography-data verify-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qnob15lghrzunce6 \  
  --validation-data 801
```

```

--message-
data "3b343038383439303031303733393431353d32343038323236303030373030303f33" \
--verification-attributes='Algorithm=HMAC_SHA256' \
--mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C

```

Ausgabe:

```

{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qno15lghrzunce6,
  "KeyCheckValue": "2976E7",
}

```

Weitere Informationen finden Sie unter [Verify MAC](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [VerifyMac](#) in der AWS CLI Befehlsreferenz.

verify-pin-data

Das folgende Codebeispiel zeigt die Verwendung `verify-pin-data`.

AWS CLI

Um ein zu verifizieren PIN

Das folgende `verify-pin-data` Beispiel validiert a PIN für a. PAN

```

aws payment-cryptography-data verify-pin-data \
--verification-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/37y2ts145p5zjbh2 \
--encryption-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/ivi5ksfsuplneuyt \
--primary-account-number 171234567890123 \
--pin-block-format ISO_FORMAT_0 \
--verification-attributes
  VisaPin="{PinVerificationKeyIndex=1,VerificationValue=5507}" \
--encrypted-pin-block AC17DC148BDA645E

```

Ausgabe:

```
{
```

```
"VerificationKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2",
"VerificationKeyCheckValue": "7F2363",
"EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt",
"EncryptionKeyCheckValue": "7CC9E2",
}
```

Weitere Informationen finden Sie unter [PINDaten verifizieren](#) im AWS Payment Cryptography User Guide.

- API-Einheiten finden Sie [VerifyPinData](#) in der AWS CLI Befehlsreferenz.

Amazon Pinpoint Pinpoint-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Pinpoint Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-app

Das folgende Codebeispiel zeigt die Verwendung `create-app`.

AWS CLI

Beispiel 1: Erstellen einer Anwendung

Im folgenden `create-app`-Beispiel wird eine neue Anwendung (Projekt) erstellt.

```
aws pinpoint create-app \
```

```
--create-application-request Name=ExampleCorp
```

Ausgabe:

```
{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {}
  }
}
```

Beispiel 2: Erstellen einer mit Tags versehenen Anwendung

Im folgenden `create-app`-Beispiel wird eine neue Anwendung (Projekt) erstellt und der Anwendung ein Tag (Schlüssel und Wert) zugeordnet.

```
aws pinpoint create-app \
--create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

Ausgabe:

```
{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Stack": "Test"
    }
  }
}
```

- API-Einheiten finden Sie [CreateApp](#) in der AWS CLI Befehlsreferenz.

create-sms-template

Das folgende Codebeispiel zeigt, wie man es benutzt `create-sms-template`.

AWS CLI

Erstellt eine Nachrichtenvorlage für Nachrichten, die über den SMS Kanal gesendet werden

Im folgenden `create-sms-template` Beispiel wird eine SMS Nachrichtenvorlage erstellt.

```
aws pinpoint create-sms-template \  
  --template-name TestTemplate \  
  --sms-template-request file://myfile.json \  
  --region us-east-1
```

Inhalt von `myfile.json`:

```
{  
  "Body": "hello\n how are you?\n food is good",  
  "TemplateDescription": "Test SMS Template"  
}
```

Ausgabe:

```
{  
  "CreateTemplateMessageBody": {  
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/  
TestTemplate/SMS",  
    "Message": "Created",  
    "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"  
  }  
}
```

Weitere Informationen finden Sie unter [Amazon Pinpoint Pinpoint-Nachrichtenvorlagen](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CreateSmsTemplate](#) in der AWS CLI Befehlsreferenz.

delete-app

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-app`.

AWS CLI

So löschen Sie eine Anwendung

Im folgenden `delete-app`-Beispiel wird eine Anwendung (Projekt) gelöscht.

```
aws pinpoint delete-app \  
  --application-id 810c7aab86d42fb2b56c8c966example
```

Ausgabe:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {}  
  }  
}
```

- API-Einheiten finden Sie [DeleteApp](#) in der AWS CLI Befehlsreferenz.

get-apns-channel

Das folgende Codebeispiel zeigt, wie man es benutzt `get-apns-channel`.

AWS CLI

Um Informationen über den Status und die Einstellungen des APNs Kanals für eine Anwendung abzurufen

Im folgenden `get-apns-channel` Beispiel werden Informationen über den Status und die Einstellungen des APNs Kanals für eine Anwendung abgerufen.

```
aws pinpoint get-apns-channel \  
  --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \  
  --region us-east-1
```

Ausgabe:

```
{  
  "APNSChannelResponse": {  
    "ApplicationId": "9ab1068eb0a6461c86cce7f27ce0efd7",  
    "CreationDate": "2019-05-09T21:54:45.082Z",
```

```

    "DefaultAuthenticationMethod": "CERTIFICATE",
    "Enabled": true,
    "HasCredential": true,
    "HasTokenKey": false,
    "Id": "apns",
    "IsArchived": false,
    "LastModifiedDate": "2019-05-09T22:04:01.067Z",
    "Platform": "APNS",
    "Version": 2
  }
}

```

- API-Einheiten finden Sie unter [GetApnsChannel AWS CLI](#) Befehlsreferenz.

get-app

Das folgende Codebeispiel zeigt, wie man es benutzt `get-app`.

AWS CLI

Um Informationen über eine Anwendung (Projekt) abzurufen

Im folgenden `get-app` Beispiel werden Informationen über eine Anwendung (Projekt) abgerufen.

```

aws pinpoint get-app \
  --application-id 810c7aab86d42fb2b56c8c966example \
  --region us-east-1

```

Ausgabe:

```

{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Year": "2019",
      "Stack": "Production"
    }
  }
}

```

- API-Einheiten finden Sie unter [GetApp AWS CLI-Befehlsreferenz](#).

get-apps

Das folgende Codebeispiel zeigt, wie man es benutzt `get-apps`.

AWS CLI

Um Informationen über all Ihre Anwendungen abzurufen

Im folgenden `get-apps` Beispiel werden Informationen zu all Ihren Anwendungen (Projekten) abgerufen.

```
aws pinpoint get-apps
```

Ausgabe:

```
{
  "ApplicationsResponse": {
    "Item": [
      {
        "Arn": "arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
          "Year": "2019",
          "Stack": "Production"
        }
      },
      {
        "Arn": "arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/42d8c7eb0990a57ba1d5476a3example",
        "Id": "42d8c7eb0990a57ba1d5476a3example",
        "Name": "AnyCompany",
        "tags": {}
      },
      {
        "Arn": "arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/80f5c382b638ffe5ad12376bbexample",
        "Id": "80f5c382b638ffe5ad12376bbexample",
        "Name": "ExampleCorp_Test",
        "tags": {
```

```

        "Year": "2019",
        "Stack": "Test"
      }
    ],
    "NextToken":
    "eyJJdcmVhdG1vbkrhdGUiOiIyMDE5LTA3LTE2VDE0jM40jUzLjkwM1oiLCJBY2NvdW50SWQiOiI1MTIzOTcxODM4Nz"
  }
}

```

Das Vorhandensein des `NextToken` Antwortwerts weist darauf hin, dass mehr Ausgabe verfügbar ist. Rufen Sie den Befehl erneut auf und geben Sie diesen Wert als `NextToken` Eingabeparameter an.

- API-Einheiten finden Sie [GetApps](#) in der AWS CLI Befehlsreferenz.

get-campaign

Das folgende Codebeispiel zeigt, wie man es benutzt `get-campaign`.

AWS CLI

Um Informationen über den Status, die Konfiguration und andere Einstellungen einer Kampagne abzurufen

Im folgenden `get-campaign` Beispiel werden Informationen über den Status, die Konfiguration und andere Einstellungen für eine Kampagne abgerufen.

```

aws pinpoint get-campaign \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \
  --region us-east-1

```

Ausgabe:

```

{
  "CampaignResponse": {
    "AdditionalTreatments": [],
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/a1e63c6cc0eb43ed826ffcc3cc90b30d",

```

```

    "CreationDate": "2019-10-08T18:40:16.581Z",
    "Description": " ",
    "HoldoutPercent": 0,
    "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",
    "IsPaused": false,
    "LastModifiedDate": "2019-10-08T18:40:16.581Z",
    "Limits": {
      "Daily": 0,
      "MaximumDuration": 60,
      "MessagesPerSecond": 50,
      "Total": 0
    },
    "MessageConfiguration": {
      "EmailMessage": {
        "FromAddress": "sender@example.com",
        "HtmlBody": "<!DOCTYPE html>\n <html lang=\"en\">\n <head>\n
        <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />\n</head>
        \n<body>Hello</body>\n</html>",
        "Title": "PinpointDemo"
      }
    },
    "Name": "MyCampaign",
    "Schedule": {
      "IsLocalTime": false,
      "StartTime": "IMMEDIATE",
      "Timezone": "utc"
    },
    "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
    "SegmentVersion": 1,
    "State": {
      "CampaignStatus": "COMPLETED"
    },
    "tags": {},
    "TemplateConfiguration": {},
    "Version": 1
  }
}

```

- API-Einheiten finden Sie unter [GetCampaign AWS CLI-Befehlsreferenz](#).

get-campaigns

Das folgende Codebeispiel zeigt, wie man es benutzt `get-campaigns`.

AWS CLI

Ruft Informationen über den Status, die Konfiguration und andere Einstellungen für alle Kampagnen ab, die einer Anwendung zugeordnet sind

Im folgenden `get-campaigns` Beispiel werden Informationen über den Status, die Konfiguration und andere Einstellungen für alle Kampagnen abgerufen, die einer Anwendung zugeordnet sind.

```
aws pinpoint get-campaigns \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

Ausgabe:

```
{
  "CampaignsResponse": {
    "Item": [
      {
        "AdditionalTreatments": [],
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/7e1280344c8f4a9aa40a00b006fe44f1",
        "CreationDate": "2019-10-08T18:40:22.905Z",
        "Description": " ",
        "HoldoutPercent": 0,
        "Id": "7e1280344c8f4a9aa40a00b006fe44f1",
        "IsPaused": false,
        "LastModifiedDate": "2019-10-08T18:40:22.905Z",
        "Limits": {},
        "MessageConfiguration": {
          "EmailMessage": {
            "FromAddress": "sender@example.com",
            "HtmlBody": "<!DOCTYPE html>\n  <html lang=\"en\n\n>\n  <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />\n  </head>\n  <body>Hello</body>\n</html>",
            "Title": "PinpointDemo Test"
          }
        },
        "Name": "MyCampaign1",
        "Schedule": {
          "IsLocalTime": false,
          "QuietTime": {}
        }
      }
    ]
  }
}
```

```

        "StartTime": "IMMEDIATE",
        "Timezone": "UTC"
    },
    "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
    "SegmentVersion": 1,
    "State": {
        "CampaignStatus": "COMPLETED"
    },
    "tags": {},
    "TemplateConfiguration": {},
    "Version": 1
},
{
    "AdditionalTreatments": [],
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/
ale63c6cc0eb43ed826ffcc3cc90b30d",
    "CreationDate": "2019-10-08T18:40:16.581Z",
    "Description": " ",
    "HoldoutPercent": 0,
    "Id": "ale63c6cc0eb43ed826ffcc3cc90b30d",
    "IsPaused": false,
    "LastModifiedDate": "2019-10-08T18:40:16.581Z",
    "Limits": {
        "Daily": 0,
        "MaximumDuration": 60,
        "MessagesPerSecond": 50,
        "Total": 0
    },
    "MessageConfiguration": {
        "EmailMessage": {
            "FromAddress": "sender@example.com",
            "HtmlBody": "<!DOCTYPE html>\n    <html lang=\"en
\n    <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html;
charset=utf-8\" />\n</head>\n<body>Demo</body>\n</html>",
            "Title": "PinpointDemo"
        }
    },
    "Name": "MyCampaign2",
    "Schedule": {
        "IsLocalTime": false,
        "StartTime": "IMMEDIATE",
        "Timezone": "utc"
    }
}

```

```

    },
    "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
    "SegmentVersion": 1,
    "State": {
      "CampaignStatus": "COMPLETED"
    },
    "tags": {},
    "TemplateConfiguration": {},
    "Version": 1
  }
]
}

```

- API-Einheiten finden Sie unter [GetCampaigns AWS CLI-Befehlsreferenz](#).

get-channels

Das folgende Codebeispiel zeigt, wie man es benutzt `get-channels`.

AWS CLI

Ruft Informationen über den Verlauf und den Status jedes Kanals für eine Anwendung ab

Im folgenden `get-channels` Beispiel werden Informationen über den Verlauf und den Status jedes Kanals für eine Anwendung abgerufen.

```

aws pinpoint get-channels \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1

```

Ausgabe:

```

{
  "ChannelsResponse": {
    "Channels": {
      "GCM": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:28:23.182Z",
        "Enabled": true,
        "HasCredential": true,

```



```
    "Id": "gcm",
    "IsArchived": false,
    "LastModifiedDate": "2019-10-08T18:28:23.182Z",
    "Version": 1
  },
  "SMS": {
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "CreationDate": "2019-10-08T18:39:18.511Z",
    "Enabled": true,
    "Id": "sms",
    "IsArchived": false,
    "LastModifiedDate": "2019-10-08T18:39:18.511Z",
    "Version": 1
  },
  "EMAIL": {
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "CreationDate": "2019-10-08T18:27:23.990Z",
    "Enabled": true,
    "Id": "email",
    "IsArchived": false,
    "LastModifiedDate": "2019-10-08T18:27:23.990Z",
    "Version": 1
  },
  "IN_APP": {
    "Enabled": true,
    "IsArchived": false,
    "Version": 0
  }
}
}
```

- API-Einheiten finden Sie unter [GetChannels AWS CLI](#) Befehlsreferenz.

get-email-channel

Das folgende Codebeispiel zeigt, wie man es benutzt `get-email-channel`.

AWS CLI

Um Informationen über den Status und die Einstellungen des E-Mail-Kanals für eine Anwendung abzurufen

Im folgenden `get-email-channel` Beispiel werden Status und Einstellungen des E-Mail-Kanals für eine Anwendung abgerufen.

```
aws pinpoint get-email-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Ausgabe:

```
{  
  "EmailChannelResponse": {  
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
    "CreationDate": "2019-10-08T18:27:23.990Z",  
    "Enabled": true,  
    "FromAddress": "sender@example.com",  
    "Id": "email",  
    "Identity": "arn:aws:ses:us-east-1:AIDACKCEVSQ6C2EXAMPLE:identity/  
sender@example.com",  
    "IsArchived": false,  
    "LastModifiedDate": "2019-10-08T18:27:23.990Z",  
    "MessagesPerSecond": 1,  
    "Platform": "EMAIL",  
    "RoleArn": "arn:aws:iam::AIDACKCEVSQ6C2EXAMPLE:role/pinpoint-events",  
    "Version": 1  
  }  
}
```

- API-Einheiten finden Sie unter [GetEmailChannel AWS CLI](#) Befehlsreferenz.

get-endpoint

Das folgende Codebeispiel zeigt, wie man es benutzt `get-endpoint`.

AWS CLI

So rufen Sie Informationen über die Einstellungen und Attribute eines bestimmten Endpunkts für eine Anwendung ab

Das folgende `get-endpoint`-Beispiel ruft Informationen über die Einstellungen und Attribute eines bestimmten Endpunkts für eine Anwendung ab.

```
aws pinpoint get-endpoint \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

```
--application-id 611e3e3cdd47474c9c1399a505665b91 \  
--endpoint-id testendpoint \  
--region us-east-1
```

Ausgabe:

```
{  
  "EndpointResponse": {  
    "Address": "+11234567890",  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "Attributes": {},  
    "ChannelType": "SMS",  
    "CohortId": "63",  
    "CreationDate": "2019-01-28T23:55:11.534Z",  
    "EffectiveDate": "2021-08-06T00:04:51.763Z",  
    "EndpointStatus": "ACTIVE",  
    "Id": "testendpoint",  
    "Location": {  
      "Country": "USA"  
    },  
    "Metrics": {  
      "SmsDelivered": 1.0  
    },  
    "OptOut": "ALL",  
    "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",  
    "User": {  
      "UserAttributes": {  
        "Age": [  
          "24"  
        ]  
      },  
      "UserId": "testuser"  
    }  
  }  
}
```

- API-Einheiten finden Sie [GetEndpoint](#) in der AWS CLI Befehlsreferenz.

get-gcm-channel

Das folgende Codebeispiel zeigt, wie man es benutzt `get-gcm-channel`.

AWS CLI

Um Informationen über den Status und die Einstellungen des GCM Kanals für eine Anwendung abzurufen

Im folgenden `get-gcm-channel` Beispiel werden Informationen über den Status und die Einstellungen des GCM Kanals für eine Anwendung abgerufen.

```
aws pinpoint get-gcm-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Ausgabe:

```
{  
  "GCMChannelResponse": {  
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
    "CreationDate": "2019-10-08T18:28:23.182Z",  
    "Enabled": true,  
    "HasCredential": true,  
    "Id": "gcm",  
    "IsArchived": false,  
    "LastModifiedDate": "2019-10-08T18:28:23.182Z",  
    "Platform": "GCM",  
    "Version": 1  
  }  
}
```

- API-Einheiten finden Sie unter [GetGcmChannel AWS CLIBefehlsreferenz](#).

get-sms-channel

Das folgende Codebeispiel zeigt, wie man es benutzt `get-sms-channel`.

AWS CLI

Um Informationen über den Status und die Einstellungen des SMS Kanals für eine Anwendung abzurufen

Im folgenden `get-sms-channel`-Beispiel werden Status und Einstellungen des SMS-Kanals für eine Anwendung abgerufen.

```
aws pinpoint get-sms-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Ausgabe:

```
{  
  "SMSChannelResponse": {  
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
    "CreationDate": "2019-10-08T18:39:18.511Z",  
    "Enabled": true,  
    "Id": "sms",  
    "IsArchived": false,  
    "LastModifiedDate": "2019-10-08T18:39:18.511Z",  
    "Platform": "SMS",  
    "PromotionalMessagesPerSecond": 20,  
    "TransactionalMessagesPerSecond": 20,  
    "Version": 1  
  }  
}
```

- API-Einheiten finden Sie [GetSmsChannel](#) in der AWS CLI Befehlsreferenz.

get-sms-template

Das folgende Codebeispiel zeigt, wie man es benutzt `get-sms-template`.

AWS CLI

Ruft den Inhalt und die Einstellungen einer Nachrichtenvorlage für Nachrichten ab, die über den SMS-Kanal gesendet werden.

Im folgenden `get-sms-template` Beispiel werden der Inhalt und die Einstellungen einer SMS-Nachrichtenvorlage abgerufen.

```
aws pinpoint get-sms-template \  
  --template-name TestTemplate \  
  --region us-east-1
```

Ausgabe:

```
{
  "SMSTemplateResponse": {
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
    "Body": "hello\n how are you?\n food is good",
    "CreationDate": "2023-06-20T21:37:30.124Z",
    "LastModifiedDate": "2023-06-20T21:37:30.124Z",
    "tags": {},
    "TemplateDescription": "Test SMS Template",
    "TemplateName": "TestTemplate",
    "TemplateType": "SMS",
    "Version": "1"
  }
}
```

Weitere Informationen finden Sie unter [Amazon Pinpoint Pinpoint-Nachrichtenvorlagen](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSmsTemplate](#) in der AWS CLI Befehlsreferenz.

get-voice-channel

Das folgende Codebeispiel zeigt, wie man es benutzt `get-voice-channel`.

AWS CLI

Um Informationen über den Status und die Einstellungen des Sprachkanals für eine Anwendung abzurufen

Im folgenden `get-voice-channel` Beispiel werden Status und Einstellungen des Sprachkanals für eine Anwendung abgerufen.

```
aws pinpoint get-voice-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

Ausgabe:

```
{
  "VoiceChannelResponse": {
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "CreationDate": "2022-04-28T00:17:03.836Z",
```

```
    "Enabled": true,  
    "Id": "voice",  
    "IsArchived": false,  
    "LastModifiedDate": "2022-04-28T00:17:03.836Z",  
    "Platform": "VOICE",  
    "Version": 1  
  }  
}
```

- API-Einheiten finden Sie unter [GetVoiceChannel AWS CLI](#) Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um eine Liste von Tags für eine Ressource abzurufen

Im folgenden `list-tags-for-resource` Beispiel werden alle Tags (Schlüsselnamen und Werte) abgerufen, die der angegebenen Ressource zugeordnet sind.

```
aws pinpoint list-tags-for-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

Ausgabe:

```
{  
  "TagsModel": {  
    "tags": {  
      "Year": "2019",  
      "Stack": "Production"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter 'Tagging Amazon Pinpoint Resources < <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>>' im Amazon Pinpoint Developer Guide.

- API-Einheiten [ListTagsForResource AWS CLI](#) finden Sie in der Befehlsreferenz.

phone-number-validate

Das folgende Codebeispiel zeigt, wie man es benutzt `phone-number-validate`.

AWS CLI

Ruft Informationen über eine Telefonnummer ab

Im Folgenden werden Informationen zu einer Telefonnummer `phone-number-validate` abgerufen.

```
aws pinpoint phone-number-validate \  
  --number-validate-request PhoneNumber="+12065550142" \  
  --region us-east-1
```

Ausgabe:

```
{  
  "NumberValidateResponse": {  
    "Carrier": "ExampleCorp Mobile",  
    "City": "Seattle",  
    "CleansedPhoneNumberE164": "+12065550142",  
    "CleansedPhoneNumberNational": "2065550142",  
    "Country": "United States",  
    "CountryCodeIso2": "US",  
    "CountryCodeNumeric": "1",  
    "OriginalPhoneNumber": "+12065550142",  
    "PhoneType": "MOBILE",  
    "PhoneTypeCode": 0,  
    "Timezone": "America/Los_Angeles",  
    "ZipCode": "98101"  
  }  
}
```

Weitere Informationen finden Sie unter [Amazon Pinpoint SMS Pinpoint-Kanal](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

- API-Einheiten finden Sie [PhoneNumberValidate](#) in der AWS CLI Befehlsreferenz.

send-messages

Das folgende Codebeispiel zeigt, wie man es benutzt `send-messages`.

AWS CLI

Um eine SMS Nachricht über den Endpunkt einer Anwendung zu senden

Im folgenden `send-messages`-Beispiel wird eine Direktnachricht für eine Anwendung mit einem Endpunkt gesendet.

```
aws pinpoint send-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --message-request file://myfile.json \  
  --region us-west-2
```

Inhalt von `myfile.json`:

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Endpoints": {  
    "testendpoint": {}  
  }  
}
```

Ausgabe:

```
{  
  "MessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "EndpointResult": {  
      "testendpoint": {  
        "Address": "+12345678900",  
        "DeliveryStatus": "SUCCESSFUL",  
        "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",  
        "StatusCode": 200,  
        "StatusMessage": "MessageId:  
itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"  
      }  
    },  
    "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Amazon Pinpoint SMS Pinpoint-Kanal](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

- API-Einheiten finden Sie [SendMessages](#) in der AWS CLI Befehlsreferenz.

send-users-messages

Das folgende Codebeispiel zeigt, wie man es benutzt `send-users-messages`.

AWS CLI

Um eine SMS-Nachricht für einen Benutzer einer Anwendung zu senden

Im folgenden `send-users-messages` Beispiel wird eine Direktnachricht an einen Benutzer einer Anwendung gesendet.

```
aws pinpoint send-users-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --send-users-message-request file://myfile.json \  
  --region us-west-2
```

Inhalt von `myfile.json`:

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Users": {  
    "testuser": {}  
  }  
}
```

Ausgabe:

```
{  
  "SendUsersMessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",
```

```

    "Result": {
      "testuser": {
        "testuserendpoint": {
          "DeliveryStatus": "SUCCESSFUL",
          "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",
          "StatusCode": 200,
          "StatusMessage": "MessageId:
7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",
          "Address": "+12345678900"
        }
      }
    }
  }
}

```

Weitere Informationen finden Sie unter [Amazon Pinpoint SMS Pinpoint-Kanal](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

- API-Einheiten finden Sie [SendUsersMessages](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um einer Ressource Tags hinzuzufügen

Im folgenden Beispiel werden einer Ressource zwei Tags (Schlüsselnamen und Werte) hinzugefügt.

```

aws pinpoint list-tags-for-resource \
  --resource-arn arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
  --tags-model tags={Stack=Production,Year=2019}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter 'Tagging Amazon Pinpoint Resources < <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>>' im Amazon Pinpoint Developer Guide.

- API-Einheiten [TagResource AWS CLI](#) finden Sie in der Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Beispiel 1: Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das angegebene Tag (Schlüsselname und Wert) aus einer Ressource entfernt.

```
aws pinpoint untag-resource \  
  --resource-arn arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tag-keys Year
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um mehrere Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden die angegebenen Tags (Schlüsselnamen und Werte) aus einer Ressource entfernt.

```
aws pinpoint untag-resource \  
  --resource-arn arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tag-keys Year Stack
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter 'Tagging Amazon Pinpoint Resources < <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>>' __ im Amazon Pinpoint Developer Guide.

- API-Einheiten [UntagResource AWS CLI](#) finden Sie in der Befehlsreferenz.

update-sms-channel

Das folgende Codebeispiel zeigt, wie man es benutzt `update-sms-channel`.

AWS CLI

Um den SMS Kanal zu aktivieren oder den Status und die Einstellungen des SMS Kanals für eine Anwendung zu aktualisieren.

Das folgende `update-sms-channel` Beispiel aktiviert den SMS Kanal für einen SMS Kanal für eine Anwendung.

```
aws pinpoint update-sms-channel \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --sms-channel-request Enabled=true \  
  --region us-west-2
```

Ausgabe:

```
{  
  "SMSChannelResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "CreationDate": "2019-01-28T23:25:25.224Z",  
    "Enabled": true,  
    "Id": "sms",  
    "IsArchived": false,  
    "LastModifiedDate": "2023-05-18T23:22:50.977Z",  
    "Platform": "SMS",  
    "PromotionalMessagesPerSecond": 20,  
    "TransactionalMessagesPerSecond": 20,  
    "Version": 3  
  }  
}
```

Weitere Informationen finden Sie unter [Amazon Pinpoint SMS Pinpoint-Kanal](#) im Amazon Pinpoint Pinpoint-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSmsChannel](#) in der AWS CLI Befehlsreferenz.

Beispiele für Amazon Polly mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Polly Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-lexicon

Das folgende Codebeispiel zeigt die Verwendung `delete-lexicon`.

AWS CLI

Um ein Lexikon zu löschen

Im folgenden `delete-lexicon` Beispiel wird das angegebene Lexikon gelöscht.

```
aws polly delete-lexicon \  
  --name w3c
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden des DeleteLexicon Vorgangs](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [DeleteLexicon](#) in der AWS CLI Befehlsreferenz.

get-lexicon

Das folgende Codebeispiel zeigt, wie man es benutzt `get-lexicon`.

AWS CLI

Um den Inhalt eines Lexikons abzurufen

Im folgenden `get-lexicon` Beispiel wird der Inhalt des angegebenen Aussprachelexikons abgerufen.

```
aws polly get-lexicon \
  --name w3c
```

Ausgabe:

```
{
  "Lexicon": {
    "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=
\n\"1.0\" \n      xmlns=      \"http://www.w3.org/2005/01/pronunciation-lexicon
\n\" \n      xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" \n
xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon \n
http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\" \n
      alphabet=\"ipa\" \n      xml:lang=\"en-US\">\n  <lexeme>\n    <grapheme>W3C</
grapheme>\n    <alias>World Wide Web Consortium</alias>\n  </lexeme>\n</lexicon>
\n",
    "Name": "w3c"
  },
  "LexiconAttributes": {
    "Alphabet": "ipa",
    "LanguageCode": "en-US",
    "LastModified": 1603908910.99,
    "LexiconArn": "arn:aws:polly:us-west-2:880185128111:lexicon/w3c",
    "LexemesCount": 1,
    "Size": 492
  }
}
```

Weitere Informationen finden Sie unter [Verwenden des GetLexicon Vorgangs](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [GetLexicon](#) in der AWS CLI Befehlsreferenz.

get-speech-synthesis-task

Das folgende Codebeispiel zeigt, wie man es benutzt `get-speech-synthesis-task`.

AWS CLI

Um Informationen über eine Sprachsynthese-Aufgabe zu erhalten

Im folgenden `get-speech-synthesis-task` Beispiel werden Informationen über die angegebene Sprachsyntheseaufgabe abgerufen.

```
aws polly get-speech-synthesis-task \  
--task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Ausgabe:

```
{  
  "SynthesisTask": {  
    "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",  
    "TaskStatus": "completed",  
    "OutputUri": "https://s3.us-west-2.amazonaws.com/my-s3-  
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",  
    "CreationTime": 1603911042.689,  
    "RequestCharacters": 1311,  
    "OutputFormat": "mp3",  
    "TextType": "text",  
    "VoiceId": "Joanna"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen langer Audiodateien](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [GetSpeechSynthesisTask](#) in der AWS CLI Befehlsreferenz.

list-lexicons

Das folgende Codebeispiel zeigt, wie man es benutzt `list-lexicons`.

AWS CLI

Um Ihre Lexika aufzulisten

Das folgende `list-lexicons` Beispiel listet Ihre Aussprachelexika auf.

```
aws polly list-lexicons
```

Ausgabe:

```
{  
  "Lexicons": [  

```



```
{
  "Name": "w3c",
  "Attributes": {
    "Alphabet": "ipa",
    "LanguageCode": "en-US",
    "LastModified": 1603908910.99,
    "LexiconArn": "arn:aws:polly:us-east-2:123456789012:lexicon/w3c",
    "LexemesCount": 1,
    "Size": 492
  }
}
]
```

Weitere Informationen finden Sie unter [Verwenden des ListLexicons Vorgangs](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [ListLexicons](#) in der AWS CLI Befehlsreferenz.

list-speech-synthesis-tasks

Das folgende Codebeispiel zeigt, wie man es benutzt `list-speech-synthesis-tasks`.

AWS CLI

Um Ihre Sprachsynthese-Aufgaben aufzulisten

Das folgende `list-speech-synthesis-tasks` Beispiel listet Ihre Sprachsynthese-Aufgaben auf.

```
aws polly list-speech-synthesis-tasks
```

Ausgabe:

```
{
  "SynthesisTasks": [
    {
      "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
      "TaskStatus": "completed",
      "OutputUri": "https://s3.us-west-2.amazonaws.com/my-s3-bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
      "CreationTime": 1603911042.689,
    }
  ]
}
```

```

        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen langer Audiodateien](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [ListSpeechSynthesisTasks](#) in der AWS CLI Befehlsreferenz.

put-lexicon

Das folgende Codebeispiel zeigt, wie man es benutzt `put-lexicon`.

AWS CLI

Um ein Lexikon zu speichern

Im folgenden `put-lexicon` Beispiel wird das angegebene Aussprachelexikon gespeichert. Die `example.pls` Datei spezifiziert ein PLS W3C-kompatibles Lexikon.

```

aws polly put-lexicon \
  --name w3c \
  --content file://example.pls

```

Inhalt von `example.pls`

```

{
  <?xml version="1.0" encoding="UTF-8"?>
  <lexicon version="1.0"
    xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
      http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
    alphabet="ipa"
    xml:lang="en-US">
    <lexeme>
      <grapheme>W3C</grapheme>
      <alias>World Wide Web Consortium</alias>

```

```
    </lexeme>
  </lexicon>
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden des PutLexicon Vorgangs](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [PutLexicon](#) in der AWS CLI Befehlsreferenz.

start-speech-synthesis-task

Das folgende Codebeispiel zeigt, wie man es benutzt `start-speech-synthesis-task`.

AWS CLI

Um Text zu synthetisieren

Im folgenden `start-speech-synthesis-task` Beispiel wird der Text im angegebenen Bucket synthetisiert `text_file.txt` und die resultierende MP3 Datei gespeichert.

```
aws polly start-speech-synthesis-task \
  --output-format mp3 \
  --output-s3-bucket-name my-s3-bucket \
  --text file://text_file.txt \
  --voice-id Joanna
```

Ausgabe:

```
{
  "SynthesisTask": {
    "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
    "TaskStatus": "scheduled",
    "OutputUri": "https://s3.us-east-2.amazonaws.com/my-s3-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
    "CreationTime": 1603911042.689,
    "RequestCharacters": 1311,
    "OutputFormat": "mp3",
    "TextType": "text",
    "VoiceId": "Joanna"
  }
}
```

```
}  
}
```

Weitere Informationen finden Sie unter [Erstellen langer Audiodateien](#) im Amazon Polly Developer Guide.

- API-Einheiten finden Sie [StartSpeechSynthesisTask](#) in der AWS CLI Befehlsreferenz.

AWS-Preisliste Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS-Preisliste.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-services

Das folgende Codebeispiel zeigt die Verwendung `describe-services`.

AWS CLI

Um Service-Metadaten abzurufen

In diesem Beispiel werden die Metadaten für den EC2 Amazon-Servicecode abgerufen.

Befehl:

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --  
max-items 1
```

Ausgabe:

```
{
  "Services": [
    {
      "ServiceCode": "AmazonEC2",
      "AttributeNames": [
        "volumeType",
        "maxIopsvolume",
        "instance",
        "instanceCapacity10xlarge",
        "locationType",
        "instanceFamily",
        "operatingSystem",
        "clockSpeed",
        "LeaseContractLength",
        "ecu",
        "networkPerformance",
        "instanceCapacity8xlarge",
        "group",
        "maxThroughputvolume",
        "gpuMemory",
        "ebsOptimized",
        "elasticGpuType",
        "maxVolumeSize",
        "gpu",
        "processorFeatures",
        "intelAvxAvailable",
        "instanceCapacity4xlarge",
        "servicecode",
        "groupDescription",
        "processorArchitecture",
        "physicalCores",
        "productFamily",
        "enhancedNetworkingSupported",
        "intelTurboAvailable",
        "memory",
        "dedicatedEbsThroughput",
        "vcpu",
        "OfferingClass",
        "instanceCapacityLarge",
        "capacitystatus",
        "termType",
        "storage",
```

```

        "intelAvx2Available",
        "storageMedia",
        "physicalProcessor",
        "provisioned",
        "servicename",
        "PurchaseOption",
        "instanceCapacity18xlarge",
        "instanceType",
        "tenancy",
        "usagetype",
        "normalizationSizeFactor",
        "instanceCapacity2xlarge",
        "instanceCapacity16xlarge",
        "maxIopsBurstPerformance",
        "instanceCapacity12xlarge",
        "instanceCapacity32xlarge",
        "instanceCapacityXlarge",
        "licenseModel",
        "currentGeneration",
        "preInstalledSw",
        "location",
        "instanceCapacity24xlarge",
        "instanceCapacity9xlarge",
        "instanceCapacityMedium",
        "operation"
    ]
}
],
"FormatVersion": "aws_v1"
}

```

- API-Einheiten finden Sie [DescribeServices](#) in der AWS CLI Befehlsreferenz.

get-attribute-values

Das folgende Codebeispiel zeigt, wie man es benutzt `get-attribute-values`.

AWS CLI

Um eine Liste von Attributwerten abzurufen

Im folgenden `get-attribute-values` Beispiel wird eine Liste von Werten abgerufen, die für das angegebene Attribut verfügbar sind.

```
aws pricing get-attribute-values \  
  --service-code AmazonEC2 \  
  --attribute-name volumeType \  
  --max-items 2
```

Ausgabe:

```
{  
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",  
  "AttributeValues": [  
    {  
      "Value": "Cold HDD"  
    },  
    {  
      "Value": "General Purpose"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [GetAttributeValues AWS CLI-Befehlsreferenz](#).

get-products

Das folgende Codebeispiel zeigt, wie man es benutzt `get-products`.

AWS CLI

Um eine Liste von Produkten abzurufen

In diesem Beispiel wird eine Liste von Produkten abgerufen, die den angegebenen Kriterien entsprechen.

Befehl:

```
aws pricing get-products --filters file://filters.json --format-version aws_v1 --  
max-results 1 --service-code AmazonEC2
```

filters.json:

```
[
```

```
{
  "Type": "TERM_MATCH",
  "Field": "ServiceCode",
  "Value": "AmazonEC2"
},
{
  "Type": "TERM_MATCH",
  "Field": "volumeType",
  "Value": "Provisioned IOPS"
}
]
```

Ausgabe:

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVSagyIFn770XQ0zdUIc09BY6ucBG9itXAZGZF/
zioUz0sUKh6PCcPwa0yPZRiMePb986TeoKYB9155fw/
CyoMq5ymnGmT1Vj39TljjbbAlhcqnVfTmPIilx8Uy5bdDaBYy/e/20fw9Edzsykbs8LTBUbniDQ
+BBds5yeI9AQkUepuKk3aEahFPxJ55kx/zk",
  "PriceList": [
    "{\"product\":{\"productFamily\":\"Storage\"},\"attributes\":{\"storageMedia\":
\\\"SSD-backed\\\", \"maxThroughputvolume\": \"320 MB/sec\", \"volumeType\": \"Provisioned
IOPS\", \"maxIopsvolume\": \"20000\", \"servicecode\": \"AmazonEC2\", \"usagetype
\": \"APS1-EBS:VolumeUsage.piops\", \"locationType\": \"AWS Region\", \"location\":
\\\"Asia Pacific (Singapore)\\\", \"servicename\": \"Amazon Elastic Compute Cloud\",
\\\"maxVolumeSize\": \"16 TiB\", \"operation\": \"\"}, \"sku\": \"3MKHN58N7RDDVGKJ\"},
\\\"serviceCode\": \"AmazonEC2\", \"terms\": {\"OnDemand\": {\"3MKHN58N7RDDVGKJ.JRTCKXETXF
\": {\"priceDimensions\": {\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\": {\"unit\": \"GB-
Mo\", \"endRange\": \"Inf\", \"description\": \"$0.138 per GB-month of Provisioned
IOPS SSD (io1) provisioned storage - Asia Pacific (Singapore)\", \"appliesTo
\": [], \"rateCode\": \"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\", \"beginRange\":
\\\"0\", \"pricePerUnit\": {\"USD\": \"0.1380000000\"}}}, \"sku\": \"3MKHN58N7RDDVGKJ
\", \"effectiveDate\": \"2018-08-01T00:00:00Z\", \"offerTermCode\": \"JRTCKXETXF
\", \"termAttributes\": {}}}}, \"version\": \"20180808005701\", \"publicationDate\":
\\\"2018-08-08T00:57:01Z\\\"}
  ]
}
```

- API-Einheiten finden Sie [GetProducts](#) in der AWS CLI Befehlsreferenz.

AWS Private CA Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Private CA.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-certificate-authority-audit-report

Das folgende Codebeispiel zeigt die Verwendung `create-certificate-authority-audit-report`.

AWS CLI

Um einen Prüfbericht einer Zertifizierungsstelle zu erstellen

Der folgende `create-certificate-authority-audit-report` Befehl erstellt einen Prüfbericht für die private Zertifizierungsstelle, die durch den identifiziert wurdeARN.

```
aws acm-pca create-certificate-authority-audit-report --certificate-authority-arn arn:aws:acm-pca:us-east-1:accountid:certificate-authority/12345678-1234-1234-1234-123456789012 --s3-bucket-name your-bucket-name --audit-report-response-format JSON
```

- API-Einheiten finden Sie [CreateCertificateAuthorityAuditReport](#) in der AWS CLI Befehlsreferenz.

create-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `create-certificate-authority`.

AWS CLI

Um eine private Zertifizierungsstelle zu erstellen

Der folgende `create-certificate-authority` Befehl erstellt eine private Zertifizierungsstelle in Ihrem AWS Konto.

```
aws acm-pca create-certificate-authority --certificate-authority-configuration
file://C:\ca_config.txt --revocation-configuration file://C:\revoke_config.txt --
certificate-authority-type "SUBORDINATE" --idempotency-token 98256344
```

- API-Einheiten finden Sie [CreateCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

`delete-certificate-authority`

Das folgende Codebeispiel zeigt die Verwendung `delete-certificate-authority`.

AWS CLI

Um eine private Zertifizierungsstelle zu löschen

Der folgende `delete-certificate-authority` Befehl löscht die Zertifizierungsstelle, die durch den ARN identifiziert wurde.

```
aws acm-pca delete-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- API-Einheiten finden Sie [DeleteCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

`describe-certificate-authority-audit-report`

Das folgende Codebeispiel zeigt die Verwendung `describe-certificate-authority-audit-report`.

AWS CLI

Um einen Prüfbericht für eine Zertifizierungsstelle zu beschreiben

Der folgende `describe-certificate-authority-audit-report` Befehl listet Informationen zum angegebenen Prüfbericht für die Zertifizierungsstelle auf, die durch den identifizierten ARN.

```
aws acm-pca describe-certificate-authority-audit-report --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/99999999-8888-7777-6666-555555555555 --audit-report-  
id 11111111-2222-3333-4444-555555555555
```

- API-Einheiten finden Sie [DescribeCertificateAuthorityAuditReport](#) in der AWS CLI Befehlsreferenz.

describe-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `describe-certificate-authority`.

AWS CLI

Um eine private Zertifizierungsstelle zu beschreiben

Der folgende `describe-certificate-authority` Befehl listet Informationen über die private Zertifizierungsstelle auf, die durch die identifiziert wurde ARN.

```
aws acm-pca describe-certificate-authority --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012
```

- API-Einheiten finden Sie [DescribeCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

get-certificate-authority-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-certificate-authority-certificate`.

AWS CLI

Um ein Zertifikat einer Zertifizierungsstelle (CA) abzurufen

Mit dem folgenden `get-certificate-authority-certificate` Befehl werden das Zertifikat und die Zertifikatskette für die private Zertifizierungsstelle abgerufen, die von der ARN angegeben wurde.

```
aws acm-pca get-certificate-authority-certificate --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012 --output text
```

- API-Einheiten finden Sie unter [GetCertificateAuthorityCertificate AWS CLI Befehlsreferenz](#).

get-certificate-authority-csr

Das folgende Codebeispiel zeigt die Verwendung `get-certificate-authority-csr`.

AWS CLI

Um die Zertifikatsignieranforderung für eine Zertifizierungsstelle abzurufen

Der folgende `get-certificate-authority-csr` Befehl ruft die CSR für die private Zertifizierungsstelle ab, die von der ARN angegeben wurde.

```
aws acm-pca get-certificate-authority-csr --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012 --output text
```

- API-Einheiten finden Sie [GetCertificateAuthorityCsr](#) in der AWS CLI Befehlsreferenz.

get-certificate

Das folgende Codebeispiel zeigt die Verwendung `get-certificate`.

AWS CLI

Um ein ausgestelltes Zertifikat abzurufen

Im folgenden `get-certificate` Beispiel wird ein Zertifikat von der angegebenen privaten Zertifizierungsstelle abgerufen.

```
aws acm-pca get-certificate \  
  --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012 \  
  --certificate-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012/  
certificate/6707447683a9b7f4055627ffd55cebcc \  
  --output text
```

Ausgabe:

```
-----BEGIN CERTIFICATE-----
```

```

MIIEDzCCAvegAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvrZMwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqsKtu/KSS7YDjK0
KQny73D6LtmD0EbAyq10XiDxqY41lvKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE----- -----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIRA0skdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
kdRGB6P2hpxstD0UIwAoCbhoaWwfA4ybJznf+j0QhAziNlRdKQRR8n0DWpKt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----

```

Der erste Teil der Ausgabe ist das Zertifikat selbst. Der zweite Teil ist die Zertifikatskette, die mit dem Root-CA-Zertifikat verknüpft ist. Beachten Sie, dass bei Verwendung der `--output text` Option ein TAB Zeichen zwischen den beiden Zertifikatsteilen eingefügt wird (das ist die Ursache für den eingezogenen Text). Wenn Sie beabsichtigen, diese Ausgabe zu verwenden und die Zertifikate mit anderen Tools zu analysieren, müssen Sie das TAB Zeichen möglicherweise entfernen, damit es korrekt verarbeitet wird.

- API-Einheiten finden Sie [GetCertificate](#) in der AWS CLI Befehlsreferenz.

import-certificate-authority-certificate

Das folgende Codebeispiel zeigt die Verwendung `import-certificate-authority-certificate`.

AWS CLI

Um Ihr Zertifizierungsstellenzertifikat zu importieren ACM PCA

Mit dem folgenden `import-certificate-authority-certificate` Befehl wird das signierte private CA-Zertifikat für die durch ARN angegebene Zertifizierungsstelle importiert ACMPCA.

```

aws acm-pca import-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --certificate file://C:\ca_cert.pem
--certificate-chain file://C:\ca_cert_chain.pem

```

- API-Einheiten finden Sie [ImportCertificateAuthorityCertificate](#) in der AWS CLI Befehlsreferenz.

issue-certificate

Das folgende Codebeispiel zeigt die Verwendung `issue-certificate`.

AWS CLI

Um ein privates Zertifikat auszustellen

Der folgende `issue-certificate` Befehl verwendet die von der angegebene private ZertifizierungsstelleARN, um ein privates Zertifikat auszustellen.

```
aws acm-pca issue-certificate --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012
--csr file://C:\cert_1.csr --signing-algorithm "SHA256WITHRSA" --validity
Value=365,Type="DAYS" --idempotency-token 1234
```

- APIEinzelheiten finden Sie [IssueCertificate](#) in der AWS CLI Befehlsreferenz.

list-certificate-authorities

Das folgende Codebeispiel zeigt die Verwendung `list-certificate-authorities`.

AWS CLI

Um Ihre privaten Zertifizierungsstellen aufzulisten

Der folgende `list-certificate-authorities` Befehl listet Informationen über alle privaten Daten CAs in Ihrem Konto auf.

```
aws acm-pca list-certificate-authorities --max-results 10
```

- APIEinzelheiten finden Sie [ListCertificateAuthorities](#) in der AWS CLI Befehlsreferenz.

list-tags

Das folgende Codebeispiel zeigt die Verwendung `list-tags`.

AWS CLI

Um die Tags für Ihre Zertifizierungsstelle aufzulisten

Der folgende `list-tags` Befehl listet die Tags auf, die der von der angegebenen privaten Zertifizierungsstelle zugeordnet sindARN.

```
aws acm-pca list-tags --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --max-results 10
```

- APIEinzelheiten finden Sie [ListTags](#) in der AWS CLI Befehlsreferenz.

revoke-certificate

Das folgende Codebeispiel zeigt die Verwendung `revoke-certificate`.

AWS CLI

Um ein privates Zertifikat zu widerrufen

Der folgende `revoke-certificate` Befehl widerruft ein privates Zertifikat von der Zertifizierungsstelle, die durch den ARN identifiziert wurde.

```
aws acm-pca revoke-certificate --certificate-authority-arn arn:aws:acm-pca:us-west-2:1234567890:certificate-authority/12345678-1234-1234-1234-123456789012 --certificate-serial 67:07:44:76:83:a9:b7:f4:05:56:27:ff:d5:5c:eb:cc --revocation-reason "KEY_COMPROMISE"
```

- APIEinzelheiten finden Sie [RevokeCertificate](#) in der AWS CLI Befehlsreferenz.

tag-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `tag-certificate-authority`.

AWS CLI

Um Tags an eine private Zertifizierungsstelle anzuhängen

Mit dem folgenden `tag-certificate-authority` Befehl werden ein oder mehrere Tags an Ihre private CA angehängt.

```
aws acm-pca tag-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice
```

- API-Einheiten finden Sie [TagCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

untag-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `untag-certificate-authority`.

AWS CLI

Um ein oder mehrere Tags von Ihrer privaten Zertifizierungsstelle zu entfernen

Der folgende `untag-certificate-authority` Befehl entfernt Tags aus der privaten Zertifizierungsstelle, die durch den identifiziert wurdeARN.

```
aws acm-pca untag-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --tags Key=Purpose,Value=Website
```

- API-Einheiten finden Sie [UntagCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

update-certificate-authority

Das folgende Codebeispiel zeigt die Verwendung `update-certificate-authority`.

AWS CLI

Um die Konfiguration Ihrer privaten Zertifizierungsstelle zu aktualisieren

Der folgende `update-certificate-authority` Befehl aktualisiert den Status und die Konfiguration der privaten Zertifizierungsstelle, die durch den identifiziert wurdeARN.

```
aws acm-pca update-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-1232456789012 --revocation-configuration file://C:\revoke_config.txt --status "DISABLED"
```

- API-Einheiten finden Sie [UpdateCertificateAuthority](#) in der AWS CLI Befehlsreferenz.

AWS Proton Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Proton.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-service-instance-deployment

Das folgende Codebeispiel zeigt die Verwendung `cancel-service-instance-deployment`.

AWS CLI

Um die Bereitstellung einer Service-Instanz abzurechnen

Im folgenden `cancel-service-instance-deployment` Beispiel wird die Bereitstellung einer Serviceinstanz abgebrochen.

```
aws proton cancel-service-instance-deployment \  
  --service-instance-name "instance-one" \  
  --service-name "simple-svc"
```

Ausgabe:

```
{  
  "serviceInstance": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-  
instance/instance-one",  
    "createdAt": "2021-04-02T21:29:59.962000+00:00",  
    "deploymentStatus": "CANCELLING",  
    "environmentName": "simple-env",  
    "lastDeploymentAttemptedAt": "2021-04-02T21:45:15.406000+00:00",  
    "lastDeploymentSucceededAt": "2021-04-02T21:38:00.823000+00:00",  
    "name": "instance-one",  
    "serviceName": "simple-svc",
```

```

    "spec": "proton: ServiceSpec\npipeline:\n
my_sample_pipeline_optional_input: abc\n my_sample_pipeline_required_input:
'123'\ninstances:\n- name: my-instance\n environment: MySimpleEnv
\n spec:\n my_sample_service_instance_optional_input: def\n
my_sample_service_instance_required_input: '456'\n- name: my-other-instance\n
environment: MySimpleEnv\n spec:\n my_sample_service_instance_required_input:
'789'\n",
    "templateMajorVersion": "1",
    "templateMinorVersion": "1",
    "templateName": "svc-simple"
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer Dienstinstanz](#) im AWS Proton Administrator Guide oder [Aktualisieren einer Dienstinstanz](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie [CancelServiceInstanceDeployment](#) in der AWS CLI Befehlsreferenz.

cancel-service-pipeline-deployment

Das folgende Codebeispiel zeigt die Verwendung `cancel-service-pipeline-deployment`.

AWS CLI

Um eine Service-Pipeline-Bereitstellung abubrechen

Im folgenden `cancel-service-pipeline-deployment` Beispiel wird eine Service-Pipeline-Bereitstellung storniert.

```
aws proton cancel-service-pipeline-deployment \
  --service-name "simple-svc"
```

Ausgabe:

```
{
  "pipeline": {
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline",
    "createdAt": "2021-04-02T21:29:59.962000+00:00",
    "deploymentStatus": "CANCELLING",
    "lastDeploymentAttemptedAt": "2021-04-02T22:02:45.095000+00:00",
    "lastDeploymentSucceededAt": "2021-04-02T21:39:28.991000+00:00",
  }
}
```

```

    "templateMajorVersion": "1",
    "templateMinorVersion": "1",
    "templateName": "svc-simple"
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren einer Service-Pipeline](#) im AWS Proton Administrator Guide oder [Aktualisieren einer Service-Pipeline](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie unter [CancelServicePipelineDeployment AWS CLI Befehlsreferenz](#).

create-service

Das folgende Codebeispiel zeigt die Verwendung `create-service`.

AWS CLI

Um einen Dienst zu erstellen

Im folgenden `create-service` Beispiel wird ein Dienst mit einer Dienstpipeline erstellt.

```

aws proton create-service \
  --name "MySimpleService" \
  --template-name "fargate-service" \
  --template-major-version "1" \
  --branch-name "mainline" \
  --repository-connection-arn "arn:aws:codestar-connections:region-id:account-
id:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
  --repository-id "myorg/myapp" \
  --spec file://spec.yaml

```

Inhalt von `spec.yaml`:

```

proton: ServiceSpec

pipeline:
  my_sample_pipeline_required_input: "hello"
  my_sample_pipeline_optional_input: "bye"

instances:
  - name: "acme-network-dev"
    environment: "ENV_NAME"

```

```
spec:
  my_sample_service_instance_required_input: "hi"
  my_sample_service_instance_optional_input: "ho"
```

Ausgabe:

```
{
  "service": {
    "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
    "createdAt": "2020-11-18T19:50:27.460000+00:00",
    "lastModifiedAt": "2020-11-18T19:50:27.460000+00:00",
    "name": "MySimpleService",
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "repositoryId": "myorg/myapp",
    "status": "CREATE_IN_PROGRESS",
    "templateName": "fargate-service"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen eines Dienstes](#) im The AWS Proton Administrator Guide und [Create a Service](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie unter [CreateService AWS CLI](#) Befehlsreferenz.

delete-service

Das folgende Codebeispiel zeigt die Verwendung `delete-service`.

AWS CLI

Um einen Dienst zu löschen

Im folgenden `delete-service` Beispiel wird ein Dienst gelöscht.

```
aws proton delete-service \
  --name "simple-svc"
```

Ausgabe:

```
{
```

```
"service": {
  "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",
  "branchName": "mainline",
  "createdAt": "2020-11-28T22:40:50.512000+00:00",
  "description": "Edit by updating description",
  "lastModifiedAt": "2020-11-29T00:30:39.248000+00:00",
  "name": "simple-svc",
  "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "repositoryId": "myorg/myapp",
  "status": "DELETE_IN_PROGRESS",
  "templateName": "fargate-service"
}
```

Weitere Informationen finden Sie unter [Löschen eines Dienstes](#) im The AWS Proton Administrator Guide.

- API-Einheiten finden Sie unter [DeleteService AWS CLI](#) Befehlsreferenz.

get-service-instance

Das folgende Codebeispiel zeigt die Verwendung `get-service-instance`.

AWS CLI

Um Details zur Serviceinstanz abzurufen

Im folgenden `get-service-instance` Beispiel werden Detaildaten für eine Dienstinstanz abgerufen.

```
aws proton get-service-instance \
  --name "instance-one" \
  --service-name "simple-svc"
```

Ausgabe:

```
{
  "serviceInstance": {
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
    "createdAt": "2020-11-28T22:40:50.512000+00:00",
```

```

    "deploymentStatus": "SUCCEEDED",
    "environmentName": "simple-env",
    "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
    "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
    "name": "instance-one",
    "serviceName": "simple-svc",
    "spec": "proton: ServiceSpec\npipeline:\n
my_sample_pipeline_optional_input: hello world\n
my_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\n
environment: my-simple-env\n spec:\n   my_sample_service_instance_optional_input:
0la\n   my_sample_service_instance_required_input: Ciao\n",
    "templateMajorVersion": "1",
    "templateMinorVersion": "0",
    "templateName": "svc-simple"
  }
}

```

Weitere Informationen finden Sie unter [Servicedaten anzeigen](#) im The AWS Proton Administrator Guide oder [View Service Data](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie unter [GetServiceInstance AWS CLI](#) Befehlsreferenz.

get-service

Das folgende Codebeispiel zeigt die Verwendung `get-service`.

AWS CLI

Um Servicedetails zu erhalten

Im folgenden `get-service` Beispiel werden Detaildaten für einen Dienst abgerufen.

```
aws proton get-service \
  --name "simple-svc"
```

Ausgabe:

```

{
  "service": {
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",
    "branchName": "mainline",
    "createdAt": "2020-11-28T22:40:50.512000+00:00",

```

```

    "lastModifiedAt": "2020-11-28T22:44:51.207000+00:00",
    "name": "simple-svc",
    "pipeline": {
      "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
pipeline/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "createdAt": "2020-11-28T22:40:50.512000+00:00",
      "deploymentStatus": "SUCCEEDED",
      "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
      "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
      "spec": "proton: ServiceSpec\npipeline:\n
my_sample_pipeline_required_input: hello\n my_sample_pipeline_optional_input:
bye\ninstances:\n- name: instance-svc-simple\n environment: my-simple-
env\n spec:\n  my_sample_service_instance_required_input: hi\n
my_sample_service_instance_optional_input: ho\n",
      "templateMajorVersion": "1",
      "templateMinorVersion": "1",
      "templateName": "svc-simple"
    },
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "repositoryId": "myorg/myapp",
    "spec": "proton: ServiceSpec\npipeline:\n
my_sample_pipeline_required_input: hello\n my_sample_pipeline_optional_input:
bye\ninstances:\n- name: instance-svc-simple\n environment: my-simple-
env\n spec:\n  my_sample_service_instance_required_input: hi\n
my_sample_service_instance_optional_input: ho\n",
    "status": "ACTIVE",
    "templateName": "svc-simple"
  }
}

```

Weitere Informationen finden Sie unter [Servicedaten anzeigen](#) im The AWS Proton Administrator Guide oder [View Service Data](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie unter [GetService AWS CLI](#) Befehlsreferenz.

list-service-instances

Das folgende Codebeispiel zeigt die Verwendung `list-service-instances`.

AWS CLI

Beispiel 1: Um alle Dienstinstanzen aufzulisten

Das folgende `list-service-instances` Beispiel listet Dienstinstanzen auf.

```
aws proton list-service-instances
```

Ausgabe:

```
{
  "serviceInstances": [
    {
      "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
service-instance/instance-one",
      "createdAt": "2020-11-28T22:40:50.512000+00:00",
      "deploymentStatus": "SUCCEEDED",
      "environmentArn": "arn:aws:proton:region-id:123456789012:environment/
simple-env",
      "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
      "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
      "name": "instance-one",
      "serviceName": "simple-svc",
      "templateMajorVersion": "1",
      "templateMinorVersion": "0",
      "templateName": "fargate-service"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Dienstinstanzdaten anzeigen](#) im The AWS Proton Administrator Guide oder [View Service Instance-Daten](#) im The AWS Proton User Guide.

Beispiel 2: Um die angegebene Dienstinstanz aufzulisten

Im folgenden `get-service-instance` Beispiel wird eine Dienstinstanz abgerufen.

```
aws proton get-service-instance \
  --name "instance-one" \
  --service-name "simple-svc"
```

Ausgabe:

```
{
  "serviceInstance": {
```



```

    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-
instance/instance-one",
    "createdAt": "2020-11-28T22:40:50.512000+00:00",
    "deploymentStatus": "SUCCEEDED",
    "environmentName": "simple-env",
    "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
    "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
    "name": "instance-one",
    "serviceName": "simple-svc",
    "spec": "proton: ServiceSpec\npipeline:\n
my_sample_pipeline_optional_input: hello world\n
my_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\n
environment: my-simple-env\n spec:\n   my_sample_service_instance_optional_input:
Ola\n   my_sample_service_instance_required_input: Ciao\n",
    "templateMajorVersion": "1",
    "templateMinorVersion": "0",
    "templateName": "svc-simple"
  }
}

```

Weitere Informationen finden Sie unter [Dienstinstanzdaten anzeigen](#) im The AWS Proton Administrator Guide oder [View Service Instance-Daten](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie [ListServiceInstances](#) in der AWS CLI Befehlsreferenz.

update-service-instance

Das folgende Codebeispiel zeigt die Verwendung `update-service-instance`.

AWS CLI

Um eine Dienstinstanz auf eine neue Nebenversion zu aktualisieren

Im folgenden `update-service-instance` Beispiel wird eine Dienstinstanz auf eine neue Nebenversion ihrer Dienstvorlage aktualisiert, die eine neue Instanz namens "my-other-instance" mit einer neuen erforderlichen Eingabe hinzufügt.

```

aws proton update-service-instance \
  --service-name "simple-svc" \
  --spec "file://service-spec.yaml" \
  --template-major-version "1" \
  --template-minor-version "1" \

```

```
--deployment-type "MINOR_VERSION" \  
--name "instance-one"
```

Inhalt von `service-spec.yaml`:

```
proton: ServiceSpec  
pipeline:  
  my_sample_pipeline_optional_input: "abc"  
  my_sample_pipeline_required_input: "123"  
instances:  
  - name: "instance-one"  
    environment: "simple-env"  
    spec:  
      my_sample_service_instance_optional_input: "def"  
      my_sample_service_instance_required_input: "456"  
  - name: "my-other-instance"  
    environment: "simple-env"  
    spec:  
      my_sample_service_instance_required_input: "789"
```

Ausgabe:

```
{  
  "serviceInstance": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-  
instance/instance-one",  
    "createdAt": "2021-04-02T21:29:59.962000+00:00",  
    "deploymentStatus": "IN_PROGRESS",  
    "environmentName": "arn:aws:proton:region-id:123456789012:environment/  
simple-env",  
    "lastDeploymentAttemptedAt": "2021-04-02T21:38:00.823000+00:00",  
    "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",  
    "name": "instance-one",  
    "serviceName": "simple-svc",  
    "templateMajorVersion": "1",  
    "templateMinorVersion": "0",  
    "templateName": "svc-simple"  
  }  
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer Dienstinstantz](#) im AWS Proton Administrator Guide oder [Aktualisieren einer Dienstinstantz](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie [UpdateServiceInstance](#) in der AWS CLI Befehlsreferenz.

update-service-pipeline

Das folgende Codebeispiel zeigt die Verwendung `update-service-pipeline`.

AWS CLI

Um eine Service-Pipeline zu aktualisieren

Im folgenden `update-service-pipeline` Beispiel wird eine Dienstpipeline auf eine neue Nebenversion ihrer Dienstvorlage aktualisiert.

```
aws proton update-service-pipeline \
  --service-name "simple-svc" \
  --spec "file://service-spec.yaml" \
  --template-major-version "1" \
  --template-minor-version "1" \
  --deployment-type "MINOR_VERSION"
```

Ausgabe:

```
{
  "pipeline": {
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "createdAt": "2021-04-02T21:29:59.962000+00:00",
    "deploymentStatus": "IN_PROGRESS",
    "lastDeploymentAttemptedAt": "2021-04-02T21:39:28.991000+00:00",
    "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
    "spec": "proton: ServiceSpec\n\npipeline:\n
my_sample_pipeline_optional_input: \"abc\"\n
my_sample_pipeline_required_input:
\"123\"\n\ninstances:\n
- name: \"my-instance\"\n
  environment: \"MySimpleEnv\"\n\n
spec:\n
  my_sample_service_instance_optional_input: \"def\"\n\n
  my_sample_service_instance_required_input: \"456\"\n
  - name: \"my-other-instance\"\n
  environment: \"MySimpleEnv\"\n\n
spec:\n
my_sample_service_instance_required_input: \"789\"\n\n",
    "templateMajorVersion": "1",
    "templateMinorVersion": "0",
    "templateName": "svc-simple"
  }
}
```

Weitere Informationen finden Sie unter [Aktualisieren einer Service-Pipeline](#) im AWS Proton Administrator Guide oder [Aktualisieren einer Service-Pipeline](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie unter [UpdateServicePipeline AWS CLI-Befehlsreferenz](#).

update-service

Das folgende Codebeispiel zeigt die Verwendung `update-service`.

AWS CLI

Um einen Dienst zu aktualisieren

Im folgenden `update-service` Beispiel wird eine Dienstbeschreibung bearbeitet.

```
aws proton update-service \  
  --name "MySimpleService" \  
  --description "Edit by updating description"
```

Ausgabe:

```
{  
  "service": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",  
    "branchName": "mainline",  
    "createdAt": "2021-03-12T22:39:42.318000+00:00",  
    "description": "Edit by updating description",  
    "lastModifiedAt": "2021-03-12T22:44:21.975000+00:00",  
    "name": "MySimpleService",  
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-  
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "repositoryId": "myorg/myapp",  
    "status": "ACTIVE",  
    "templateName": "fargate-service"  
  }  
}
```

Weitere Informationen finden Sie unter [Bearbeiten eines Dienstes](#) im AWS Proton Administrator Guide oder [Bearbeiten eines Dienstes](#) im The AWS Proton User Guide.

- API-Einheiten finden Sie [UpdateService](#) in der AWS CLI Befehlsreferenz.

QLDBBeispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren QLDB.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-journal-kinesis-stream

Das folgende Codebeispiel zeigt die Verwendung `cancel-journal-kinesis-stream`.

AWS CLI

Um einen Journal-Stream abubrechen

Im folgenden `cancel-journal-kinesis-stream` Beispiel wird der angegebene Journal-Stream aus einem Ledger gelöscht.

```
aws qldb cancel-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISckqwe4y25YyHLzYUFAf
```

Ausgabe:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFAf"  
}
```

Weitere Informationen finden Sie unter [Streaming-Journaldaten von Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [CancelJournalKinesisStream](#) in der AWS CLI Befehlsreferenz.

create-ledger

Das folgende Codebeispiel zeigt die Verwendung `create-ledger`.

AWS CLI

Beispiel 1: Um ein Ledger mit Standardeigenschaften zu erstellen

Im folgenden `create-ledger` Beispiel wird ein Ledger mit dem Namen `myExampleLedger` und dem Berechtigungsmodus erstellt. `STANDARD` Die optionalen Parameter für den Löschschutz und den AWS KMS Schlüssel sind nicht angegeben, sodass sie standardmäßig auf einen AWS eigenen KMS Schlüssel `true` bzw. einen eigenen Schlüssel setzen.

```
aws qlldb create-ledger \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

Ausgabe:

```
{  
  "State": "CREATING",  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": true,  
  "CreationDateTime": 1568839243.951,  
  "Name": "myExampleLedger",  
  "PermissionsMode": "STANDARD"  
}
```

Beispiel 2: Um ein Ledger mit deaktiviertem Löschschutz, einem vom Kunden verwalteten KMS Schlüssel und bestimmten Tags zu erstellen

Im folgenden `create-ledger` Beispiel wird ein Ledger mit dem Namen `myExampleLedger2` und dem Berechtigungsmodus erstellt. `STANDARD` Die Löschschutzfunktion ist deaktiviert, der angegebene vom Kunden verwaltete KMS Schlüssel wird für die Verschlüsselung im Ruhezustand verwendet, und die angegebenen Tags werden an die Ressource angehängt.

```
aws qlldb create-ledger \  
  --name myExampleLedger2 \  
  --tags Tag1=tag1,Tag2=tag2
```

```
--permissions-mode STANDARD \  
--no-deletion-protection \  
--kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111 \  
--tags IsTest=true,Domain=Test
```

Ausgabe:

```
{  
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger2",  
  "DeletionProtection": false,  
  "CreationDateTime": 1568839543.557,  
  "State": "CREATING",  
  "Name": "myExampleLedger2",  
  "PermissionsMode": "STANDARD",  
  "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Basic Operations for Amazon QLDB Ledgers](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [CreateLedger](#) in der AWS CLI Befehlsreferenz.

delete-ledger

Das folgende Codebeispiel zeigt die Verwendung `delete-ledger`.

AWS CLI

Um ein Ledger zu löschen

Im folgenden `delete-ledger` Beispiel wird das angegebene Ledger gelöscht.

```
aws qldb delete-ledger \  
  --name myExampleLedger
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Basic Operations for Amazon QLDB Ledgers](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [DeleteLedger](#) in der AWS CLI Befehlsreferenz.

describe-journal-kinesis-stream

Das folgende Codebeispiel zeigt die Verwendung `describe-journal-kinesis-stream`.

AWS CLI

Um einen Journal-Stream zu beschreiben

Im folgenden `describe-journal-kinesis-stream` Beispiel werden die Details für den angegebenen Journal-Stream aus einem Ledger angezeigt.

```
aws qlldb describe-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISCKqwe4y25YyHLzYUFaf
```

Ausgabe:

```
{  
  "Stream": {  
    "LedgerName": "myExampleLedger",  
    "CreationTime": 1591221984.677,  
    "InclusiveStartTime": 1590710400.0,  
    "ExclusiveEndTime": 1590796799.0,  
    "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",  
    "StreamId": "7ISCKqwe4y25YyHLzYUFaf",  
    "Arn": "arn:aws:qlldb:us-east-1:123456789012:stream/  
myExampleLedger/7ISCKqwe4y25YyHLzYUFaf",  
    "Status": "ACTIVE",  
    "KinesisConfiguration": {  
      "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-  
qlldb",  
      "AggregationEnabled": true  
    },  
    "StreamName": "myExampleLedger-stream"  
  }  
}
```

Weitere Informationen finden Sie unter [Streaming-Journaldaten von Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [DescribeJournalKinesisStream](#) in der AWS CLI Befehlsreferenz.

describe-journal-s3-export

Das folgende Codebeispiel zeigt die Verwendung `describe-journal-s3-export`.

AWS CLI

Um einen Journalexportjob zu beschreiben

Im folgenden `describe-journal-s3-export` Beispiel werden die Details für den angegebenen Exportauftrag aus einem Hauptbuch angezeigt.

```
aws qlldb describe-journal-s3-export \  
  --name myExampleLedger \  
  --export-id ADR20NPKN5LINYGb4dp7yZ
```

Ausgabe:

```
{  
  "ExportDescription": {  
    "S3ExportConfiguration": {  
      "Bucket": "awsExampleBucket",  
      "Prefix": "ledgerexport1/",  
      "EncryptionConfiguration": {  
        "ObjectEncryptionType": "SSE_S3"  
      }  
    },  
    "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",  
    "Status": "COMPLETED",  
    "ExportCreationTime": 1568847801.418,  
    "InclusiveStartTime": 1568764800.0,  
    "ExclusiveEndTime": 1568847599.0,  
    "LedgerName": "myExampleLedger",  
    "ExportId": "ADR20NPKN5LINYGb4dp7yZ"  
  }  
}
```

Weitere Informationen finden Sie unter [Exportieren Ihres Journals in Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie unter [DescribeJournalS3Export](#) in der AWS CLI Befehlsreferenz.

describe-ledger

Das folgende Codebeispiel zeigt die Verwendung `describe-ledger`

AWS CLI

Um ein Hauptbuch zu beschreiben

Im folgenden `describe-ledger` Beispiel werden die Details für das angegebene Ledger angezeigt.

```
aws qlldb describe-ledger \  
  --name myExampleLedger
```

Ausgabe:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "State": "ACTIVE",  
  "Name": "myExampleLedger",  
  "DeletionProtection": true,  
  "PermissionsMode": "STANDARD",  
  "EncryptionDescription": {  
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",  
    "EncryptionStatus": "ENABLED"  
  }  
}
```

Weitere Informationen finden Sie unter [Basic Operations for Amazon QLDB Ledgers](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [DescribeLedger](#) in der AWS CLI Befehlsreferenz.

export-journal-to-s3

Das folgende Codebeispiel zeigt die Verwendung `export-journal-to-s3`.

AWS CLI

Um Journalblöcke nach S3 zu exportieren

Im folgenden `export-journal-to-s3` Beispiel wird aus einem Hauptbuch mit dem Namen `myExampleLedger` ein Exportauftrag für Journalblöcke innerhalb eines angegebenen Datums- und Zeitbereichs erstellt. Der Exportjob schreibt die Blöcke in einen angegebenen Amazon S3 S3-Bucket.

```
aws qlldb export-journal-to-s3 \  
  --name myExampleLedger \  
  --inclusive-start-time 2019-09-18T00:00:00Z \  
  --exclusive-end-time 2019-09-18T22:59:59Z \  
  --role-arn arn:aws:iam::123456789012:role/my-s3-export-role \  
  --s3-export-configuration file://my-s3-export-config.json
```

Inhalt von `my-s3-export-config.json`:

```
{  
  "Bucket": "awsExampleBucket",  
  "Prefix": "ledgerexport1/",  
  "EncryptionConfiguration": {  
    "ObjectEncryptionType": "SSE_S3"  
  }  
}
```

Ausgabe:

```
{  
  "ExportId": "ADR20NPKN5LINYGb4dp7yZ"  
}
```

Weitere Informationen finden Sie unter [Exportieren Ihres Journals in Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie unter [ExportJournalToS3](#) in der AWS CLI Befehlsreferenz.

get-block

Das folgende Codebeispiel zeigt die Verwendung `get-block`.

AWS CLI

Beispiel 1: Um anhand von Eingabedateien einen Journalblock und einen Nachweis zur Überprüfung zu erhalten

Im folgenden `get-block` Beispiel werden ein Blockdatenobjekt und ein Nachweis aus dem angegebenen Ledger angefordert. Die Anforderung bezieht sich auf eine angegebene Digest-Tip- und Blockadresse.

```
aws qlldb get-block \
  --name vehicle-registration \
  --block-address file://myblockaddress.json \
  --digest-tip-address file://mydigesttipaddress.json
```

Inhalt von `myblockaddress.json`:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"
}
```

Inhalt von `mydigesttipaddress.json`:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"
}
```

Ausgabe:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},transactionId:\\"FnQeJBAicTX0Ah32ZnVtSX\\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:{{NoChM92yKRuJAB/jeLd1VnYn4DHWiF071ACfic9uHc=}},entriesHash:{{105L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:{{7kewBXhpdbClcZKxhVmpoMhpUG0JtWQD0iY2LPfZkYA=}},entriesHashList:{{eRSwnmAM7WWANWDd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},{{y5cCB7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX+KR+W/EJl4qD1mmV77KZQg=}}},transactionInfo:{statements:[{statement:\\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\\nINSERT INTO r.Owners.SecondaryOwners\\n  VALUE { 'PersonId' : 'CMVdR77XP8zAglmmFDGTVt' }\\n},startTime:2019-09-16T19:37:05.302Z,statementDigest:{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:[JUJgkIcNbhS2goq8RqLuZ4:{tableName:\\"VehicleRegistration\\",tableId:\\"BFJKdXgzT9oF4wjMbuXy4G\\",statements:[0]}]},revisions:[{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},hash:
```

```
{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}}, data: {VIN:
  \ "1N4AL11D75C109151\ ", LicensePlateNumber: \ "LEWISR261LL\ ", State: \ "WA
  \ ", PendingPenaltyTicketAmount: 90.25, ValidFromDate: 2017-08-21, ValidToDate: 2020-05-11, Owners:
  {PrimaryOwner: {PersonId: \ "BFJKdXhnLRT27sXBnojNGW\ "}, SecondaryOwners:
  [{PersonId: \ "CMVdR77XP8zAg1mmFDGTvt\ "}]}, City: \ "Everett\ "}, metadata: {id:
  \ "JUJgkIcNbhS2goq8RqLuZ4\ ", version: 3, txTime: 2019-09-16T19:37:05.344Z, txId:
  \ "FnQeJBAicTX0Ah32ZnVtSX\ "}}}]}"
  },
  "Proof": {
    "IonText": "[[{{13+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
    {{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}}, {{9UnwnKSQT0i3ge1JMVa
    +tMIqCEDa0PTkWxmyHSn8UPQ=}}, {{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
    {{Irb5fNhbRiNEQ1VPhzlNgT/ZQPadSmgfdtMYcwkN0xoI=}}, {{+3CwpyG/ytf/
    vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}}, {{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
    +qE=}}]]]"
  }
}
```

Weitere Informationen finden Sie unter [Datenverifizierung bei Amazon QLDB im Amazon QLDB Developer Guide](#).

Beispiel 2: Um einen Journal-Block und einen Nachweis zur Überprüfung mithilfe einer Kurzsyntax zu erhalten

Im folgenden `get-block` Beispiel werden ein Blockdatenobjekt und ein Nachweis aus dem angegebenen Ledger mithilfe einer Kurzsyntax angefordert. Die Anforderung bezieht sich auf eine angegebene Digest-Tip-Adresse und eine Blockadresse.

```
aws qldb get-block \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:\ "KmA3ZZca7vAIiJAK9S5IwI\ ", sequenceNo:100}"'
  \
  --digest-tip-address 'IonText="{strandId:\ "KmA3ZZca7vAIiJAK9S5IwI\ ", sequenceNo:123}"'
```

Ausgabe:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\ "KmA3ZZca7vAIiJAK9S5IwI\ ", sequenceNo:100}, transactionId:\ "FnQeJBAicTX0Ah32ZnVtSX\ ", blockTimestamp:2019-09-16T19:37:05.360Z, blockHash:
```

```

{{NoChM92yKRuJAB/jeLd1VnYn4DHiWIf071ACfic9uHc=}}, entriesHash:
{{105L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}}, previousBlockHash:
{{7kewBXhpdBc1cZKxhVmpoMHPUGOJtWQD0iY2LPfZkYA=}}, entriesHashList:
[{{eRSwnmAM7WWANWDD5iG0yK+T4tDXyzUq6HZ/0fgLHos=}}, {{mHVex/
yJHAWjFPPwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCB7r7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}}, {{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}], transactionInfo: {statements: [{statement:
\FROM VehicleRegistration AS r \\\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\\n    VALUE { 'PersonId' :
'CMVdR77XP8zAg1mmFDGTvt' } \}], startTime: 2019-09-16T19:37:05.302Z, statementDigest:
{{jcgPX2vs0J0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}], documents:
{JUJgkIcNbhS2goq8RqLuZ4: {tableName: \"VehicleRegistration\", tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\", statements: [0]}}, revisions: [{blockAddress:
{strandId: \"KmA3ZZca7vAIiJAK9S5Iw1\", sequenceNo: 100}, hash:
{{mHVex/yJHAWjFPPwhBuH2GKXmKJjK2FBa9faqoUVNtg=}}, data: {VIN:
\"1N4AL11D75C109151\", LicensePlateNumber: \"LEWISR261LL\", State: \"WA
\", PendingPenaltyTicketAmount: 90.25, ValidFromDate: 2017-08-21, ValidToDate: 2020-05-11, Owners:
{PrimaryOwner: {PersonId: \"BFJKdXhnLRT27sXBnojNGW\"}, SecondaryOwners:
[{{PersonId: \"CMVdR77XP8zAg1mmFDGTvt\"}}]}, City: \"Everett\"}, metadata: {id:
\"JUJgkIcNbhS2goq8RqLuZ4\", version: 3, txTime: 2019-09-16T19:37:05.344Z, txId:
\FnQeJBAicTX0Ah32ZnVtSX\"}}]}]
},
  \"Proof\": {
    \"IonText\": \"[{{13+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}}, {{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}}, {{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhz1nGT/ZQPadSmgfdtMYcwkN0xoI=}}, {{+3CwpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}}, {{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
+qE=}}]\"
  }
}

```

Weitere Informationen finden Sie unter [Datenverifizierung bei Amazon QLDB im Amazon QLDB Developer Guide](#).

- API-Einheiten finden Sie [GetBlock](#) in der AWS CLI Befehlsreferenz.

get-digest

Das folgende Codebeispiel zeigt die Verwendung von `get-digest`.

AWS CLI

Um einen Digest für ein Hauptbuch zu erhalten

Im folgenden `get-digest` Beispiel wird für den letzten festgeschriebenen Block im Journal ein Digest aus dem angegebenen Ledger angefordert.

```
aws qlldb get-digest \  
  --name vehicle-registration
```

Ausgabe:

```
{  
  "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",  
  "DigestTipAddress": {  
    "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\", sequenceNo:123}"  
  }  
}
```

Weitere Informationen finden Sie unter [Datenverifizierung bei Amazon QLDB im Amazon QLDB Developer Guide](#).

- API-Einheiten finden Sie [GetDigest](#) in der AWS CLI Befehlsreferenz.

get-revision

Das folgende Codebeispiel zeigt die Verwendung `get-revision`.

AWS CLI

Beispiel 1: Um anhand von Eingabedateien eine Revision des Dokuments und einen Nachweis zur Überprüfung zu erhalten

Im folgenden `get-revision` Beispiel werden ein Revisionsdatenobjekt und ein Nachweis aus dem angegebenen Hauptbuch angefordert. Die Anforderung bezieht sich auf eine angegebene Übersichtsadresse, eine Dokument-ID und eine Blockadresse der Revision.

```
aws qlldb get-revision \  
  --name vehicle-registration \  
  --block-address file://myblockaddress.json \  
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \  
  --ledger-name my-ledger
```

```
--digest-tip-address file://mydigesttipaddress.json
```

Inhalt von myblockaddress.json:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"
}
```

Inhalt von mydigesttipaddress.json:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"
}
```

Ausgabe:

```
{
  "Revision": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},data:
{VIN:\\"1N4AL11D75C109151\\",LicensePlateNumber:\\"LEWISR261LL\\",State:\\"WA\\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\\"BFJKdXhnLRT27sXBnojNGW\\"},SecondaryOwners:
[[{PersonId:\\"CMVdR77XP8zAgImmFDGTvt\\"}]]},City:\\"Everett\\"},metadata:{id:
\\"JUJgkIcNbhS2goq8RqLuZ4\\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\\"FnQeJBAicTX0Ah32ZnVtSX\\"}}}"
  },
  "Proof": {
    "IonText": "[[{{eRSwnmAM7WWANWDd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvB08Lg+KgpjcnWAU=}},
{{7kewBXhpdBc1cZKxhVmpoMhpUG0JtWQD0iY2LPfZkYA=}},{{13+EXs69K1+rehlqyWLkt
+oHD1w4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWAA08010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkN0xoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
  }
}
```

Weitere Informationen finden Sie unter [Datenverifizierung bei Amazon QLDB im Amazon QLDB Developer Guide](#).

Beispiel 2: Um mithilfe einer Kurzschrift eine Überarbeitung des Dokuments und einen Nachweis zur Überprüfung zu erhalten

Im folgenden `get-revision` Beispiel werden ein Revisionsdatenobjekt und ein Nachweis aus dem angegebenen Hauptbuch mithilfe einer Kurzsyntax angefordert. Die Anforderung bezieht sich auf eine angegebene Digest-Tip-Adresse, eine Dokument-ID und eine Blockadresse der Revision.

```
aws qlldb get-revision \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5IwI\\",sequenceNo:100}"'
  \
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \
  --digest-tip-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5IwI\\",sequenceNo:123}"'
```

Ausgabe:

```
{
  "Revision": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5IwI\\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},data:
    {VIN:\\"1N4AL11D75C109151\\",LicensePlateNumber:\\"LEWISR261LL\\",State:\\"WA\\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
    {PrimaryOwner:{PersonId:\\"BFJKdXhnLRT27sXBnojNGW\\"},SecondaryOwners:
    [{PersonId:\\"CMVdR77XP8zAg1mmFDGTvt\\"}]}},City:\\"Everett\\"},metadata:{id:
    \\"JUJgkIcNbhS2goq8RqLuZ4\\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
    \\"FnQeJBAicTX0Ah32ZnVtSX\\"}}}"
  },
  "Proof": {
    "IonText": "[{{eRSwnmAM7WWANWd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
    +yJZVG1msM6gr2T52QvB08Lg+KgpjcnWAU=}},
    {{7kewBXhpdBc1cZKxhVmpoMhpUGOJtWQD0iY2LPfZkYA=}},{{13+EXs69K1+rehlqyWLkt
    +oHD1w4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},
    {{9UnwnKSQT0i3ge1JMva+tMIqCEDaOPTkwxmyHSn8UPQ=}},{{3nW6Vryghk
    +7pd6wFcTLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhz1nGT/
    ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwPYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfY=}},
    {{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
  }
}
```

Weitere Informationen finden Sie unter [Datenverifizierung bei Amazon QLDB im Amazon QLDB Developer Guide](#).

- API-Einheiten finden Sie [GetRevision](#) in der AWS CLI Befehlsreferenz.

list-journal-kinesis-streams-for-ledger

Das folgende Codebeispiel zeigt die Verwendung `list-journal-kinesis-streams-for-ledger`.

AWS CLI

Um Journal-Streams für ein Ledger aufzulisten

Das folgende `list-journal-kinesis-streams-for-ledger` Beispiel listet Journal-Streams für das angegebene Ledger auf.

```
aws qlldb list-journal-kinesis-streams-for-ledger \  
  --ledger-name myExampleLedger
```

Ausgabe:

```
{  
  "Streams": [  
    {  
      "LedgerName": "myExampleLedger",  
      "CreationTime": 1591221984.677,  
      "InclusiveStartTime": 1590710400.0,  
      "ExclusiveEndTime": 1590796799.0,  
      "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",  
      "StreamId": "7ISCKqwe4y25YyHLzYUFaf",  
      "Arn": "arn:aws:qlldb:us-east-1:123456789012:stream/  
myExampleLedger/7ISCKqwe4y25YyHLzYUFaf",  
      "Status": "ACTIVE",  
      "KinesisConfiguration": {  
        "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-  
for-qlldb",  
        "AggregationEnabled": true  
      },  
      "StreamName": "myExampleLedger-stream"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Streaming-Journaldaten von Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [ListJournalKinesisStreamsForLedger](#) in der AWS CLI Befehlsreferenz.

list-journal-s3-exports-for-ledger

Das folgende Codebeispiel zeigt die Verwendung `list-journal-s3-exports-for-ledger`.

AWS CLI

Um Journalexportaufträge für ein Hauptbuch aufzulisten

Im folgenden `list-journal-s3-exports-for-ledger` Beispiel werden Journalexportaufträge für das angegebene Buch aufgeführt.

```
aws qldb list-journal-s3-exports-for-ledger \  
  --name myExampleLedger
```

Ausgabe:

```
{  
  "JournalS3Exports": [  
    {  
      "LedgerName": "myExampleLedger",  
      "ExclusiveEndTime": 1568847599.0,  
      "ExportCreationTime": 1568847801.418,  
      "S3ExportConfiguration": {  
        "Bucket": "awsExampleBucket",  
        "Prefix": "ledgerexport1/",  
        "EncryptionConfiguration": {  
          "ObjectEncryptionType": "SSE_S3"  
        }  
      },  
      "ExportId": "ADR20NPKN5LINYGb4dp7yZ",  
      "RoleArn": "arn:aws:iam::123456789012:role/qldb-s3-export",  
      "InclusiveStartTime": 1568764800.0,  
      "Status": "IN_PROGRESS"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Exportieren Ihres Journals in Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie unter [ListJournalS3 ExportsForLedger](#) in der AWS CLI Befehlsreferenz.

list-journal-s3-exports

Das folgende Codebeispiel zeigt die Verwendung `list-journal-s3-exports`.

AWS CLI

Um Journal-Exportaufträge aufzulisten

Im folgenden `list-journal-s3-exports` Beispiel werden Journalexportaufträge für alle Bücher aufgeführt, die dem AWS Girokonto und der Region zugeordnet sind.

```
aws qlldb list-journal-s3-exports
```

Ausgabe:

```
{
  "JournalS3Exports": [
    {
      "Status": "IN_PROGRESS",
      "LedgerName": "myExampleLedger",
      "S3ExportConfiguration": {
        "EncryptionConfiguration": {
          "ObjectEncryptionType": "SSE_S3"
        },
        "Bucket": "awsExampleBucket",
        "Prefix": "ledgerexport1/"
      },
      "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
      "ExportCreationTime": 1568847801.418,
      "ExportId": "ADR20NPKN5LINYGb4dp7yZ",
      "InclusiveStartTime": 1568764800.0,
      "ExclusiveEndTime": 1568847599.0
    },
    {
      "Status": "COMPLETED",
      "LedgerName": "myExampleLedger2",
```

```
"S3ExportConfiguration": {
  "EncryptionConfiguration": {
    "ObjectEncryptionType": "SSE_S3"
  },
  "Bucket": "awsExampleBucket",
  "Prefix": "ledgerexport1/"
},
"RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
"ExportCreationTime": 1568846847.638,
"ExportId": "2pdvW8UQrjBAiYTMehEJDI",
"InclusiveStartTime": 1568592000.0,
"ExclusiveEndTime": 1568764800.0
}
]
}
```

Weitere Informationen finden Sie unter [Exportieren Ihres Journals in Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie unter [ListJournalS3Exports](#) in der AWS CLI Befehlsreferenz.

list-ledgers

Das folgende Codebeispiel zeigt die Verwendung. `list-ledgers`

AWS CLI

Um Ihre verfügbaren Ledger aufzulisten

Im folgenden `list-ledgers` Beispiel werden alle Bücher aufgeführt, die dem AWS Girokonto und der Region zugeordnet sind.

```
aws qldb list-ledgers
```

Ausgabe:

```
{
  "Ledgers": [
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839243.951,
      "Name": "myExampleLedger"
    }
  ]
}
```

```
    },
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839543.557,
      "Name": "myExampleLedger2"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Basic Operations for Amazon QLDB Ledgers](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [ListLedgers](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die mit einem Ledger verknüpften Tags aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet alle Tags auf, die dem angegebenen Ledger zugeordnet sind.

```
aws qldb list-tags-for-resource \
  --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger
```

Ausgabe:

```
{
  "Tags": {
    "IsTest": "true",
    "Domain": "Test"
  }
}
```

Weitere Informationen finden Sie unter [Tagging Amazon QLDB Resources](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

stream-journal-to-kinesis

Das folgende Codebeispiel zeigt die Verwendung `stream-journal-to-kinesis`.

AWS CLI

Beispiel 1: Um Journaldaten mithilfe von Eingabedateien an Kinesis Data Streams zu streamen

Im folgenden `stream-journal-to-kinesis` Beispiel wird aus einem Hauptbuch mit dem Namen `myExampleLedger` ein Stream von Journaldaten innerhalb eines bestimmten Datums- und Zeitbereichs erstellt. Der Stream sendet die Daten an einen angegebenen Amazon Kinesis Datenstream.

```
aws qlldb stream-journal-to-kinesis \  
  --ledger-name myExampleLedger \  
  --inclusive-start-time 2020-05-29T00:00:00Z \  
  --exclusive-end-time 2020-05-29T23:59:59Z \  
  --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \  
  --kinesis-configuration file://my-kinesis-config.json \  
  --stream-name myExampleLedger-stream
```

Inhalt von `my-kinesis-config.json`:

```
{  
  "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qlldb",  
  "AggregationEnabled": true  
}
```

Ausgabe:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFAf"  
}
```

Weitere Informationen finden Sie unter [Streaming-Journaldaten von Amazon QLDB](#) im Amazon QLDB Developer Guide.

Beispiel 2: Um Journaldaten mithilfe der Kurzsyntax an Kinesis Data Streams zu streamen

Im folgenden `stream-journal-to-kinesis` Beispiel wird aus einem Hauptbuch mit dem Namen ein Stream von Journaldaten innerhalb eines bestimmten Datums- und Zeitbereichs

erstellt. `myExampleLedger` Der Stream sendet die Daten an einen angegebenen Amazon Kinesis Kinesis-Datenstream.

```
aws qlldb stream-journal-to-kinesis \  
  --ledger-name myExampleLedger \  
  --inclusive-start-time 2020-05-29T00:00:00Z \  
  --exclusive-end-time 2020-05-29T23:59:59Z \  
  --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \  
  --stream-name myExampleLedger-stream \  
  --kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/  
stream-for-qlldb,AggregationEnabled=true
```

Ausgabe:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFaf"  
}
```

Weitere Informationen finden Sie unter [Streaming-Journaldaten von Amazon QLDB](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [StreamJournalToKinesis](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Hauptbuch zu taggen

Das folgende `tag-resource` Beispiel fügt einem angegebenen Ledger eine Reihe von Tags hinzu.

```
aws qlldb tag-resource \  
  --resource-arn arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tags IsTest=true,Domain=Test
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Amazon QLDB Resources](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden Tags mit den angegebenen Tagschlüsseln aus einem angegebenen Ledger entfernt.

```
aws qlldb untag-resource \  
  --resource-arn arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tag-keys IsTest Domain
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Amazon QLDB Resources](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-ledger-permissions-mode

Das folgende Codebeispiel zeigt die Verwendung `update-ledger-permissions-mode`.

AWS CLI

Beispiel 1: Um den Berechtigungsmodus eines Ledgers zu aktualisieren STANDARD

Im folgenden `update-ledger-permissions-mode` Beispiel wird dem angegebenen STANDARD Ledger der Berechtigungsmodus zugewiesen.

```
aws qlldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

Ausgabe:

```
{
  "Name": "myExampleLedger",
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "PermissionsMode": "STANDARD"
}
```

Beispiel 2: Um den Berechtigungsmodus eines Ledgers auf `_` zu aktualisieren `ALLOW ALL`

Im folgenden `update-ledger-permissions-mode` Beispiel wird dem angegebenen `ALLOW_ALL` Ledger der Berechtigungsmodus zugewiesen.

```
aws qldb update-ledger-permissions-mode \
  --name myExampleLedger \
  --permissions-mode ALLOW_ALL
```

Ausgabe:

```
{
  "Name": "myExampleLedger",
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "PermissionsMode": "ALLOW_ALL"
}
```

Weitere Informationen finden Sie unter [Basic Operations for Amazon QLDB Ledgers](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [UpdateLedgerPermissionsMode](#) in der AWS CLI Befehlsreferenz.

update-ledger

Das folgende Codebeispiel zeigt die Verwendung `update-ledger`.

AWS CLI

Beispiel 1: Um die Löschschatzeigenschaft eines Ledgers zu aktualisieren

Im folgenden `update-ledger` Beispiel wird das angegebene Ledger aktualisiert, um die Löschschatzfunktion zu deaktivieren.

```
aws qldb update-ledger \
  --name myExampleLedger \
```

--no-deletion-protection

Ausgabe:

```
{
  "CreationDateTime": 1568839243.951,
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "DeletionProtection": false,
  "Name": "myExampleLedger",
  "State": "ACTIVE"
}
```

Beispiel 2: Um den AWS KMS Schlüssel eines Ledgers auf einen vom Kunden verwalteten Schlüssel zu aktualisieren

Im folgenden `update-ledger` Beispiel wird das angegebene Ledger so aktualisiert, dass es einen vom Kunden verwalteten KMS Schlüssel für die Verschlüsselung im Ruhezustand verwendet.

```
aws qldb update-ledger \
  --name myExampleLedger \
  --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "CreationDateTime": 1568839243.951,
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "DeletionProtection": false,
  "Name": "myExampleLedger",
  "State": "ACTIVE",
  "EncryptionDescription": {
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "EncryptionStatus": "UPDATING"
  }
}
```

Beispiel 3: Um den AWS KMS Schlüssel eines Ledgers auf einen AWS eigenen Schlüssel zu aktualisieren

Im folgenden `update-ledger` Beispiel wird das angegebene Ledger so aktualisiert, dass es einen AWS eigenen KMS Schlüssel für die Verschlüsselung im Ruhezustand verwendet.

```
aws qlldb update-ledger \  
  --name myExampleLedger \  
  --kms-key AWS_OWNED_KMS_KEY
```

Ausgabe:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": false,  
  "Name": "myExampleLedger",  
  "State": "ACTIVE",  
  "EncryptionDescription": {  
    "KmsKeyArn": "AWS_OWNED_KMS_KEY",  
    "EncryptionStatus": "UPDATING"  
  }  
}
```

Weitere Informationen finden Sie unter [Basic Operations for Amazon QLDB Ledgers](#) im Amazon QLDB Developer Guide.

- API-Einheiten finden Sie [UpdateLedger](#) in der AWS CLI Befehlsreferenz.

RDS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren RDS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-option-to-option-group

Das folgende Codebeispiel zeigt die Verwendung `add-option-to-option-group`.

AWS CLI

Um eine Option zu einer Optionsgruppe hinzuzufügen

Das folgende `add-option-to-option-group` Beispiel fügt der angegebenen Optionsgruppe eine Option hinzu.

```
aws rds add-option-to-option-group \  
  --option-group-name myoptiongroup \  
  --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \  
  --apply-immediately
```

Ausgabe:

```
{  
  "OptionGroup": {  
    "OptionGroupName": "myoptiongroup",  
    "OptionGroupDescription": "Test Option Group",  
    "EngineName": "oracle-ee",  
    "MajorEngineVersion": "12.1",  
    "Options": [  
      {  
        "OptionName": "Timezone",  
        "OptionDescription": "Change time zone",  
        "Persistent": true,  
        "Permanent": false,  
        "OptionSettings": [  
          {  
            "Name": "TIME_ZONE",  
            "Value": "Australia/Sydney",  
            "DefaultValue": "UTC",  
            "Description": "Specifies the timezone the user wants to  
change the system time to",  
            "ApplyType": "DYNAMIC",  
            "DataType": "STRING",  
            "AllowedValues": "Africa/Cairo,Africa/Casablanca,Africa/  
Harare,Africa/Lagos,Africa/Luanda,Africa/Monrovia,Africa/Nairobi,Africa/
```

```

Tripoli,Africa/Windhoek,America/Araguaina,America/Argentina/Buenos_Aires,America/
Asuncion,America/Bogota,America/Caracas,America/Chicago,America/Chihuahua,America/
Cuiaba,America/Denver,America/Detroit,America/Fortaleza,America/Godthab,America/
Guatemala,America/Halifax,America/Lima,America/Los_Angeles,America/Manaus,America/
Matamoros,America/Mexico_City,America/Monterrey,America/Montevideo,America/
New_York,America/Phoenix,America/Santiago,America/Sao_Paulo,America/Tijuana,America/
Toronto,Asia/Amman,Asia/Ashgabat,Asia/Baghdad,Asia/Baku,Asia/Bangkok,Asia/
Beirut,Asia/Calcutta,Asia/Damascus,Asia/Dhaka,Asia/Hong_Kong,Asia/Irkutsk,Asia/
Jakarta,Asia/Jerusalem,Asia/Kabul,Asia/Karachi,Asia/Kathmandu,Asia/Kolkata,Asia/
Krasnoyarsk,Asia/Magadan,Asia/Manila,Asia/Muscat,Asia/Novosibirsk,Asia/Rangoon,Asia/
Riyadh,Asia/Seoul,Asia/Shanghai,Asia/Singapore,Asia/Taipei,Asia/Tehran,Asia/
Tokyo,Asia/Ulaanbaatar,Asia/Vladivostok,Asia/Yakutsk,Asia/Yerevan,Atlantic/
Azores,Atlantic/Cape_Verde,Australia/Adelaide,Australia/Brisbane,Australia/
Darwin,Australia/Eucla,Australia/Hobart,Australia/Lord_Howe,Australia/
Perth,Australia/Sydney,Brazil/DeNoronha,Brazil/East,Canada/Newfoundland,Canada/
Saskatchewan,Etc/GMT-3,Europe/Amsterdam,Europe/Athens,Europe/Berlin,Europe/
Dublin,Europe/Helsinki,Europe/Kaliningrad,Europe/London,Europe/Madrid,Europe/
Moscow,Europe/Paris,Europe/Prague,Europe/Rome,Europe/Sarajevo,Pacific/Apia,Pacific/
Auckland,Pacific/Chatham,Pacific/Fiji,Pacific/Guam,Pacific/Honolulu,Pacific/
Kiritimati,Pacific/Marquesas,Pacific/Samoa,Pacific/Tongatapu,Pacific/Wake,US/
Alaska,US/Central,US/East-Indiana,US/Eastern,US/Pacific,UTC",
        "IsModifiable": true,
        "IsCollection": false
    }
],
"DBSecurityGroupMemberships": [],
"VpcSecurityGroupMemberships": []
},
{
    "OptionName": "OEM",
    "OptionDescription": "Oracle 12c EM Express",
    "Persistent": false,
    "Permanent": false,
    "Port": 5500,
    "OptionSettings": [],
    "DBSecurityGroupMemberships": [
        {
            "DBSecurityGroupName": "default",
            "Status": "authorized"
        }
    ],
    "VpcSecurityGroupMemberships": []
}
],

```

```
"AllowsVpcAndNonVpcInstanceMemberships": false,  
"OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"  
}  
}
```

Weitere Informationen finden Sie unter [Hinzufügen einer Option zu einer Optionsgruppe](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AddOptionToOptionGroup](#) in der AWS CLI Befehlsreferenz.

add-role-to-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `add-role-to-db-cluster`.

AWS CLI

So ordnen Sie einem DB-Cluster eine AWS Identity and Access Management (IAM) -Rolle zu

Im folgenden `add-role-to-db-cluster` Beispiel wird eine Rolle einem DB-Cluster zugeordnet.

```
aws rds add-role-to-db-cluster \  
  --db-cluster-identifizier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Zuordnen einer IAM Rolle zu einem Amazon Aurora My SQL DB-Cluster](#) im Amazon Aurora Benutzerhandbuch.

- APIEinzelheiten finden Sie [AddRoleToDbCluster](#) in der AWS CLI Befehlsreferenz.

add-role-to-db-instance

Das folgende Codebeispiel zeigt die Verwendung `add-role-to-db-instance`.

AWS CLI

So ordnen Sie einer DB-Instance eine AWS Identity and Access Management (IAM) -Rolle zu

Im folgenden `add-role-to-db-instance` Beispiel wird die Rolle einer Oracle-DB-Instance mit dem Namen `test-instance` hinzugefügt.

```
aws rds add-role-to-db-instance \  
  --db-instance-identifizier test-instance \  
  --feature-name S3_INTEGRATION \  
  --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Voraussetzungen für die Amazon RDS Oracle-Integration mit Amazon S3](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AddRoleToDbInstance](#) unter AWS CLI Befehlsreferenz.

add-source-identifizier-to-subscription

Das folgende Codebeispiel zeigt die Verwendung `add-source-identifizier-to-subscription`.

AWS CLI

Um einem Abonnement eine Quell-ID hinzuzufügen

Im folgenden `add-source-identifizier` Beispiel wird einem vorhandenen Abonnement eine weitere Quell-ID hinzugefügt.

```
aws rds add-source-identifizier-to-subscription \  
  --subscription-name my-instance-events \  
  --source-identifizier test-instance-repl
```

Ausgabe:

```
{  
  "EventSubscription": {  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "CustSubscriptionId": "my-instance-events",  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-  
events",  
    "Enabled": false,  
    "Status": "modifying",  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ],  
  },  
}
```



```

    "CustomerAwsId": "123456789012",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
    "SourceType": "db-instance",
    "SourceIdsList": [
        "test-instance",
        "test-instance-repl"
    ]
}
}

```

- API-Einheiten finden Sie [AddSourceIdentifierToSubscription](#) unter AWS CLI Befehlsreferenz.

add-tags-to-resource

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-resource`.

AWS CLI

Um einer Ressource Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel werden einer RDS Datenbank Tags hinzugefügt.

```

aws rds add-tags-to-resource \
  --resource-name arn:aws:rds:us-east-1:123456789012:db:database-mysql \
  --tags "[{\"Key\": \"Name\", \"Value\": \"MyDatabase\"}, {\"Key\": \"Environment\", \"Value\": \"test\"}]"

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Amazon RDS Resources](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [AddTagsToResource](#) in der AWS CLI Befehlsreferenz.

apply-pending-maintenance-action

Das folgende Codebeispiel zeigt die Verwendung `apply-pending-maintenance-action`.

AWS CLI

So führen Sie ausstehende Wartungsaktionen durch

Im folgenden `apply-pending-maintenance-action` Beispiel werden die ausstehenden Wartungsaktionen für einen DB-Cluster angewendet.

```
aws rds apply-pending-maintenance-action \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:my-db-cluster \  
  --apply-action system-update \  
  --opt-in-type immediate
```

Ausgabe:

```
{  
  "ResourcePendingMaintenanceActions": {  
    "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:cluster:my-db-  
cluster",  
    "PendingMaintenanceActionDetails": [  
      {  
        "Action": "system-update",  
        "OptInStatus": "immediate",  
        "CurrentApplyDate": "2021-01-23T01:07:36.100Z",  
        "Description": "Upgrade to Aurora PostgreSQL 3.3.2"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Wartung einer DB-Instance](#) im RDSAmazon-Benutzerhandbuch und [Wartung eines Amazon Aurora DB-Clusters](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [ApplyPendingMaintenanceAction](#) unter AWS CLI Befehlsreferenz.

authorize-db-security-group-ingress

Das folgende Codebeispiel zeigt die Verwendung `authorize-db-security-group-ingress`.

AWS CLI

So ordnen Sie einer DB-Instance eine AWS Identity and Access Management (IAM) -Rolle zu

Im folgenden `authorize-db-security-group-ingress` Beispiel wird die Standardsicherheitsgruppe mit einer Eingangsregel für den CIDR IP-Bereich 192.0.2.0/24 konfiguriert.

```
aws rds authorize-db-security-group-ingress \
  --db-security-group-name default \
  --cidrip 192.0.2.0/24
```

Ausgabe:

```
{
  "DBSecurityGroup": {
    "OwnerId": "123456789012",
    "DBSecurityGroupName": "default",
    "DBSecurityGroupDescription": "default",
    "EC2SecurityGroups": [],
    "IPRanges": [
      {
        "Status": "authorizing",
        "CIDRIP": "192.0.2.0/24"
      }
    ],
    "DBSecurityGroupArn": "arn:aws:rds:us-east-1:111122223333:secgrp:default"
  }
}
```

Weitere Informationen finden Sie unter [Autorisieren des Netzwerkzugriffs auf eine DB-Sicherheitsgruppe von einem IP-Bereich](#) aus im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AuthorizeDbSecurityGroupIngress](#) in der AWS CLI Befehlsreferenz.

backtrack-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `backtrack-db-cluster`.

AWS CLI

Um einen Aurora-DB-Cluster zurückzuverfolgen

Im folgenden `backtrack-db-cluster` Beispiel wird der angegebene DB-Cluster-Beispielcluster auf den 19. März 2018 um 10 Uhr zurückgesetzt.

```
aws rds backtrack-db-cluster --db-cluster-identifizier sample-cluster --backtrack-
to 2018-03-19T10:00:00+00:00
```

Dieser Befehl gibt einen JSON Block aus, der die Änderung an der Ressource bestätigt. RDS

- API-Einheiten finden Sie [BacktrackDbCluster](#) in der AWS CLI Befehlsreferenz.

cancel-export-task

Das folgende Codebeispiel zeigt die Verwendung `cancel-export-task`.

AWS CLI

Um einen Snapshot-Export nach Amazon S3 abzubrechen

Im folgenden `cancel-export-task` Beispiel wird eine laufende Exportaufgabe abgebrochen, bei der ein Snapshot nach Amazon S3 exportiert wird.

```
aws rds cancel-export-task \  
  --export-task-identifier my-s3-export-1
```

Ausgabe:

```
{  
  "ExportTaskIdentifier": "my-s3-export-1",  
  "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:publisher-final-snapshot",  
  "SnapshotTime": "2019-03-24T20:01:09.815Z",  
  "S3Bucket": "mybucket",  
  "S3Prefix": "",  
  "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/export-snap-S3-role",  
  "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcd0000-7bfd-4594-af38-aabbccddeeff",  
  "Status": "CANCELING",  
  "PercentProgress": 0,  
  "TotalExtractedDataInGB": 0  
}
```

Weitere Informationen finden Sie unter [Abbrechen einer Snapshot-Exportaufgabe](#) im RDS Amazon-Benutzerhandbuch oder [Abbrechen einer Snapshot-Exportaufgabe](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [CancelExportTask](#) in der AWS CLI Befehlsreferenz.

copy-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `copy-db-cluster-parameter-group`.

AWS CLI

Um eine DB-Cluster-Parametergruppe zu kopieren

Im folgenden `copy-db-cluster-parameter-group` Beispiel wird eine Kopie einer DB-Cluster-Parametergruppe erstellt.

```
aws rds copy-db-cluster-parameter-group \  
  --source-db-cluster-parameter-group-identifier mydbclusterpg \  
  --target-db-cluster-parameter-group-identifier mydbclusterpgcopy \  
  --target-db-cluster-parameter-group-description "Copy of mydbclusterpg parameter  
group"
```

Ausgabe:

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "mydbclusterpgcopy",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
pg:mydbclusterpgcopy",  
    "DBParameterGroupFamily": "aurora-mysql5.7",  
    "Description": "Copy of mydbclusterpg parameter group"  
  }  
}
```

Weitere Informationen finden Sie unter [Kopieren einer DB-Cluster-Parametergruppe](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [CopyDbClusterParameterGroup](#) unter AWS CLI Befehlsreferenz.

copy-db-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-db-cluster-snapshot`.

AWS CLI

Um einen DB-Cluster-Snapshot zu kopieren

Im folgenden `copy-db-cluster-snapshot` Beispiel wird eine Kopie eines DB-Cluster-Snapshots einschließlich seiner Tags erstellt.

```
aws rds copy-db-cluster-snapshot \  

```

```

--source-db-cluster-snapshot-identifizier arn:aws:rds:us-east-1:123456789012:cluster-snapshot:rds:myaurora-2019-06-04-09-16
--target-db-cluster-snapshot-identifizier myclustersnapshotcopy \
--copy-tags

```

Ausgabe:

```

{
  "DBClusterSnapshot": {
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1e"
    ],
    "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
    "DBClusterIdentifier": "myaurora",
    "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
    "Engine": "aurora-mysql",
    "AllocatedStorage": 0,
    "Status": "available",
    "Port": 0,
    "VpcId": "vpc-6594f31c",
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
    "MasterUsername": "myadmin",
    "EngineVersion": "5.7.mysql_aurora.2.04.2",
    "LicenseModel": "aurora-mysql",
    "SnapshotType": "manual",
    "PercentProgress": 100,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-snapshot:myclustersnapshotcopy",
    "IAMDatabaseAuthenticationEnabled": false
  }
}

```

Weitere Informationen finden Sie unter [Kopieren eines Snapshots](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [CopyDbClusterSnapshot](#) unter AWS CLI Befehlsreferenz.

copy-db-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `copy-db-parameter-group`.

AWS CLI

Um eine DB-Cluster-Parametergruppe zu kopieren

Im folgenden `copy-db-parameter-group` Beispiel wird eine Kopie einer DB-Parametergruppe erstellt.

```
aws rds copy-db-parameter-group \  
  --source-db-parameter-group-identifier mydbpg \  
  --target-db-parameter-group-identifier mydbpgcopy \  
  --target-db-parameter-group-description "Copy of mydbpg parameter group"
```

Ausgabe:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbpgcopy",  
    "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",  
    "DBParameterGroupFamily": "mysql5.7",  
    "Description": "Copy of mydbpg parameter group"  
  }  
}
```

Weitere Informationen finden Sie unter [Kopieren einer DB-Parametergruppe](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CopyDbParameterGroup](#) unter AWS CLI Befehlsreferenz.

copy-db-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-db-snapshot`.

AWS CLI

Um einen DB-Snapshot zu kopieren

Im folgenden `copy-db-snapshot` Beispiel wird eine Kopie eines DB-Snapshots erstellt.

```
aws rds copy-db-snapshot \
  --source-db-snapshot-identifizier rds:database-mysql-2019-06-06-08-38
  --target-db-snapshot-identifizier mydbsnapshotcopy
```

Ausgabe:

```
{
  "DBSnapshot": {
    "VpcId": "vpc-6594f31c",
    "Status": "creating",
    "Encrypted": true,
    "SourceDBSnapshotIdentifizier": "arn:aws:rds:us-
east-1:123456789012:snapshot:rds:database-mysql-2019-06-06-08-38",
    "MasterUsername": "admin",
    "Iops": 1000,
    "Port": 3306,
    "LicenseModel": "general-public-license",
    "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshotcopy",
    "EngineVersion": "5.6.40",
    "OptionGroupName": "default:mysql-5-6",
    "ProcessorFeatures": [],
    "Engine": "mysql",
    "StorageType": "io1",
    "DbiResourceId": "db-ZI7UJ5BLKMBYFGX7FDENCKADC4",
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
    "SnapshotType": "manual",
    "IAMDatabaseAuthenticationEnabled": false,
    "SourceRegion": "us-east-1",
    "DBInstanceIdentifizier": "database-mysql",
    "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
    "AvailabilityZone": "us-east-1f",
    "PercentProgress": 0,
    "AllocatedStorage": 100,
    "DBSnapshotIdentifizier": "mydbsnapshotcopy"
  }
}
```

Weitere Informationen finden Sie unter [Kopieren eines Snapshots](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CopyDbSnapshot](#) in der AWS CLI Befehlsreferenz.

copy-option-group

Das folgende Codebeispiel zeigt die Verwendung `copy-option-group`.

AWS CLI

Um eine Optionsgruppe zu kopieren

Im folgenden `copy-option-group` Beispiel wird eine Kopie einer Optionsgruppe erstellt.

```
aws rds copy-option-group \  
  --source-option-group-identifizier myoptiongroup \  
  --target-option-group-identifizier new-option-group \  
  --target-option-group-description "My option group copy"
```

Ausgabe:

```
{  
  "OptionGroup": {  
    "Options": [],  
    "OptionGroupName": "new-option-group",  
    "MajorEngineVersion": "11.2",  
    "OptionGroupDescription": "My option group copy",  
    "AllowsVpcAndNonVpcInstanceMemberships": true,  
    "EngineName": "oracle-ee",  
    "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:new-option-group"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer Kopie einer Optionsgruppe](#) im RDS Amazon-Benutzerhandbuch.

- API Einheiten finden Sie [CopyOptionGroup](#) unter AWS CLI Befehlsreferenz.

create-blue-green-deployment

Das folgende Codebeispiel zeigt die Verwendung `create-blue-green-deployment`.

AWS CLI

Beispiel 1: Um eine blaue/grüne Bereitstellung für eine RDS for My SQL DB-Instance zu erstellen

Das folgende `create-blue-green-deployment` Beispiel erstellt eine blaue/grüne Bereitstellung für eine My SQL DB-Instance.

```
aws rds create-blue-green-deployment \  
  --blue-green-deployment-name bgd-cli-test-instance \  
  --source arn:aws:rds:us-east-1:123456789012:db:my-db-instance \  
  --target-engine-version 8.0 \  
  --target-db-parameter-group-name mysql-80-group
```

Ausgabe:

```
{  
  "BlueGreenDeployment": {  
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",  
    "BlueGreenDeploymentName": "bgd-cli-test-instance",  
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",  
    "SwitchoverDetails": [  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-1"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-2"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-3"  
      }  
    ],  
    "Tasks": [  
      {  
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",  
        "Status": "PENDING"  
      },  
      {  
        "Name": "DB_ENGINE_VERSION_UPGRADE",  
        "Status": "PENDING"  
      }  
    ]  
  }  
}
```

```

    {
      "Name": "CONFIGURE_BACKUPS",
      "Status": "PENDING"
    },
    {
      "Name": "CREATING_TOPOLOGY_OF_SOURCE",
      "Status": "PENDING"
    }
  ],
  "Status": "PROVISIONING",
  "CreateTime": "2022-02-25T21:18:51.183000+00:00"
}

```

Weitere Informationen finden Sie unter [Erstellen einer blauen/grünen Bereitstellung](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: So erstellen Sie eine blaue/grüne Bereitstellung für einen Aurora My SQL DB-Cluster

Das folgende create-blue-green-deployment Beispiel erstellt eine blaue/grüne Bereitstellung für einen Aurora My SQL DB-Cluster.

```

aws rds create-blue-green-deployment \
  --blue-green-deployment-name my-blue-green-deployment \
  --source arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster \
  --target-engine-version 8.0 \
  --target-db-cluster-parameter-group-name ams-80-binlog-enabled \
  --target-db-parameter-group-name mysql-80-cluster-group

```

Ausgabe:

```

{
  "BlueGreenDeployment": {
    "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
    "BlueGreenDeploymentName": "my-blue-green-deployment",
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
        "Status": "PROVISIONING"
      }
    ]
  }
}

```

```
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-1",
      "Status": "PROVISIONING"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-2",
      "Status": "PROVISIONING"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-3",
      "Status": "PROVISIONING"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
excluded-member-endpoint",
      "Status": "PROVISIONING"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
reader-endpoint",
      "Status": "PROVISIONING"
    }
  ],
  "Tasks": [
    {
      "Name": "CREATING_READ_REPLICA_OF_SOURCE",
      "Status": "PENDING"
    },
    {
      "Name": "DB_ENGINE_VERSION_UPGRADE",
      "Status": "PENDING"
    },
    {
      "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
      "Status": "PENDING"
    },
    {
      "Name": "CREATE_CUSTOM_ENDPOINTS",
      "Status": "PENDING"
    }
  ]
],
```

```

    "Status": "PROVISIONING",
    "CreateTime": "2022-02-25T21:12:00.288000+00:00"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen einer blauen/grünen Bereitstellung](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateBlueGreenDeployment AWS CLI Befehlsreferenz](#).

create-db-cluster-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster-endpoint`.

AWS CLI

Um einen benutzerdefinierten DB-Cluster-Endpoint zu erstellen

Im folgenden `create-db-cluster-endpoint` Beispiel wird ein benutzerdefinierter DB-Cluster-Endpoint erstellt und mit dem angegebenen Aurora-DB-Cluster verknüpft.

```

aws rds create-db-cluster-endpoint \
  --db-cluster-endpoint-identifier mycustomendpoint \
  --endpoint-type reader \
  --db-cluster-identifier mydbcluster \
  --static-members dbinstance1 dbinstance2

```

Ausgabe:

```

{
  "DBClusterEndpointIdentifier": "mycustomendpoint",
  "DBClusterIdentifier": "mydbcluster",
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
  "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-east-1.rds.amazonaws.com",
  "Status": "creating",
  "EndpointType": "CUSTOM",
  "CustomEndpointType": "READER",
  "StaticMembers": [
    "dbinstance1",
    "dbinstance2"
  ],
}

```

```
"ExcludedMembers": [],
  "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

Weitere Informationen finden Sie unter [Amazon Aurora Connection Management](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDbClusterEndpoint](#) unter AWS CLI Befehlsreferenz.

create-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster-parameter-group`.

AWS CLI

Um eine DB-Cluster-Parametergruppe zu erstellen

Im folgenden `create-db-cluster-parameter-group` Beispiel wird eine DB-Cluster-Parametergruppe erstellt.

```
aws rds create-db-cluster-parameter-group \
  --db-cluster-parameter-group-name mydbclusterparametergroup \
  --db-parameter-group-family aurora5.6 \
  --description "My new cluster parameter group"
```

Ausgabe:

```
{
  "DBClusterParameterGroup": {
    "DBClusterParameterGroupName": "mydbclusterparametergroup",
    "DBParameterGroupFamily": "aurora5.6",
    "Description": "My new cluster parameter group",
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen einer DB-Cluster-Parametergruppe](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDbClusterParameterGroup](#) unter AWS CLI Befehlsreferenz.

create-db-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster-snapshot`.

AWS CLI

Um einen DB-Cluster-Snapshot zu erstellen

Im folgenden `create-db-cluster-snapshot` Beispiel wird ein DB-Cluster-Snapshot erstellt.

```
aws rds create-db-cluster-snapshot \  
  --db-cluster-identifizier mydbcluster \  
  --db-cluster-snapshot-identifizier mydbclustersnapshot
```

Ausgabe:

```
{  
  "DBClusterSnapshot": {  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1e"  
    ],  
    "DBClusterSnapshotIdentifizier": "mydbclustersnapshot",  
    "DBClusterIdentifizier": "mydbcluster",  
    "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",  
    "Engine": "aurora-mysql",  
    "AllocatedStorage": 1,  
    "Status": "creating",  
    "Port": 0,  
    "VpcId": "vpc-6594f31c",  
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",  
    "MasterUsername": "myadmin",  
    "EngineVersion": "5.7.mysql_aurora.2.04.2",  
    "LicenseModel": "aurora-mysql",  
    "SnapshotType": "manual",  
    "PercentProgress": 0,  
    "StorageEncrypted": true,  
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",  
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
snapshot:mydbclustersnapshot",  
    "IAMDatabaseAuthenticationEnabled": false  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Erstellen eines DB-Cluster-Snapshots](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDbClusterSnapshot](#) unter AWS CLI Befehlsreferenz.

create-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-db-cluster`.

AWS CLI

Beispiel 1: So erstellen Sie einen My SQL 5.7-kompatiblen DB-Cluster

Im folgenden `create-db-cluster` Beispiel wird ein My SQL 5.7-kompatibler DB-Cluster unter Verwendung der Standard-Engine-Version erstellt. Ersetzen Sie das Beispielkennwort `secret99` durch ein sicheres Passwort. Wenn Sie die Konsole verwenden, um einen DB-Cluster zu erstellen, erstellt Amazon RDS automatisch die Writer-DB-Instance für Ihren DB-Cluster. Wenn Sie jedoch den verwenden, AWS CLI um einen DB-Cluster zu erstellen, müssen Sie die Writer-DB-Instance für Ihren DB-Cluster mit dem `create-db-instance` AWS CLI Befehl explizit erstellen.

```
aws rds create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --engine aurora-mysql \  
  --engine-version 5.7 \  
  --master-username admin \  
  --master-user-password secret99 \  
  --db-subnet-group-name default \  
  --vpc-security-group-ids sg-0b9130572daf3dc16
```

Ausgabe:

```
{  
  "DBCluster": {  
    "DBSubnetGroup": "default",  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-0b9130572daf3dc16",  
        "Status": "active"  
      }  
    ]  
  }  
}
```



```

    ],
    "AllocatedStorage": 1,
    "AssociatedRoles": [],
    "PreferredBackupWindow": "09:12-09:42",
    "ClusterCreateTime": "2023-02-27T23:21:33.048Z",
    "DeletionProtection": false,
    "IAMDatabaseAuthenticationEnabled": false,
    "ReadReplicaIdentifiers": [],
    "EngineMode": "provisioned",
    "Engine": "aurora-mysql",
    "StorageEncrypted": false,
    "MultiAZ": false,
    "PreferredMaintenanceWindow": "mon:04:31-mon:05:01",
    "HttpEndpointEnabled": false,
    "BackupRetentionPeriod": 1,
    "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
    "DBClusterIdentifier": "sample-cluster",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1e"
    ],
    ],
    "MasterUsername": "master",
    "EngineVersion": "5.7.mysql_aurora.2.11.1",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
    "DBClusterMembers": [],
    "Port": 3306,
    "Status": "creating",
    "Endpoint": "sample-cluster.cluster-cnpxexample.us-east-1.rds.amazonaws.com",
    "DBClusterParameterGroup": "default.aurora-mysql5.7",
    "HostedZoneId": "Z2R2ITUGPM61AM",
    "ReaderEndpoint": "sample-cluster.cluster-ro-cnpxexample.us-
east-1.rds.amazonaws.com",
    "CopyTagsToSnapshot": false
  }
}

```

Beispiel 2: Um einen Postgre SQL --kompatiblen DB-Cluster zu erstellen

Das folgende `create-db-cluster` Beispiel erstellt einen Postgre SQL -kompatiblen DB-Cluster unter Verwendung der Standard-Engine-Version. Ersetzen Sie das Beispielkennwort `secret99` durch ein sicheres Passwort. Wenn Sie die Konsole verwenden, um einen DB-Cluster zu erstellen, erstellt Amazon RDS automatisch die Writer-DB-Instance für Ihren DB-Cluster.

Wenn Sie jedoch den verwenden, AWS CLI um einen DB-Cluster zu erstellen, müssen Sie die Writer-DB-Instance für Ihren DB-Cluster mit dem `create-db-instance` AWS CLI Befehl explizit erstellen.

```
aws rds create-db-cluster \  
  --db-cluster-identifier sample-pg-cluster \  
  --engine aurora-postgresql \  
  --master-username master \  
  --master-user-password secret99 \  
  --db-subnet-group-name default \  
  --vpc-security-group-ids sg-0b9130572daf3dc16
```

Ausgabe:

```
{  
  "DBCluster": {  
    "Endpoint": "sample-pg-cluster.cluster-cnpxample.us-east-1.rds.amazonaws.com",  
    "HttpEndpointEnabled": false,  
    "DBClusterMembers": [],  
    "EngineMode": "provisioned",  
    "CopyTagsToSnapshot": false,  
    "HostedZoneId": "Z2R2ITUGPM61AM",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "AllocatedStorage": 1,  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-0b9130572daf3dc16",  
        "Status": "active"  
      }  
    ],  
    "DeletionProtection": false,  
    "StorageEncrypted": false,  
    "BackupRetentionPeriod": 1,  
    "PreferredBackupWindow": "09:56-10:26",  
    "ClusterCreateTime": "2023-02-27T23:26:08.371Z",  
    "DBClusterParameterGroup": "default.aurora-postgresql13",  
    "EngineVersion": "13.7",  
    "Engine": "aurora-postgresql",  
    "Status": "creating",  
    "DBClusterIdentifier": "sample-pg-cluster",  
    "MultiAZ": false,  
    "Port": 5432,  
  }  
}
```

```

    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-pg-
cluster",
    "AssociatedRoles": [],
    "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
    "PreferredMaintenanceWindow": "wed:03:33-wed:04:03",
    "ReaderEndpoint": "sample-pg-cluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
    "MasterUsername": "master",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c"
    ],
    "ReadReplicaIdentifiers": [],
    "DBSubnetGroup": "default"
}
}

```

Weitere Informationen finden Sie unter [Erstellen eines Amazon Aurora Aurora-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDbCluster](#) unter AWS CLI Befehlsreferenz.

create-db-instance-read-replica

Das folgende Codebeispiel zeigt die Verwendung `create-db-instance-read-replica`.

AWS CLI

Um eine DB-Instance zu erstellen, lesen Sie [Replica](#)

In diesem Beispiel wird eine Read Replica einer vorhandenen DB-Instance mit dem Namen `test-instance` erstellt. Die Read Replica trägt den Namen `test-instance-repl`.

```

aws rds create-db-instance-read-replica \
  --db-instance-identifier test-instance-repl \
  --source-db-instance-identifier test-instance

```

Ausgabe:

```

{
  "DBInstance": {
    "IAMDatabaseAuthenticationEnabled": false,

```

```

    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
    "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
    "DBInstanceIdentifier": "test-instance-repl",
    ...some output truncated...
  }
}

```

- API-Einheiten finden Sie [CreateDbInstanceReadReplica](#) in der AWS CLI Befehlsreferenz.

create-db-instance

Das folgende Codebeispiel zeigt die Verwendung `create-db-instance`.

AWS CLI

Um eine DB-Instance zu erstellen

Im folgenden `create-db-instance` Beispiel werden die erforderlichen Optionen verwendet, um eine neue DB-Instance zu starten.

```

aws rds create-db-instance \
  --db-instance-identifier test-mysql-instance \
  --db-instance-class db.t3.micro \
  --engine mysql \
  --master-username admin \
  --master-user-password secret99 \
  --allocated-storage 20

```

Ausgabe:

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "test-mysql-instance",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "creating",
    "MasterUsername": "admin",
    "AllocatedStorage": 20,
    "PreferredBackupWindow": "12:55-13:25",
    "BackupRetentionPeriod": 1,
    "DBSecurityGroups": [],

```

```
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-12345abc",
    "Status": "active"
  }
],
"DBParameterGroups": [
  {
    "DBParameterGroupName": "default.mysql5.7",
    "ParameterApplyStatus": "in-sync"
  }
],
"DBSubnetGroup": {
  "DBSubnetGroupName": "default",
  "DBSubnetGroupDescription": "default",
  "VpcId": "vpc-2ff2ff2f",
  "SubnetGroupStatus": "Complete",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      },
    },
  ]
}
```

```

        "SubnetStatus": "Active"
      }
    ]
  },
  "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
  "PendingModifiedValues": {
    "MasterUserPassword": "*****"
  },
  "MultiAZ": false,
  "EngineVersion": "5.7.22",
  "AutoMinorVersionUpgrade": true,
  "ReadReplicaDBInstanceIdentifiers": [],
  "LicenseModel": "general-public-license",
  "OptionGroupMemberships": [
    {
      "OptionGroupName": "default:mysql-5-7",
      "Status": "in-sync"
    }
  ],
  "PubliclyAccessible": true,
  "StorageType": "gp2",
  "DbInstancePort": 0,
  "StorageEncrypted": false,
  "DbiResourceId": "db-5555EXAMPLE444444444EXAMPLE",
  "CACertificateIdentifier": "rds-ca-2019",
  "DomainMemberships": [],
  "CopyTagsToSnapshot": false,
  "MonitoringInterval": 0,
  "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
  "IAMDatabaseAuthenticationEnabled": false,
  "PerformanceInsightsEnabled": false,
  "DeletionProtection": false,
  "AssociatedRoles": []
}
}

```

Weitere Informationen finden Sie unter [Erstellen einer Amazon RDS DB-Instance](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [C createDBInstance](#) in der AWS CLI Befehlsreferenz.

create-db-parameter-group

Das folgende Codebeispiel zeigt, wie man `create-db-parameter-group`.

AWS CLI

Um eine DB-Parametergruppe zu erstellen

Im folgenden `create-db-parameter-group` Beispiel wird eine DB-Parametergruppe erstellt.

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --db-parameter-group-family MySQL5.6 \  
  --description "My new parameter group"
```

Ausgabe:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbparametergroup",  
    "DBParameterGroupFamily": "mysql5.6",  
    "Description": "My new parameter group",  
    "DBParameterGroupArn": "arn:aws:rds:us-  
east-1:123456789012:pg:mydbparametergroup"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer DB-Parametergruppe](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [reateDBParameterC-Gruppe](#) in der AWS CLI Befehlsreferenz.

create-db-proxy-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-db-proxy-endpoint`.

AWS CLI

Um einen DB-Proxyendpunkt für eine RDS Datenbank zu erstellen

Im folgenden `create-db-proxy-endpoint` Beispiel wird ein DB-Proxyendpunkt erstellt.

```
aws rds create-db-proxy-endpoint \  

```

```
--db-proxy-name proxyExample \  
--db-proxy-endpoint-name "proxyep1" \  
--vpc-subnet-ids subnetgroup1 subnetgroup2
```

Ausgabe:

```
{  
  "DBProxyEndpoint": {  
    "DBProxyEndpointName": "proxyep1",  
    "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-  
endpoint:prx-endpoint-0123a01b12345c0ab",  
    "DBProxyName": "proxyExample",  
    "Status": "creating",  
    "VpcId": "vpc-1234567",  
    "VpcSecurityGroupIds": [  
      "sg-1234",  
      "sg-5678"  
    ],  
    "VpcSubnetIds": [  
      "subnetgroup1",  
      "subnetgroup2"  
    ],  
    "Endpoint": "proxyep1.endpoint.proxy-ab0cd1efghij.us-  
east-1.rds.amazonaws.com",  
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",  
    "TargetRole": "READ_WRITE",  
    "IsDefault": false  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Proxy-Endpunkts](#) im RDSAmazon-Benutzerhandbuch und [Erstellen eines Proxy-Endpunkts](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDbProxyEndpoint](#) in der AWS CLI Befehlsreferenz.

create-db-proxy

Das folgende Codebeispiel zeigt die Verwendung `create-db-proxy`.

AWS CLI

Um einen DB-Proxy für eine RDS Datenbank zu erstellen

Das folgende `create-db-proxy` Beispiel erstellt einen DB-Proxy.

```
aws rds create-db-proxy \
  --db-proxy-name proxyExample \
  --engine-family MYSQL \
  --auth
  Description="proxydescription1",AuthScheme="SECRETS",SecretArn="arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",IAMAuth="DISABLED",ClientPasswordAuthType="MYSO
\
  --role-arn arn:aws:iam::123456789123:role/ProxyRole \
  --vpc-subnet-ids subnetgroup1 subnetgroup2
```

Ausgabe:

```
{
  "DBProxy": {
    "DBProxyName": "proxyExample",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
    "EngineFamily": "MYSQL",
    "VpcId": "vpc-1234567",
    "VpcSecuritytGroupIds": [
      "sg-1234",
      "sg-5678",
      "sg-9101"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
        "Description": "proxydescription1",
        "AuthScheme": "SECRETS",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
        "IAMAuth": "DISABLED"
      }
    ]",
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyRole",
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "RequireTLS": false,
    "IdleClientTimeout": 1800,
```

```
    "DebuggingLogging": false,  
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",  
    "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen eines RDS Proxys](#) im RDSAmazon-Benutzerhandbuch und [Erstellen eines RDS Proxys](#) im Amazon Aurora-Benutzerhandbuch.

- APIEinzelheiten finden Sie [CreateDbProxy](#) in der AWS CLI Befehlsreferenz.

create-db-security-group

Das folgende Codebeispiel zeigt die Verwendung `create-db-security-group`.

AWS CLI

So erstellen Sie eine Amazon RDS DB-Sicherheitsgruppe

Der folgende `create-db-security-group` Befehl erstellt eine neue Amazon RDS DB-Sicherheitsgruppe:

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-group-description "My Test Security Group"
```

Im Beispiel hat die neue DB-Sicherheitsgruppe einen Namen `mysecgroup` und eine Beschreibung.

Ausgabe:

```
{  
  "DBSecurityGroup": {  
    "OwnerId": "123456789012",  
    "DBSecurityGroupName": "mysecgroup",  
    "DBSecurityGroupDescription": "My Test Security Group",  
    "VpcId": "vpc-a1b2c3d4",  
    "EC2SecurityGroups": [],  
    "IPRanges": [],  
    "DBSecurityGroupArn": "arn:aws:rds:us-west-2:123456789012:secgrp:mysecgroup"  
  }  
}
```

- API-Einheiten finden Sie [CreateDbSecurityGroup](#) unter AWS CLI Befehlsreferenz.

create-db-shard-group

Das folgende Codebeispiel zeigt die Verwendung `create-db-shard-group`.

AWS CLI

Beispiel 1: So erstellen Sie einen SQL primären Aurora Postgre-DB-Cluster

Im folgenden `create-db-cluster` Beispiel wird ein SQL primärer Aurora Postgre-DB-Cluster erstellt, der mit Aurora Serverless v2 und Aurora Limitless Database kompatibel ist.

```
aws rds create-db-cluster \  
  --db-cluster-identifier my-sv2-cluster \  
  --engine aurora-postgresql \  
  --engine-version 15.2-limitless \  
  --storage-type aurora-iopt1 \  
  --serverless-v2-scaling-configuration MinCapacity=2,MaxCapacity=16 \  
  --enable-limitless-database \  
  --master-username myuser \  
  --master-user-password mypassword \  
  --enable-cloudwatch-logs-exports postgresql
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-2b",  
      "us-east-2c",  
      "us-east-2a"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "my-sv2-cluster",  
    "DBClusterParameterGroup": "default.aurora-postgresql15",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "my-sv2-cluster.cluster-cekyceexample.us-  
east-2.rds.amazonaws.com",  
    "ReaderEndpoint": "my-sv2-cluster.cluster-ro-cekyceexample.us-  
east-2.rds.amazonaws.com",
```

```
"MultiAZ": false,
"Engine": "aurora-postgresql",
"EngineVersion": "15.2-limitless",
"Port": 5432,
"MasterUsername": "myuser",
"PreferredBackupWindow": "06:05-06:35",
"PreferredMaintenanceWindow": "mon:08:25-mon:08:55",
"ReadReplicaIdentifiers": [],
"DBClusterMembers": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-#####",
    "Status": "active"
  }
],
"HostedZoneId": "Z2XHWR1EXAMPLE",
"StorageEncrypted": false,
"DbClusterResourceId": "cluster-XYEDT6ML6FHIXH4Q2J1EXAMPLE",
"DBClusterArn": "arn:aws:rds:us-east-2:123456789012:cluster:my-sv2-cluster",
"AssociatedRoles": [],
"IAMDatabaseAuthenticationEnabled": false,
"ClusterCreateTime": "2024-02-19T16:24:07.771000+00:00",
"EnabledCloudwatchLogsExports": [
  "postgresql"
],
"EngineMode": "provisioned",
"DeletionProtection": false,
"HttpEndpointEnabled": false,
"CopyTagsToSnapshot": false,
"CrossAccountClone": false,
"DomainMemberships": [],
"TagList": [],
"StorageType": "aurora-iopt1",
"AutoMinorVersionUpgrade": true,
"ServerlessV2ScalingConfiguration": {
  "MinCapacity": 2.0,
  "MaxCapacity": 16.0
},
"NetworkType": "IPV4",
"I00ptimizedNextAllowedModificationTime":
"2024-03-21T16:24:07.781000+00:00",
"LimitlessDatabase": {
  "Status": "not-in-use",
  "MinRequiredACU": 96.0
}
```

```

    }
  }
}

```

Beispiel 2: Um die primäre (Writer-) DB-Instance zu erstellen

Im folgenden `create-db-instance` Beispiel wird eine primäre (Writer-) DB-Instance von Aurora Serverless v2 erstellt. Wenn Sie die Konsole verwenden, um einen DB-Cluster zu erstellen, erstellt Amazon RDS automatisch die Writer-DB-Instance für Ihren DB-Cluster. Wenn Sie jedoch den verwenden, AWS CLI um einen DB-Cluster zu erstellen, müssen Sie die Writer-DB-Instance für Ihren DB-Cluster mit dem `create-db-instance` AWS CLI Befehl explizit erstellen.

```

aws rds create-db-instance \
  --db-instance-identifier my-sv2-instance \
  --db-cluster-identifier my-sv2-cluster \
  --engine aurora-postgresql \
  --db-instance-class db.serverless

```

Ausgabe:

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "my-sv2-instance",
    "DBInstanceClass": "db.serverless",
    "Engine": "aurora-postgresql",
    "DBInstanceStatus": "creating",
    "MasterUsername": "myuser",
    "AllocatedStorage": 1,
    "PreferredBackupWindow": "06:05-06:35",
    "BackupRetentionPeriod": 1,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-#####",
        "Status": "active"
      }
    ],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.aurora-postgresql15",
        "ParameterApplyStatus": "in-sync"
      }
    ]
  }
}

```

```
],
  "DBSubnetGroup": {
    "DBSubnetGroupName": "default",
    "DBSubnetGroupDescription": "default",
    "VpcId": "vpc-#####",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-#####",
        "SubnetAvailabilityZone": {
          "Name": "us-east-2c"
        },
        "SubnetOutpost": {},
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-#####",
        "SubnetAvailabilityZone": {
          "Name": "us-east-2a"
        },
        "SubnetOutpost": {},
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-#####",
        "SubnetAvailabilityZone": {
          "Name": "us-east-2b"
        },
        "SubnetOutpost": {},
        "SubnetStatus": "Active"
      }
    ]
  },
  "PreferredMaintenanceWindow": "fri:09:01-fri:09:31",
  "PendingModifiedValues": {
    "PendingCloudwatchLogsExports": {
      "LogTypesToEnable": [
        "postgresql"
      ]
    }
  },
  "MultiAZ": false,
  "EngineVersion": "15.2-limitless",
  "AutoMinorVersionUpgrade": true,
```

```

    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "postgresql-license",
    "OptionGroupMemberships": [
      {
        "OptionGroupName": "default:aurora-postgresql-15",
        "Status": "in-sync"
      }
    ],
    "PubliclyAccessible": false,
    "StorageType": "aurora-iopt1",
    "DbInstancePort": 0,
    "DBClusterIdentifier": "my-sv2-cluster",
    "StorageEncrypted": false,
    "DbiResourceId": "db-BIQTE3B3K3RM7M74SK5EXAMPLE",
    "CACertificateIdentifier": "rds-ca-rsa2048-g1",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
    "MonitoringInterval": 0,
    "PromotionTier": 1,
    "DBInstanceArn": "arn:aws:rds:us-east-2:123456789012:db:my-sv2-instance",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": [],
    "TagList": [],
    "CustomerOwnedIpEnabled": false,
    "BackupTarget": "region",
    "NetworkType": "IPV4",
    "StorageThroughput": 0,
    "CertificateDetails": {
      "CAIdentifier": "rds-ca-rsa2048-g1"
    },
    "DedicatedLogVolume": false
  }
}

```

Beispiel 3: So erstellen Sie die DB-Shard-Gruppe

Das folgende `create-db-shard-group` Beispiel erstellt eine DB-Shard-Gruppe in Ihrem SQL primären Aurora Postgre-DB-Cluster.

```

aws rds create-db-shard-group \
  --db-shard-group-identifier my-db-shard-group \

```

```
--db-cluster-identifizier my-sv2-cluster \  
--max-acu 768
```

Ausgabe:

```
{  
  "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",  
  "DBShardGroupIdentifizier": "my-db-shard-group",  
  "DBClusterIdentifizier": "my-sv2-cluster",  
  "MaxACU": 768.0,  
  "ComputeRedundancy": 0,  
  "Status": "creating",  
  "PubliclyAccessible": false,  
  "Endpoint": "my-sv2-cluster.limitless-cekyceexample.us-east-2.rds.amazonaws.com"  
}
```

Weitere Informationen finden Sie unter [Using Aurora Serverless v2](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateDbShardGroup AWS CLI](#) Befehlsreferenz.

create-db-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-db-snapshot`.

AWS CLI

Um einen DB-Snapshot zu erstellen

Das folgende `create-db-snapshot` Beispiel erstellt einen DB-Snapshot.

```
aws rds create-db-snapshot \  
  --db-instance-identifizier database-mysql \  
  --db-snapshot-identifizier mydbsnapshot
```

Ausgabe:

```
{  
  "DBSnapshot": {  
    "DBSnapshotIdentifizier": "mydbsnapshot",  
    "DBInstanceIdentifizier": "database-mysql",  
  }  
}
```



```

    "Engine": "mysql",
    "AllocatedStorage": 100,
    "Status": "creating",
    "Port": 3306,
    "AvailabilityZone": "us-east-1b",
    "VpcId": "vpc-6594f31c",
    "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
    "MasterUsername": "admin",
    "EngineVersion": "5.6.40",
    "LicenseModel": "general-public-license",
    "SnapshotType": "manual",
    "Iops": 1000,
    "OptionGroupName": "default:mysql-5-6",
    "PercentProgress": 0,
    "StorageType": "io1",
    "Encrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
    "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
    "IAMDatabaseAuthenticationEnabled": false,
    "ProcessorFeatures": [],
    "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
  }
}

```

Weitere Informationen finden Sie unter [Erstellen eines DB-Snapshots](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [C createDBSnapshot](#) in der AWS CLI Befehlsreferenz.

create-db-subnet-group

Das folgende Codebeispiel zeigt, wie man `create-db-subnet-group`.

AWS CLI

Um eine DB-Subnetzgruppe zu erstellen

Im folgenden `create-db-subnet-group` Beispiel wird eine DB-Subnetzgruppe erstellt, die `mysubnetgroup` unter Verwendung vorhandener Subnetze aufgerufen wird.

```

aws rds create-db-subnet-group \
  --db-subnet-group-name mysubnetgroup \
  --db-subnet-group-description "test DB subnet group" \

```

--subnet-ids

```
'["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef"]'
```

Ausgabe:

```
{
  "DBSubnetGroup": {
    "DBSubnetGroupName": "mysubnetgroup",
    "DBSubnetGroupDescription": "test DB subnet group",
    "VpcId": "vpc-0f08e7610a1b2c3d4",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
      }
    ],
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:0123456789012:subgrp:mysubnetgroup"
  }
}
```

Weitere Informationen finden Sie unter [Erstellen einer DB-Instance VPC in a](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDbSubnetGroup](#) unter AWS CLI Befehlsreferenz.

create-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-event-subscription`.

AWS CLI

Um ein Event-Abonnement zu erstellen

Im folgenden `create-event-subscription` Beispiel wird ein Abonnement für Sicherungs- und Wiederherstellungsereignisse für DB-Instances im aktuellen AWS Konto erstellt.

Benachrichtigungen werden an ein Amazon Simple Notification Service-Thema gesendet, das von angegeben ist `--sns-topic-arn`.

```
aws rds create-event-subscription \  
  --subscription-name my-instance-events \  
  --source-type db-instance \  
  --event-categories '["backup","recovery"]' \  
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Ausgabe:

```
{  
  "EventSubscription": {  
    "Status": "creating",  
    "CustSubscriptionId": "my-instance-events",  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ],  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",  
    "CustomerAwsId": "123456789012",  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-  
events",  
    "SourceType": "db-instance",  
    "Enabled": true  
  }  
}
```

- API-Einheiten finden Sie [CreateEventSubscription](#) in der AWS CLI Befehlsreferenz.

create-global-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-global-cluster`.

AWS CLI

Um einen globalen DB-Cluster zu erstellen

Das folgende `create-global-cluster` Beispiel erstellt einen neuen Aurora My SQL - kompatiblen globalen DB-Cluster.

```
aws rds create-global-cluster \  
  --global-cluster-identifier myglobalcluster \  
  --engine aurora-mysql
```

Ausgabe:

```
{  
  "GlobalCluster": {  
    "GlobalClusterIdentifier": "myglobalcluster",  
    "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",  
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-  
cluster:myglobalcluster",  
    "Status": "available",  
    "Engine": "aurora-mysql",  
    "EngineVersion": "5.7.mysql_aurora.2.07.2",  
    "StorageEncrypted": false,  
    "DeletionProtection": false,  
    "GlobalClusterMembers": []  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen einer globalen Aurora-Datenbank](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [CreateGlobalCluster](#) unter AWS CLI Befehlsreferenz.

create-option-group

Das folgende Codebeispiel zeigt die Verwendung `create-option-group`.

AWS CLI

So erstellen Sie eine RDS Amazon-Optionsgruppe

Der folgende `create-option-group` Befehl erstellt eine neue RDS Amazon-Optionsgruppe für Oracle Enterprise Edition Version 11.2, is named `MyOptionGroup` und enthält eine Beschreibung.

```
aws rds create-option-group \  
  --option-group-name MyOptionGroup \  
  --engine-name oracle-ee \  
  --major-engine-version 11.2 \  
  --option-group-description "Oracle Database Manager Database Control"
```

Ausgabe:

```
{  
  "OptionGroup": {  
    "OptionGroupName": "myoptiongroup",  
    "OptionGroupDescription": "Oracle Database Manager Database Control",  
    "EngineName": "oracle-ee",  
    "MajorEngineVersion": "11.2",  
    "Options": [],  
    "AllowsVpcAndNonVpcInstanceMemberships": true,  
    "OptionGroupArn": "arn:aws:rds:us-west-2:123456789012:og:myoptiongroup"  
  }  
}
```

- API-Einheiten finden Sie [CreateOptionGroup](#) in der AWS CLI Befehlsreferenz.

delete-blue-green-deployment

Das folgende Codebeispiel zeigt die Verwendung `delete-blue-green-deployment`.

AWS CLI

Beispiel 1: Um Ressourcen in einer grünen Umgebung für eine RDS for My SQL DB-Instance zu löschen

Im folgenden `delete-blue-green-deployment` Beispiel werden die Ressourcen in einer grünen Umgebung für eine RDS for My SQL DB-Instance gelöscht.

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-v53303651eexfake \  
  --delete-target
```

Ausgabe:

```
{  
  "BlueGreenDeployment": {  
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",  
    "BlueGreenDeploymentName": "bgd-cli-test-instance",  
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",  
    "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-  
rkfbpe",  
    "SwitchoverDetails": [  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-green-rkfbpe",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-1",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-1-green-j382ha",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-2",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-2-green-ejv4ao",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-3",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-3-green-vlpz3t",  
        "Status": "AVAILABLE"  
      }  
    ],  
  },  
}
```

```

    "Tasks": [
      {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
      },
      {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
      },
      {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
      },
      {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
      }
    ],
    "Status": "DELETING",
    "CreateTime": "2022-02-25T21:18:51.183000+00:00",
    "DeleteTime": "2022-02-25T22:25:31.331000+00:00"
  }
}

```

Weitere Informationen finden Sie unter [Löschen einer blauen/grünen Bereitstellung](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: Um Ressourcen in einer grünen Umgebung für einen Aurora My SQL DB-Cluster zu löschen

Das folgende delete-blue-green-deployment Beispiel löscht die Ressourcen in einer grünen Umgebung für einen Aurora My SQL DB-Cluster.

```

aws rds delete-blue-green-deployment \
  --blue-green-deployment-identifizier bgd-wi89nwzglccsfake \
  --delete-target

```

Ausgabe:

```

{
  "BlueGreenDeployment": {
    "BlueGreenDeploymentIdentifizier": "bgd-wi89nwzglccsfake",
    "BlueGreenDeploymentName": "my-blue-green-deployment",

```

```
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
    "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-gpmaxf",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-j2oajq",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-mkxies",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
```



```

        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwzlg",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],
"Status": "DELETING",
"CreateTime": "2022-02-25T21:12:00.288000+00:00",
"DeleteTime": "2022-02-25T22:29:11.336000+00:00"
}
}

```

Weitere Informationen finden Sie unter [Löschen einer blauen/grünen Bereitstellung](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteBlueGreenDeployment AWS CLI Befehlsreferenz](#).

delete-db-cluster-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster-endpoint`.

AWS CLI

Um einen benutzerdefinierten DB-Cluster-Endpunkt zu löschen

Im folgenden `delete-db-cluster-endpoint` Beispiel wird der angegebene benutzerdefinierte DB-Cluster-Endpoint gelöscht.

```
aws rds delete-db-cluster-endpoint \  
--db-cluster-endpoint-identifier mycustomendpoint
```

Ausgabe:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnpeexample.us-  
east-1.rds.amazonaws.com",  
  "Status": "deleting",  
  "EndpointType": "CUSTOM",  
  "CustomEndpointType": "READER",  
  "StaticMembers": [  
    "dbinstance1",  
    "dbinstance2",  
    "dbinstance3"  
  ],  
  "ExcludedMembers": [],  
  "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
endpoint:mycustomendpoint"  
}
```

Weitere Informationen finden Sie unter [Amazon Aurora Connection Management](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbClusterEndpoint](#) unter AWS CLI Befehlsreferenz.

delete-db-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster-parameter-group`.

AWS CLI

Um eine DB-Cluster-Parametergruppe zu löschen

Im folgenden `delete-db-cluster-parameter-group` Beispiel wird die angegebene DB-Cluster-Parametergruppe gelöscht.

```
aws rds delete-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbClusterParameterGroup](#) unter AWS CLI Befehlsreferenz.

delete-db-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster-snapshot`.

AWS CLI

Um einen DB-Cluster-Snapshot zu löschen

Im folgenden `delete-db-cluster-snapshot` Beispiel wird der angegebene DB-Cluster-Snapshot gelöscht.

```
aws rds delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifizier mydbclustersnapshot
```

Ausgabe:

```
{  
  "DBClusterSnapshot": {  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1e"  
    ],  
    "DBClusterSnapshotIdentifizier": "mydbclustersnapshot",  
    "DBClusterIdentifizier": "mydbcluster",  
    "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",  
    "Engine": "aurora-mysql",  
    "AllocatedStorage": 0,  
    "Status": "available",  
    "Port": 0,  
    "VpcId": "vpc-6594f31c",  
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
```

```
    "MasterUsername": "myadmin",
    "EngineVersion": "5.7.mysql_aurora.2.04.2",
    "LicenseModel": "aurora-mysql",
    "SnapshotType": "manual",
    "PercentProgress": 100,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:mydbclustersnapshot",
    "IAMDatabaseAuthenticationEnabled": false
  }
}
```

Weitere Informationen finden Sie unter [Löschen eines Snapshots](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbClusterSnapshot](#) unter AWS CLI Befehlsreferenz.

delete-db-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-db-cluster`.

AWS CLI

Beispiel 1: Um eine DB-Instance in einem DB-Cluster zu löschen

Im folgenden `delete-db-instance` Beispiel wird die letzte DB-Instance in einem DB-Cluster gelöscht. Sie können einen DB-Cluster nicht löschen, wenn er DB-Instances enthält, die sich nicht im Löschstadium befinden. Sie können keinen endgültigen Snapshot erstellen, wenn Sie eine DB-Instance in einem DB-Cluster löschen.

```
aws rds delete-db-instance \
  --db-instance-identifier database-3
```

Ausgabe:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "database-3",
    "DBInstanceClass": "db.r4.large",
    "Engine": "aurora-postgresql",
    "DBInstanceStatus": "deleting",
```

```
...output omitted...  
}  
}
```

Weitere Informationen finden Sie unter [Löschen einer DB-Instance in einem Aurora-DB-Cluster](#) im Amazon Aurora Benutzerhandbuch.

Beispiel 2: Um einen DB-Cluster zu löschen

Im folgenden `delete-db-cluster` Beispiel wird der angegebene DB-Cluster gelöscht `mycluster` und ein letzter Snapshot mit dem Namen `mycluster-final-snapshot` erstellt. Der Status des DB-Clusters ist verfügbar, während der Snapshot erstellt wird. Verwenden Sie den `describe-db-clusters` CLI Befehl, um den Fortschritt des Löschvorgangs zu verfolgen.

```
aws rds delete-db-cluster \  
  --db-cluster-identifizier mycluster \  
  --no-skip-final-snapshot \  
  --final-db-snapshot-identifizier mycluster-final-snapshot
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 20,  
    "AvailabilityZones": [  
      "eu-central-1b",  
      "eu-central-1c",  
      "eu-central-1a"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DBClusterIdentifizier": "mycluster",  
    "DBClusterParameterGroup": "default.aurora-postgresql10",  
    "DBSubnetGroup": "default-vpc-aa11bb22",  
    "Status": "available",  
  
    ...output omitted...  
  }  
}
```

Weitere Informationen finden Sie unter [Aurora Clusters with a Single DB Instance](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbCluster](#) unter AWS CLI Befehlsreferenz.

delete-db-instance-automated-backup

Das folgende Codebeispiel zeigt die Verwendung `delete-db-instance-automated-backup`.

AWS CLI

Um ein repliziertes automatisiertes Backup aus einer Region zu löschen

Im folgenden `delete-db-instance-automated-backup` Beispiel wird die automatische Sicherung mit dem angegebenen Amazon-Ressourcennamen (ARN) gelöscht.

```
aws rds delete-db-instance-automated-backup \
  --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-
  backup:ab-jkib2gfg5rv7replzadabrktni2bn4example"
```

Ausgabe:

```
{
  "DBInstanceAutomatedBackup": {
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
    "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
    "Region": "us-east-1",
    "DBInstanceIdentifier": "new-orcl-db",
    "RestoreWindow": {},
    "AllocatedStorage": 20,
    "Status": "deleting",
    "Port": 1521,
    "AvailabilityZone": "us-east-1b",
    "VpcId": "vpc-#####",
    "InstanceCreateTime": "2020-12-04T15:28:31Z",
    "MasterUsername": "admin",
    "Engine": "oracle-se2",
    "EngineVersion": "12.1.0.2.v21",
    "LicenseModel": "bring-your-own-license",
    "OptionGroupName": "default:oracle-se2-12-1",
    "Encrypted": false,
    "StorageType": "gp2",
    "IAMDatabaseAuthenticationEnabled": false,
```

```
    "BackupRetentionPeriod": 7,  
    "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-  
backup:ab-jkib2gfg5rv7replzadabrktni2bn4example"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen replizierter Backups](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbInstanceAutomatedBackup](#) in der AWS CLI Befehlsreferenz.

delete-db-instance

Das folgende Codebeispiel zeigt die Verwendung `delete-db-instance`.

AWS CLI

Um eine DB-Instance zu löschen

Im folgenden `delete-db-instance` Beispiel wird die angegebene DB-Instance gelöscht, nachdem ein letzter DB-Snapshot mit dem Namen erstellt wurde `test-instance-final-snap`.

```
aws rds delete-db-instance \  
  --db-instance-identifizier test-instance \  
  --final-db-snapshot-identifizier test-instance-final-snap
```

Ausgabe:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifizier": "test-instance",  
    "DBInstanceStatus": "deleting",  
    ...some output truncated...  
  }  
}
```

- API-Einheiten finden Sie unter [DeleteDBInstance](#) in der AWS CLI Befehlsreferenz.

delete-db-parameter-group

Das folgende Codebeispiel zeigt, wie man `delete-db-parameter-group`.

AWS CLI

Um eine DB-Parametergruppe zu löschen

Im folgenden command Beispiel wird eine DB-Parametergruppe gelöscht.

```
aws rds delete-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [deleteDBParameterD-Gruppe](#) in der AWS CLI Befehlsreferenz.

delete-db-proxy-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-db-proxy-endpoint`.

AWS CLI

Um einen DB-Proxyendpunkt für eine RDS Datenbank zu löschen

Im folgenden `delete-db-proxy-endpoint` Beispiel wird ein DB-Proxyendpunkt für die Zieldatenbank gelöscht.

```
aws rds delete-db-proxy-endpoint \  
  --db-proxy-endpoint-name proxyEP1
```

Ausgabe:

```
{  
  "DBProxyEndpoint":  
    {  
      "DBProxyEndpointName": "proxyEP1",  
      "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-  
endpoint:prx-endpoint-0123a01b12345c0ab",  
      "DBProxyName": "proxyExample",  
      "Status": "deleting",  
      "VpcId": "vpc-1234567",  
      "VpcSecurityGroupIds": [  

```



```

        "sg-1234",
        "sg-5678"
    ],
    "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
    ],
    "Endpoint": "proxyEP1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
    "CreateDate": "2023-04-13T01:49:38.568000+00:00",
    "TargetRole": "READ_ONLY",
    "IsDefault": false
  }
}

```

Weitere Informationen finden Sie unter [Löschen eines Proxy-Endpunkts](#) im RDSAmazon-Benutzerhandbuch und [Löschen eines Proxy-Endpunkts](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbProxyEndpoint](#) in der AWS CLI Befehlsreferenz.

delete-db-proxy

Das folgende Codebeispiel zeigt die Verwendung `delete-db-proxy`.

AWS CLI

Um einen DB-Proxy für eine RDS Datenbank zu löschen

Das folgende `delete-db-proxy` Beispiel löscht einen DB-Proxy.

```

aws rds delete-db-proxy \
  --db-proxy-name proxyExample

```

Ausgabe:

```

{
  "DBProxy": {
    "DBProxyName": "proxyExample",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
    "Status": "deleting",
    "EngineFamily": "PostgreSQL",
  }
}

```

```

    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
        "sg-1234",
        "sg-5678"
    ],
    "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
    ],
    "Auth": "[
        {
            "Description": "proxydescription`"
            "AuthScheme": "SECRETS",
            "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
            "IAMAuth": "DISABLED"
        } ],
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
    "RequireTLS": false,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdateDate": "2023-04-13T01:49:38.568000+00:00"
}
}

```

Weitere Informationen finden Sie unter [Löschen eines RDS Proxys](#) im RDS Amazon-Benutzerhandbuch und [Löschen eines RDS Proxys](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbProxy](#) in der AWS CLI Befehlsreferenz.

delete-db-security-group

Das folgende Codebeispiel zeigt die Verwendung `delete-db-security-group`.

AWS CLI

Um eine DB-Sicherheitsgruppe zu löschen

Im folgenden `delete-db-security-group` Beispiel wird eine DB-Sicherheitsgruppe mit dem Namen `mysecuritygroup` gelöscht.

```
aws rds delete-db-security-group \  
  --db-security-group-name mysecuritygroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit DB-Sicherheitsgruppen \(EC2-Classik-Plattform\)](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteDbSecurityGroup AWS CLIBefehlsreferenz](#).

delete-db-shard-group

Das folgende Codebeispiel zeigt die Verwendung `delete-db-shard-group`.

AWS CLI

Beispiel 1: Um eine DB-Shard-Gruppe erfolglos zu löschen

Das folgende `delete-db-shard-group` Beispiel zeigt den Fehler, der auftritt, wenn Sie versuchen, eine DB-Shard-Gruppe zu löschen, bevor Sie alle Ihre Datenbanken und Schemas löschen.

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifizier Limitless-test-shard-grp
```

Ausgabe:

```
An error occurred (InvalidDBShardGroupState) when calling the DeleteDBShardGroup  
operation: Unable to delete the DB shard group limitless-test-db-shard-group.  
Delete all of your Limitless Database databases and schemas, then try again.
```

Beispiel 2: Um eine DB-Shard-Gruppe erfolgreich zu löschen

Im folgenden `delete-db-shard-group` Beispiel wird eine DB-Shard-Gruppe gelöscht, nachdem Sie alle Ihre Datenbanken und Schemas, einschließlich des Schemas, gelöscht haben.

```
public
```

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifizier Limitless-test-shard-grp
```

Ausgabe:

```
{
  "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
  "DBShardGroupIdentifier": "limitless-test-shard-grp",
  "DBClusterIdentifier": "limitless-test-cluster",
  "MaxACU": 768.0,
  "ComputeRedundancy": 0,
  "Status": "deleting",
  "PubliclyAccessible": true,
  "Endpoint": "limitless-test-cluster.limitless-cekyceexample.us-east-2.rds.amazonaws.com"
}
```

Weitere Informationen finden Sie unter [Löschen von Aurora-DB-Clustern und DB-Instances](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDbShardGroup](#) unter AWS CLI Befehlsreferenz.

delete-db-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-db-snapshot`.

AWS CLI

Um einen DB-Snapshot zu löschen

Das folgende `delete-db-snapshot` Beispiel löscht den angegebenen DB-Snapshot.

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier mydbsnapshot
```

Ausgabe:

```
{
  "DBSnapshot": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBInstanceIdentifier": "database-mysql",
    "SnapshotCreateTime": "2019-06-18T22:08:40.702Z",
    "Engine": "mysql",
    "AllocatedStorage": 100,
    "Status": "deleted",
    "Port": 3306,
    "AvailabilityZone": "us-east-1b",
  }
}
```

```
"VpcId": "vpc-6594f31c",
"InstanceCreateTime": "2019-04-30T15:45:53.663Z",
"MasterUsername": "admin",
"EngineVersion": "5.6.40",
"LicenseModel": "general-public-license",
"SnapshotType": "manual",
"Iops": 1000,
"OptionGroupName": "default:mysql-5-6",
"PercentProgress": 100,
"StorageType": "io1",
"Encrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
"IAMDatabaseAuthenticationEnabled": false,
"ProcessorFeatures": [],
"DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
}
}
```

Weitere Informationen finden Sie unter [Löschen eines Snapshots](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteDbSnapshot](#) in der AWS CLI Befehlsreferenz.

delete-db-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `delete-db-subnet-group`.

AWS CLI

Um eine DB-Subnetzgruppe zu löschen

Im folgenden `delete-db-subnet-group` Beispiel wird die DB-Subnetzgruppe namens `mysubnetgroup` gelöscht.

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit einer DB-Instance VPC in a](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteDbSubnetGroup](#) unter AWS CLI Befehlsreferenz.

delete-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `delete-event-subscription`.

AWS CLI

Um ein Event-Abonnement zu löschen

Im folgenden `delete-event-subscription` Beispiel wird das angegebene Ereignisabonnement gelöscht.

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

Ausgabe:

```
{
  "EventSubscription": {
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-events",
    "CustomerAwsId": "123456789012",
    "Enabled": false,
    "SourceIdsList": [
      "test-instance"
    ],
    "SourceType": "db-instance",
    "EventCategoriesList": [
      "backup",
      "recovery"
    ],
    "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
    "CustSubscriptionId": "my-instance-events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
    "Status": "deleting"
  }
}
```

- API-Einheiten finden Sie unter [DeleteEventSubscription AWS CLI Befehlsreferenz](#).

delete-global-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-global-cluster`.

AWS CLI

Um einen globalen DB-Cluster zu löschen

Das folgende `delete-global-cluster` Beispiel löscht einen Aurora My SQL -kompatiblen globalen DB-Cluster. Die Ausgabe zeigt den Cluster, den Sie löschen, aber nachfolgende `describe-global-clusters` Befehle führen diesen DB-Cluster nicht auf.

```
aws rds delete-global-cluster \  
  --global-cluster-identifier myglobalcluster
```

Ausgabe:

```
{  
  "GlobalCluster": {  
    "GlobalClusterIdentifier": "myglobalcluster",  
    "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",  
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-  
cluster:myglobalcluster",  
    "Status": "available",  
    "Engine": "aurora-mysql",  
    "EngineVersion": "5.7.mysql_aurora.2.07.2",  
    "StorageEncrypted": false,  
    "DeletionProtection": false,  
    "GlobalClusterMembers": []  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen einer globalen Aurora-Datenbank](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteGlobalCluster](#) unter AWS CLI Befehlsreferenz.

delete-option-group

Das folgende Codebeispiel zeigt die Verwendung `delete-option-group`.

AWS CLI

Um eine Optionsgruppe zu löschen

Im folgenden `delete-option-group` Beispiel wird die angegebene Optionsgruppe gelöscht.

```
aws rds delete-option-group \  
  --option-group-name myoptiongroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer Optionsgruppe](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeleteOptionGroup](#) in der AWS CLI Befehlsreferenz.

deregister-db-proxy-targets

Das folgende Codebeispiel zeigt die Verwendung `deregister-db-proxy-targets`.

AWS CLI

Um ein DB-Proxyziel von der Datenbank-Zielgruppe abzumelden

Das folgende `deregister-db-proxy-targets` Beispiel entfernt die Zuordnung zwischen dem Proxy `proxyExample` und seinem Ziel.

```
aws rds deregister-db-proxy-targets \  
  --db-proxy-name proxyExample \  
  --db-instance-identifiers database-1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen eines RDS Proxys](#) im RDSAmazon-Benutzerhandbuch und [Löschen eines RDS Proxys](#) im Amazon Aurora-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DeregisterDbProxyTargets](#) in der AWS CLI Befehlsreferenz.

describe-account-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-account-attributes`.

AWS CLI

Zur Beschreibung von Kontoattributen

Im folgenden `describe-account-attributes` Beispiel werden die Attribute für das AWS Girokonto abgerufen.

aws rds describe-account-attributes

Ausgabe:

```
{
  "AccountQuotas": [
    {
      "Max": 40,
      "Used": 4,
      "AccountQuotaName": "DBInstances"
    },
    {
      "Max": 40,
      "Used": 0,
      "AccountQuotaName": "ReservedDBInstances"
    },
    {
      "Max": 100000,
      "Used": 40,
      "AccountQuotaName": "AllocatedStorage"
    },
    {
      "Max": 25,
      "Used": 0,
      "AccountQuotaName": "DBSecurityGroups"
    },
    {
      "Max": 20,
      "Used": 0,
      "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
    },
    {
      "Max": 50,
      "Used": 1,
      "AccountQuotaName": "DBParameterGroups"
    },
    {
      "Max": 100,
      "Used": 3,
      "AccountQuotaName": "ManualSnapshots"
    },
    {
      "Max": 20,
```

```

    "Used": 0,
    "AccountQuotaName": "EventSubscriptions"
  },
  {
    "Max": 50,
    "Used": 1,
    "AccountQuotaName": "DBSubnetGroups"
  },
  {
    "Max": 20,
    "Used": 1,
    "AccountQuotaName": "OptionGroups"
  },
  {
    "Max": 20,
    "Used": 6,
    "AccountQuotaName": "SubnetsPerDBSubnetGroup"
  },
  {
    "Max": 5,
    "Used": 0,
    "AccountQuotaName": "ReadReplicasPerMaster"
  },
  {
    "Max": 40,
    "Used": 1,
    "AccountQuotaName": "DBClusters"
  },
  {
    "Max": 50,
    "Used": 0,
    "AccountQuotaName": "DBClusterParameterGroups"
  },
  {
    "Max": 5,
    "Used": 0,
    "AccountQuotaName": "DBClusterRoles"
  }
]
}

```

- API-Einheiten finden Sie unter [DescribeAccountAttributes AWS CLI](#) Befehlsreferenz.

describe-blue-green-deployments

Das folgende Codebeispiel zeigt die Verwendung `describe-blue-green-deployments`.

AWS CLI

Beispiel 1: Um eine blaue/grüne Bereitstellung einer RDS DB-Instance nach Abschluss der Erstellung zu beschreiben

Im folgenden `describe-blue-green-deployment` Beispiel werden nach Abschluss der Erstellung die Details einer blauen/grünen Bereitstellung abgerufen.

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-v53303651eexfake
```

Ausgabe:

```
{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
      "BlueGreenDeploymentName": "bgd-cli-test-instance",
      "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
      "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-rkfbpe",
      "SwitchoverDetails": [
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-rkfbpe",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-1",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-1-green-j382ha",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-2",
```

```

        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ejv4ao",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-vlpz3t",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
    }
],
"Status": "AVAILABLE",
"CreateTime": "2022-02-25T21:18:51.183000+00:00"
}
]
}

```

Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellungen anzeigen](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: Um eine blaue/grüne Bereitstellung für einen Aurora My SQL DB-Cluster zu beschreiben

Im folgenden `describe-blue-green-deployment` Beispiel werden die Details einer blauen/grünen Bereitstellung abgerufen.

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifizier bgd-wi89nwzglccsfake
```

Ausgabe:

```
{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifizier": "bgd-wi89nwzglccsfake",
      "BlueGreenDeploymentName": "my-blue-green-deployment",
      "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
      "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-green-3rnukl",
      "SwitchoverDetails": [
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-green-3rnukl",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1-green-gpmaxf",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2-green-j2oajq",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-3",
```

```
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-3-green-mkxies",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-excluded-member-endpoint-green-4sqjrq",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-reader-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-reader-endpoint-green-gwzlg",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],
"Status": "AVAILABLE",
"CreateTime": "2022-02-25T21:12:00.288000+00:00"
}
]
```

Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellung anzeigen](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Beispiel 3: Um eine blaue/grüne Bereitstellung für einen Aurora SQL My-Cluster nach dem Switchover zu beschreiben

Im folgenden `describe-blue-green-deployment` Beispiel werden die Details zu einer blauen/grünen Bereitstellung abgerufen, nachdem die grüne Umgebung zur Produktionsumgebung heraufgestuft wurde.

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

Ausgabe:

```
{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
      "BlueGreenDeploymentName": "my-blue-green-deployment",
      "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-old1",
      "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
      "SwitchoverDetails": [
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-old1",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
          "Status": "SWITCHOVER_COMPLETED"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1-old1",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1",
          "Status": "SWITCHOVER_COMPLETED"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2-old1",
```

```

        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
        "Status": "SWITCHOVER_COMPLETED"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
        "Status": "SWITCHOVER_COMPLETED"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
        "Status": "SWITCHOVER_COMPLETED"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
        "Status": "SWITCHOVER_COMPLETED"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],

```



```

        "Status": "SWITCHOVER_COMPLETED",
        "CreateTime": "2022-02-25T22:38:49.522000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellung anzeigen](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Beispiel 4: Um eine kombinierte blaue/grüne Bereitstellung zu beschreiben

Im folgenden `describe-blue-green-deployment` Beispiel werden die Details einer kombinierten Blau/Grün-Bereitstellung abgerufen.

```
aws rds describe-blue-green-deployments
```

Ausgabe:

```

{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-wi89nwzgfakelccs",
      "BlueGreenDeploymentName": "my-blue-green-deployment",
      "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
      "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-green-3rnukl",
      "SwitchoverDetails": [
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-green-3rnukl",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1-green-gpmaxf",
          "Status": "AVAILABLE"
        }
      ]
    }
  ]
}

```

```
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2-green-j2oajq",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-3-green-mkxies",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-excluded-member-endpoint-green-4sqjrq",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-reader-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-reader-endpoint-green-gwwzlg",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
]
```

```
    }
  ],
  "Status": "AVAILABLE",
  "CreateTime": "2022-02-25T21:12:00.288000+00:00"
},
{
  "BlueGreenDeploymentIdentifier": "bgd-v5330365fake1eex",
  "BlueGreenDeploymentName": "bgd-cli-test-instance",
  "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-old1",
  "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
  "SwitchoverDetails": [
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-old1",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
      "Status": "SWITCHOVER_COMPLETED"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-old1",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
      "Status": "SWITCHOVER_COMPLETED"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-old1",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
      "Status": "SWITCHOVER_COMPLETED"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-old1",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
      "Status": "SWITCHOVER_COMPLETED"
    }
  ],
  "Tasks": [
    {
      "Name": "CREATING_READ_REPLICA_OF_SOURCE",
      "Status": "COMPLETED"
    }
  ]
}
```

```

    },
    {
      "Name": "DB_ENGINE_VERSION_UPGRADE",
      "Status": "COMPLETED"
    },
    {
      "Name": "CONFIGURE_BACKUPS",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATING_TOPOLOGY_OF_SOURCE",
      "Status": "COMPLETED"
    }
  ],
  "Status": "SWITCHOVER_COMPLETED",
  "CreateTime": "2022-02-25T22:33:22.225000+00:00"
}
]
}

```

Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellungen anzeigen](#) im RDSAmazon-Benutzerhandbuch und [Blau/Grün-Bereitstellungen anzeigen](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeBlueGreenDeployments](#) in der AWS CLI Befehlsreferenz.

describe-certificates

Das folgende Codebeispiel zeigt die Verwendung `describe-certificates`.

AWS CLI

Um Zertifikate zu beschreiben

Im folgenden `describe-certificates` Beispiel werden die Details des Zertifikats abgerufen, das der Standardregion des Benutzers zugeordnet ist.

```
aws rds describe-certificates
```

Ausgabe:

```
{
  "Certificates": [
```

```

    {
      "CertificateIdentifier": "rds-ca-ecc384-g1",
      "CertificateType": "CA",
      "Thumbprint": "2ee3dcc06e50192559b13929e73484354f23387d",
      "ValidFrom": "2021-05-24T22:06:59+00:00",
      "ValidTill": "2121-05-24T23:06:59+00:00",
      "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-ecc384-g1",
      "CustomerOverride": false
    },
    {
      "CertificateIdentifier": "rds-ca-rsa4096-g1",
      "CertificateType": "CA",
      "Thumbprint": "19da4f2af579a8ae1f6a0fa77aa5befd874b4cab",
      "ValidFrom": "2021-05-24T22:03:20+00:00",
      "ValidTill": "2121-05-24T23:03:20+00:00",
      "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa4096-g1",
      "CustomerOverride": false
    },
    {
      "CertificateIdentifier": "rds-ca-rsa2048-g1",
      "CertificateType": "CA",
      "Thumbprint": "7c40cb42714b6fdb2b296f9bbd0e8bb364436a76",
      "ValidFrom": "2021-05-24T21:59:00+00:00",
      "ValidTill": "2061-05-24T22:59:00+00:00",
      "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa2048-g1",
      "CustomerOverride": true,
      "CustomerOverrideValidTill": "2061-05-24T22:59:00+00:00"
    },
    {
      "CertificateIdentifier": "rds-ca-2019",
      "CertificateType": "CA",
      "Thumbprint": "d40ddb29e3750dfffa671c3140bbf5f478d1c8096",
      "ValidFrom": "2019-08-22T17:08:50+00:00",
      "ValidTill": "2024-08-22T17:08:50+00:00",
      "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-2019",
      "CustomerOverride": false
    }
  ],
  "DefaultCertificateForNewLaunches": "rds-ca-rsa2048-g1"
}

```

Weitere Informationen finden Sie unter [Verwenden von SSL/TLS zum Verschlüsseln einer Verbindung zu einer DB-Instance](#) im RDS Amazon-Benutzerhandbuch und [Verwenden](#)

[vonSSL/TLSzum Verschlüsseln einer Verbindung zu einem DB-Cluster](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCertificates](#) in der AWS CLI Befehlsreferenz.

describe-db-cluster-backtracks

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-backtracks`.

AWS CLI

Um Backtracks für einen DB-Cluster zu beschreiben

Im folgenden `describe-db-cluster-backtracks` Beispiel werden Details zum angegebenen DB-Cluster abgerufen.

```
aws rds describe-db-cluster-backtracks \  
  --db-cluster-identifier mydbcluster
```

Ausgabe:

```
{  
  "DBClusterBacktracks": [  
    {  
      "DBClusterIdentifier": "mydbcluster",  
      "BacktrackIdentifier": "2f5f5294-0dd2-44c9-9f50-EXAMPLE",  
      "BacktrackTo": "2021-02-12T04:59:22Z",  
      "BacktrackedFrom": "2021-02-12T14:37:31.640Z",  
      "BacktrackRequestCreationTime": "2021-02-12T14:36:18.819Z",  
      "Status": "COMPLETED"  
    },  
    {  
      "DBClusterIdentifier": "mydbcluster",  
      "BacktrackIdentifier": "3c7a6421-af2a-4ea3-ae95-EXAMPLE",  
      "BacktrackTo": "2021-02-11T22:53:46Z",  
      "BacktrackedFrom": "2021-02-12T00:09:27.006Z",  
      "BacktrackRequestCreationTime": "2021-02-12T00:07:53.487Z",  
      "Status": "COMPLETED"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Backtracking eines Aurora-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeDbClusterBacktracks AWS CLI Befehlsreferenz](#).

describe-db-cluster-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-endpoints`.

AWS CLI

Beispiel 1: Um DB-Cluster-Endpunkte zu beschreiben

Im folgenden `describe-db-cluster-endpoints` Beispiel werden Details für Ihre DB-Cluster-Endpunkte abgerufen. Die gängigsten Arten von Aurora-Clustern haben zwei Endpunkte. Ein Endpunkt hat Typ `WRITER`. Sie können diesen Endpunkt für alle SQL-Anweisungen verwenden. Der andere Endpunkt hat Typ `READER`. Sie können diesen Endpunkt nur für `SELECT` und andere schreibgeschützte Anweisungen SQL verwenden.

```
aws rds describe-db-cluster-endpoints
```

Ausgabe:

```
{
  "DBClusterEndpoints": [
    {
      "DBClusterIdentifier": "my-database-1",
      "Endpoint": "my-database-1.cluster-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "creating",
      "EndpointType": "WRITER"
    },
    {
      "DBClusterIdentifier": "my-database-1",
      "Endpoint": "my-database-1.cluster-ro-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "creating",
      "EndpointType": "READER"
    },
    {
      "DBClusterIdentifier": "mydbcluster",
```

```

        "Endpoint": "mydbcluster.cluster-cnpeaxmle.us-east-1.rds.amazonaws.com",
        "Status": "available",
        "EndpointType": "WRITER"
    },
    {
        "DBClusterIdentifier": "mydbcluster",
        "Endpoint": "mydbcluster.cluster-ro-cnpeaxmle.us-
east-1.rds.amazonaws.com",
        "Status": "available",
        "EndpointType": "READER"
    }
]
}

```

Beispiel 2: Um die DB-Cluster-Endpunkte eines einzelnen DB-Clusters zu beschreiben

Im folgenden `describe-db-cluster-endpoints` Beispiel werden Details für die DB-Cluster-Endpunkte eines einzelnen angegebenen DB-Clusters abgerufen. Aurora Serverless-Cluster haben nur einen einzigen Endpunkt mit einem Typ von `WRITER`.

```

aws rds describe-db-cluster-endpoints \
  --db-cluster-identifier serverless-cluster

```

Ausgabe:

```

{
  "DBClusterEndpoints": [
    {
      "Status": "available",
      "Endpoint": "serverless-cluster.cluster-cnpeaxmle.us-
east-1.rds.amazonaws.com",
      "DBClusterIdentifier": "serverless-cluster",
      "EndpointType": "WRITER"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Amazon Aurora Connection Management](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbClusterEndpoints](#) unter AWS CLI Befehlsreferenz.

describe-db-cluster-parameter-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-parameter-groups`.

AWS CLI

Um DB-Cluster-Parametergruppen zu beschreiben

Im folgenden `describe-db-cluster-parameter-groups` Beispiel werden Details für Ihre DB-Cluster-Parametergruppen abgerufen.

```
aws rds describe-db-cluster-parameter-groups
```

Ausgabe:

```
{
  "DBClusterParameterGroups": [
    {
      "DBClusterParameterGroupName": "default.aurora-mysql5.7",
      "DBParameterGroupFamily": "aurora-mysql5.7",
      "Description": "Default cluster parameter group for aurora-mysql5.7",
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora-mysql5.7"
    },
    {
      "DBClusterParameterGroupName": "default.aurora-postgresql9.6",
      "DBParameterGroupFamily": "aurora-postgresql9.6",
      "Description": "Default cluster parameter group for aurora-postgresql9.6",
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora-postgresql9.6"
    },
    {
      "DBClusterParameterGroupName": "default.aurora5.6",
      "DBParameterGroupFamily": "aurora5.6",
      "Description": "Default cluster parameter group for aurora5.6",
      "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora5.6"
    },
    {
      "DBClusterParameterGroupName": "mydbclusterpg",
      "DBParameterGroupFamily": "aurora-mysql5.7",
      "Description": "My DB cluster parameter group",

```

```

        "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpg"
    },
    {
        "DBClusterParameterGroupName": "mydbclusterpgcopy",
        "DBParameterGroupFamily": "aurora-mysql5.7",
        "Description": "Copy of mydbclusterpg parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpgcopy"
    }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbClusterParameterGroups](#) unter AWS CLI Befehlsreferenz.

describe-db-cluster-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-parameters`.

AWS CLI

Beispiel 1: Um die Parameter in einer DB-Cluster-Parametergruppe zu beschreiben

Im folgenden `describe-db-cluster-parameters` Beispiel werden Details zu den Parametern in einer DB-Cluster-Parametergruppe abgerufen.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name mydbclusterpg

```

Ausgabe:

```

{
  "Parameters": [
    {
      "ParameterName": "allow-suspicious-udfs",
      "Description": "Controls whether user-defined functions that have only
an xxx symbol for the main function can be loaded",
      "Source": "engine-default",

```

```

        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": false,
        "ApplyMethod": "pending-reboot",
        "SupportedEngineModes": [
            "provisioned"
        ]
    },
    {
        "ParameterName": "aurora_lab_mode",
        "ParameterValue": "0",
        "Description": "Enables new features in the Aurora engine.",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": true,
        "ApplyMethod": "pending-reboot",
        "SupportedEngineModes": [
            "provisioned"
        ]
    },
    ...some output truncated...
]
}

```

Beispiel 2: Um nur die Parameternamen in einer DB-Cluster-Parametergruppe aufzulisten

Im folgenden `describe-db-cluster-parameters` Beispiel werden nur die Namen der Parameter in einer DB-Cluster-Parametergruppe abgerufen.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].{ParameterName:ParameterName}'

```

Ausgabe:

```

[
  {
    "ParameterName": "allow-suspicious-udfs"
  },
  {

```

```

    "ParameterName": "aurora_binlog_read_buffer_size"
  },
  {
    "ParameterName": "aurora_binlog_replication_max_yield_seconds"
  },
  {
    "ParameterName": "aurora_binlog_use_large_read_buffer"
  },
  {
    "ParameterName": "aurora_lab_mode"
  },
  ...some output truncated...
}
]

```

Beispiel 3: Um nur die änderbaren Parameter in einer DB-Cluster-Parametergruppe zu beschreiben

Im folgenden `describe-db-cluster-parameters` Beispiel werden nur die Namen der Parameter abgerufen, die Sie in einer DB-Cluster-Parametergruppe ändern können.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].[ParameterName:ParameterName,IsModifiable:IsModifiable] | [?IsModifiable == `true`]'

```

Ausgabe:

```

[
  {
    "ParameterName": "aurora_binlog_read_buffer_size",
    "IsModifiable": true
  },
  {
    "ParameterName": "aurora_binlog_replication_max_yield_seconds",
    "IsModifiable": true
  },
  {
    "ParameterName": "aurora_binlog_use_large_read_buffer",
    "IsModifiable": true
  },
]

```

```

    {
      "ParameterName": "aurora_lab_mode",
      "IsModifiable": true
    },
    ...some output truncated...
  }
]

```

Beispiel 4: Um nur die änderbaren booleschen Parameter in einer DB-Cluster-Parametergruppe zu beschreiben

Im folgenden `describe-db-cluster-parameters` Beispiel werden nur die Namen der Parameter abgerufen, die Sie in einer DB-Cluster-Parametergruppe ändern können und die einen booleschen Datentyp haben.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].
{ParameterName:ParameterName,DataType:DataType,IsModifiable:IsModifiable} | [?DataType == `boolean` | [?IsModifiable == `true`]'

```

Ausgabe:

```

[
  {
    "DataType": "boolean",
    "ParameterName": "aurora_binlog_use_large_read_buffer",
    "IsModifiable": true
  },
  {
    "DataType": "boolean",
    "ParameterName": "aurora_lab_mode",
    "IsModifiable": true
  },
  {
    "DataType": "boolean",
    "ParameterName": "autocommit",
    "IsModifiable": true
  },
  {
    "DataType": "boolean",

```

```
    "ParameterName": "automatic_sp_privileges",
    "IsModifiable": true
  },
  ...some output truncated...
}
]
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbClusterParameters](#) unter AWS CLI Befehlsreferenz.

describe-db-cluster-snapshot-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-snapshot-attributes`.

AWS CLI

Um die Attributnamen und -werte für einen DB-Cluster-Snapshot zu beschreiben

Im folgenden `describe-db-cluster-snapshot-attributes` Beispiel werden Details zu den Attributnamen und -werten für den angegebenen DB-Cluster-Snapshot abgerufen.

```
aws rds describe-db-cluster-snapshot-attributes \
  --db-cluster-snapshot-identifier myclustersnapshot
```

Ausgabe:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "myclustersnapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012"
        ]
      }
    ]
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Freigeben eines DB-Cluster-Snapshots](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbClusterSnapshotAttributes](#) unter AWS CLI Befehlsreferenz.

describe-db-cluster-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-db-cluster-snapshots`.

AWS CLI

Um einen DB-Cluster-Snapshot für einen DB-Cluster zu beschreiben

Im folgenden `describe-db-cluster-snapshots` Beispiel werden die Details für die DB-Cluster-Snapshots für den angegebenen DB-Cluster abgerufen.

```
aws rds describe-db-cluster-snapshots \
  --db-cluster-identifier mydbcluster
```

Ausgabe:

```
{
  "DBClusterSnapshots": [
    {
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1e"
      ],
      "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
      "DBClusterIdentifier": "mydbcluster",
      "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
      "Engine": "aurora-mysql",
      "AllocatedStorage": 0,
      "Status": "available",
      "Port": 0,
      "VpcId": "vpc-6594f31c",
      "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
      "MasterUsername": "myadmin",
      "EngineVersion": "5.7.mysql_aurora.2.04.2",
    }
  ]
}
```

```

        "LicenseModel": "aurora-mysql",
        "SnapshotType": "manual",
        "PercentProgress": 100,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:814387698303:cluster-
snapshot:myclustersnapshotcopy",
        "IAMDatabaseAuthenticationEnabled": false
    },
    {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1e"
        ],
        "DBClusterSnapshotIdentifier": "rds:mydbcluster-2019-06-20-09-16",
        "DBClusterIdentifier": "mydbcluster",
        "SnapshotCreateTime": "2019-06-20T09:16:26.569Z",
        "Engine": "aurora-mysql",
        "AllocatedStorage": 0,
        "Status": "available",
        "Port": 0,
        "VpcId": "vpc-6594f31c",
        "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
        "MasterUsername": "myadmin",
        "EngineVersion": "5.7.mysql_aurora.2.04.2",
        "LicenseModel": "aurora-mysql",
        "SnapshotType": "automated",
        "PercentProgress": 100,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:rds:mydbcluster-2019-06-20-09-16",
        "IAMDatabaseAuthenticationEnabled": false
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen eines DB-Cluster-Snapshots](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbClusterSnapshots](#) unter AWS CLI Befehlsreferenz.

describe-db-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-db-clusters`.

AWS CLI

Beispiel 1: Um einen DB-Cluster zu beschreiben

Im folgenden `describe-db-clusters` Beispiel werden die Details des angegebenen DB-Clusters abgerufen.

```
aws rds describe-db-clusters \
  --db-cluster-identifier mydbcluster
```

Ausgabe:

```
{
  "DBClusters": [
    {
      "AllocatedStorage": 1,
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1e"
      ],
      "BackupRetentionPeriod": 1,
      "DatabaseName": "mydbcluster",
      "DBClusterIdentifier": "mydbcluster",
      "DBClusterParameterGroup": "default.aurora-mysql5.7",
      "DBSubnetGroup": "default",
      "Status": "available",
      "EarliestRestorableTime": "2019-06-19T09:16:28.210Z",
      "Endpoint": "mydbcluster.cluster-cnpxample.us-
east-1.rds.amazonaws.com",
      "ReaderEndpoint": "mydbcluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
      "MultiAZ": true,
      "Engine": "aurora-mysql",
      "EngineVersion": "5.7.mysql_aurora.2.04.2",
      "LatestRestorableTime": "2019-06-20T22:38:14.908Z",
      "Port": 3306,
      "MasterUsername": "myadmin",
      "PreferredBackupWindow": "09:09-09:39",
```

```
"PreferredMaintenanceWindow": "sat:04:09-sat:04:39",
"ReadReplicaIdentifiers": [],
"DBClusterMembers": [
  {
    "DBInstanceIdentifier": "dbinstance3",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "dbinstance1",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "dbinstance2",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "mydbcluster",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "mydbcluster-us-east-1b",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "mydbcluster",
    "IsClusterWriter": true,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  }
],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
    "Status": "active"
  }
]
```

```

    }
  ],
  "HostedZoneId": "Z2R2ITUGPM61AM",
  "StorageEncrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
  "DbClusterResourceId": "cluster-AKIAIOSFODNN7EXAMPLE",
  "DBClusterArn": "arn:aws:rds:us-
east-1:123456789012:cluster:mysqlcluster",
  "AssociatedRoles": [],
  "IAMDatabaseAuthenticationEnabled": false,
  "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
  "EngineMode": "provisioned",
  "DeletionProtection": false,
  "HttpEndpointEnabled": false
}
]
}

```

Beispiel 2: Um bestimmte Attribute aller DB-Cluster aufzulisten

Im folgenden `describe-db-clusters` Beispiel werden nur die `DBClusterIdentifier` `Endpoint`, `-` und `ReaderEndpoint` Attribute aller Ihrer DB-Cluster in der aktuellen AWS Region abgerufen.

```

aws rds describe-db-clusters \
  --query 'DBClusters[.
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}]'

```

Ausgabe:

```

[
  {
    "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpxexample.us-
east-1.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpxexample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "cluster-57-2020-05-01-2270"
  },
  {
    "Endpoint": "cluster-57-2020-05-01-4615.cluster-cnpxexample.us-
east-1.rds.amazonaws.com",

```

```

    "ReaderEndpoint": "cluster-57-2020-05-01-4615.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "cluster-57-2020-05-01-4615"
  },
  {
    "Endpoint": "pg2-cluster.cluster-cnpxample.us-east-1.rds.amazonaws.com",
    "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "pg2-cluster"
  },
  ...output omitted...
}
]

```

Beispiel 3: Um DB-Cluster mit einem bestimmten Attribut aufzulisten

Im folgenden `describe-db-clusters` Beispiel werden nur die Engine Attribute `DBClusterIdentifier` und der DB-Cluster abgerufen, die die `aurora-postgresql` DB-Engine verwenden.

```

aws rds describe-db-clusters \
  --query 'DBClusters[].[DBClusterIdentifier:DBClusterIdentifier,Engine:Engine] |
  [?Engine == `aurora-postgresql`]'

```

Ausgabe:

```

[
  {
    "Engine": "aurora-postgresql",
    "DBClusterIdentifier": "pg2-cluster"
  }
]

```

Weitere Informationen finden Sie unter [Amazon Aurora DB Clusters](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbClusters](#) unter AWS CLI Befehlsreferenz.

describe-db-engine-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-db-engine-versions`.

AWS CLI

Um die DB-Engine-Versionen für die MySQL DB-Engine zu beschreiben

Im folgenden `describe-db-engine-versions` Beispiel werden Details zu den einzelnen DB-Engine-Versionen für die angegebene DB-Engine angezeigt.

```
aws rds describe-db-engine-versions \  
  --engine mysql
```

Ausgabe:

```
{  
  "DBEngineVersions": [  
    {  
      "Engine": "mysql",  
      "EngineVersion": "5.5.46",  
      "DBParameterGroupFamily": "mysql5.5",  
      "DBEngineDescription": "MySQL Community Edition",  
      "DBEngineVersionDescription": "MySQL 5.5.46",  
      "ValidUpgradeTarget": [  
        {  
          "Engine": "mysql",  
          "EngineVersion": "5.5.53",  
          "Description": "MySQL 5.5.53",  
          "AutoUpgrade": false,  
          "IsMajorVersionUpgrade": false  
        },  
        {  
          "Engine": "mysql",  
          "EngineVersion": "5.5.54",  
          "Description": "MySQL 5.5.54",  
          "AutoUpgrade": false,  
          "IsMajorVersionUpgrade": false  
        },  
        {  
          "Engine": "mysql",  
          "EngineVersion": "5.5.57",  
          "Description": "MySQL 5.5.57",  
          "AutoUpgrade": false,  
          "IsMajorVersionUpgrade": false  
        },  
        ...some output truncated...  
      ]  
    }  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Was ist Amazon Relational Database Service \(AmazonRDS\)?](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [escribeDBEngineD-Versionen](#) in der AWS CLI Befehlsreferenz.

describe-db-instance-automated-backups

Das folgende Codebeispiel zeigt, wie man `describe-db-instance-automated-backups`.

AWS CLI

Um die automatisierten Backups für eine DB-Instance zu beschreiben

Im folgenden `describe-db-instance-automated-backups` Beispiel werden Details zu den automatisierten Backups für die angegebene DB-Instance angezeigt. Die Details beinhalten replizierte automatisierte Backups in anderen AWS Regionen.

```
aws rds describe-db-instance-automated-backups \
  --db-instance-identifier new-orcl-db
```

Ausgabe:

```
{
  "DBInstanceAutomatedBackups": [
    {
      "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
      "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
      "Region": "us-east-1",
      "DBInstanceIdentifier": "new-orcl-db",
      "RestoreWindow": {
        "EarliestTime": "2020-12-07T21:05:20.939Z",
        "LatestTime": "2020-12-07T21:05:20.939Z"
      },
      "AllocatedStorage": 20,
      "Status": "replicating",
      "Port": 1521,
      "InstanceCreateTime": "2020-12-04T15:28:31Z",
```

```

    "MasterUsername": "admin",
    "Engine": "oracle-se2",
    "EngineVersion": "12.1.0.2.v21",
    "LicenseModel": "bring-your-own-license",
    "OptionGroupName": "default:oracle-se2-12-1",
    "Encrypted": false,
    "StorageType": "gp2",
    "IAMDatabaseAuthenticationEnabled": false,
    "BackupRetentionPeriod": 14,
    "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-
west-2:123456789012:auto-backup:ab-jkib2gfg5rv7replzadausbrktni2bn4example"
  }
]
}

```

Weitere Informationen [finden Sie unter Informationen zu replizierten Backups](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [DescribeDbInstanceAutomatedBackups AWS CLIBefehlsreferenz](#).

describe-db-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-db-instances`.

AWS CLI

Um eine DB-Instance zu beschreiben

Im folgenden `describe-db-instances` Beispiel werden Details zur angegebenen DB-Instance abgerufen.

```

aws rds describe-db-instances \
  --db-instance-identifier mydbinstancecf

```

Ausgabe:

```

{
  "DBInstances": [
    {
      "DBInstanceIdentifier": "mydbinstancecf",
      "DBInstanceClass": "db.t3.small",

```

```

    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "masterawsuser",
    "Endpoint": {
      "Address": "mydbinstancecf.abcxample.us-east-1.rds.amazonaws.com",
      "Port": 3306,
      "HostedZoneId": "Z2R2ITUGPM61AM"
    },
    ...some output truncated...
  }
]
}

```

- API-Einheiten finden Sie unter [D describeDBInstances](#) in der AWS CLI Befehlsreferenz.

describe-db-log-files

Das folgende Codebeispiel zeigt, wie man `describe-db-log-files` verwendet.

AWS CLI

Um die Protokolldateien für eine DB-Instance zu beschreiben

Im folgenden `describe-db-log-files` Beispiel werden Details zu den Protokolldateien für die angegebene DB-Instance abgerufen.

```

aws rds describe-db-log-files -\
  -db-instance-identifizier test-instance

```

Ausgabe:

```

{
  "DescribeDBLogFiles": [
    {
      "Size": 0,
      "LastWritten": 1533060000000,
      "LogFileName": "error/mysql-error-running.log"
    },
    {
      "Size": 2683,
      "LastWritten": 1532994300000,
      "LogFileName": "error/mysql-error-running.log.0"
    }
  ]
}

```



```

    },
    {
      "Size": 107,
      "LastWritten": 1533057300000,
      "LogFileName": "error/mysql-error-running.log.18"
    },
    {
      "Size": 13105,
      "LastWritten": 1532991000000,
      "LogFileName": "error/mysql-error-running.log.23"
    },
    {
      "Size": 0,
      "LastWritten": 1533061200000,
      "LogFileName": "error/mysql-error.log"
    },
    {
      "Size": 3519,
      "LastWritten": 1532989252000,
      "LogFileName": "mysqlUpgrade"
    }
  ]
}

```

- API-Einheiten finden Sie unter [DescribeDbLogFiles AWS CLI Befehlsreferenz](#).

describe-db-parameter-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-parameter-groups`.

AWS CLI

Um Ihre DB-Parametergruppe zu beschreiben

Im folgenden `describe-db-parameter-groups` Beispiel werden Details zu Ihren DB-Parametergruppen abgerufen.

```
aws rds describe-db-parameter-groups
```

Ausgabe:

```
{
```

```

"DBParameterGroups": [
  {
    "DBParameterGroupName": "default.aurora-mysql5.7",
    "DBParameterGroupFamily": "aurora-mysql5.7",
    "Description": "Default parameter group for aurora-mysql5.7",
    "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
  },
  {
    "DBParameterGroupName": "default.aurora-postgresql9.6",
    "DBParameterGroupFamily": "aurora-postgresql9.6",
    "Description": "Default parameter group for aurora-postgresql9.6",
    "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
  },
  {
    "DBParameterGroupName": "default.aurora5.6",
    "DBParameterGroupFamily": "aurora5.6",
    "Description": "Default parameter group for aurora5.6",
    "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
  },
  {
    "DBParameterGroupName": "default.mariadb10.1",
    "DBParameterGroupFamily": "mariadb10.1",
    "Description": "Default parameter group for mariadb10.1",
    "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
  },
  ...some output truncated...
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [describeDBParameterD-Gruppen](#) in der AWS CLI Befehlsreferenz.

describe-db-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-db-parameters`.

AWS CLI

Um die Parameter in einer DB-Parametergruppe zu beschreiben

Im folgenden `describe-db-parameters` Beispiel werden die Details der angegebenen DB-Parametergruppe abgerufen.

```
aws rds describe-db-parameters \  
  --db-parameter-group-name mydbpg
```

Ausgabe:

```
{  
  "Parameters": [  
    {  
      "ParameterName": "allow-suspicious-udfs",  
      "Description": "Controls whether user-defined functions that have only  
an xxx symbol for the main function can be loaded",  
      "Source": "engine-default",  
      "ApplyType": "static",  
      "DataType": "boolean",  
      "AllowedValues": "0,1",  
      "IsModifiable": false,  
      "ApplyMethod": "pending-reboot"  
    },  
    {  
      "ParameterName": "auto_generate_certs",  
      "Description": "Controls whether the server autogenerates SSL key and  
certificate files in the data directory, if they do not already exist.",  
      "Source": "engine-default",  
      "ApplyType": "static",  
      "DataType": "boolean",  
      "AllowedValues": "0,1",  
      "IsModifiable": false,  
      "ApplyMethod": "pending-reboot"  
    },  
    ...some output truncated...  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeDBParameters](#) in der AWS CLI Befehlsreferenz.

describe-db-proxies

Das folgende Codebeispiel zeigt, wie man `describe-db-proxies` verwendet.

AWS CLI

Um einen DB-Proxy für eine RDS-Datenbank zu beschreiben

Das folgende `describe-db-proxies` Beispiel gibt Informationen über DB-Proxys zurück.

```
aws rds describe-db-proxies
```

Ausgabe:

```
{
  "DBProxies": [
    {
      "DBProxyName": "proxyExample1",
      "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:prx-0123a01b12345c0ab",
      "Status": "available",
      "EngineFamily": "PostgreSQL",
      "VpcId": "vpc-1234567",
      "VpcSecurityGroupIds": [
        "sg-1234"
      ],
      "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
      ],
      "Auth": "[
        {
          "Description": "proxydescription1",
          "AuthScheme": "SECRETS",
          "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789123:secret:secretName-1234f",
          "IAMAuth": "DISABLED"
        }
      ]",
      "RoleArn": "arn:aws:iam::12345678912?:role/ProxyPostgreSQLRole",
    }
  ]
}
```

```

        "Endpoint": "proxyExample1.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "RequireTLS": false,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
        "CreateDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    },
    {
        "DBProxyName": "proxyExample2",
        "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-1234a12b23456c1ab",
        "Status": "available",
        "EngineFamily": "PostgreSQL",
        "VpcId": "sg-1234567",
        "VpcSecurityGroupIds": [
            "sg-1234"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Auth": "[
            {
                "Description": "proxydescription2"
                "AuthScheme": "SECRETS",
                "SecretArn": "aarn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
                "IAMAuth": "DISABLED"
            }
        ]",
        "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
        "Endpoint": "proxyExample2.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
        "RequireTLS": false,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
        "CreateDate": "2022-01-05T16:19:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
]
}

```

Weitere Informationen finden Sie unter [Einen RDS Proxy anzeigen](#) im RDSAmazon-Benutzerhandbuch und [Einen RDS Proxy anzeigen](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbProxies](#) in der AWS CLI Befehlsreferenz.

describe-db-proxy-endpoints

Das folgende Codebeispiel zeigt die Verwendung `describe-db-proxy-endpoints`.

AWS CLI

Um einen DB-Proxy-Endpunkt zu beschreiben

Das folgende `describe-db-proxy-endpoints` Beispiel gibt Informationen über DB-Proxy-Endpunkte zurück.

```
aws rds describe-db-proxy-endpoints
```

Ausgabe:

```
{
  "DBProxyEndpoints": [
    {
      "DBProxyEndpointName": "proxyEndpoint1",
      "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
      "DBProxyName": "proxyExample",
      "Status": "available",
      "VpcId": "vpc-1234567",
      "VpcSecurityGroupIds": [
        "sg-1234"
      ],
      "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
      ],
      "Endpoint": "proxyEndpoint1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
      "CreateDate": "2023-04-05T16:09:33.452000+00:00",
      "TargetRole": "READ_WRITE",
      "IsDefault": false
    },
    {
```

```

        "DBProxyEndpointName": "proxyEndpoint2",
        "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-4567a01b12345c0ab",
        "DBProxyName": "proxyExample2",
        "Status": "available",
        "VpcId": "vpc1234567",
        "VpcSecurityGroupIds": [
            "sg-5678"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Endpoint": "proxyEndpoint2.endpoint.proxy-cd1ef2klmnop.us-
east-1.rds.amazonaws.com",
        "CreateDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": false
    }
]
}

```

Weitere Informationen finden Sie unter [Anzeigen eines Proxy-Endpunkts](#) im RDSAmazon-Benutzerhandbuch und [Erstellen eines Proxy-Endpunkts](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbProxyEndpoints](#) in der AWS CLI Befehlsreferenz.

describe-db-proxy-target-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-proxy-target-groups`.

AWS CLI

Um einen DB-Proxy-Endpunkt zu beschreiben

Das folgende `describe-db-proxy-target-groups` Beispiel gibt Informationen über DB-Proxy-Zielgruppen zurück.

```

aws rds describe-db-proxy-target-groups \
  --db-proxy-name proxyExample

```

Ausgabe:

```
{
  "TargetGroups":
    {
      "DBProxyName": "proxyExample",
      "TargetGroupName": "default",
      "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
      "IsDefault": true,
      "Status": "available",
      "ConnectionPoolConfig": {
        "MaxConnectionsPercent": 100,
        "MaxIdleConnectionsPercent": 50,
        "ConnectionBorrowTimeout": 120,
        "SessionPinningFilters": []
      },
      "CreateDate": "2023-05-02T18:41:19.495000+00:00",
      "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

Weitere Informationen finden Sie unter [Einen RDS Proxy anzeigen](#) im RDSAmazon-Benutzerhandbuch und [Einen RDS Proxy anzeigen](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbProxyTargetGroups](#) in der AWS CLI Befehlsreferenz.

describe-db-proxy-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-db-proxy-targets`.

AWS CLI

Um DB-Proxyziele zu beschreiben

Das folgende `describe-db-proxy-targets` Beispiel gibt Informationen über DB-Proxyziele zurück.

```
aws rds describe-db-proxy-targets \
  --db-proxy-name proxyExample
```

Ausgabe:

```
{
```



```
"Targets": [
  {
    "Endpoint": "database1.ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "TrackedClusterId": "database1",
    "RdsResourceId": "database1-instance-1",
    "Port": 3306,
    "Type": "RDS_INSTANCE",
    "Role": "READ_WRITE",
    "TargetHealth": {
      "State": "UNAVAILABLE",
      "Reason": "PENDING_PROXY_CAPACITY",
      "Description": "DBProxy Target is waiting for proxy to scale to
desired capacity"
    }
  }
]
```

Weitere Informationen finden Sie unter [Einen RDS Proxy anzeigen](#) im RDSAmazon-Benutzerhandbuch und [Einen RDS Proxy anzeigen](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbProxyTargets](#) in der AWS CLI Befehlsreferenz.

describe-db-recommendations

Das folgende Codebeispiel zeigt die Verwendung `describe-db-recommendations`.

AWS CLI

Beispiel 1: Um alle DB-Empfehlungen aufzulisten

Das folgende `describe-db-recommendations` Beispiel listet alle DB-Empfehlungen in Ihrem AWS Konto auf.

```
aws rds describe-db-recommendations
```

Ausgabe:

```
{
  "DBRecommendations": [
    {
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
      "TypeId": "config_recommendation::old_minor_version",
```

```
"Severity": "informational",
"ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
"Status": "active",
"CreatedTime": "2024-02-21T23:14:19.292000+00:00",
"UpdatedTime": "2024-02-21T23:14:19+00:00",
"Detection": "***[resource-name]** is not running the latest minor DB
engine version",
"Recommendation": "Upgrade to latest engine version",
>Description": "Your database resources aren't running the latest minor
DB engine version. The latest minor version contains the latest security fixes and
other improvements.",
"RecommendedActions": [
  {
    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
    "Operation": "modifyDbInstance",
    "Parameters": [
      {
        "Key": "EngineVersion",
        "Value": "5.7.44"
      },
      {
        "Key": "DBInstanceIdentifier",
        "Value": "database-1"
      }
    ],
    "ApplyModes": [
      "immediately",
      "next-maintenance-window"
    ],
    "Status": "ready",
    "ContextAttributes": [
      {
        "Key": "Recommended value",
        "Value": "5.7.44"
      },
      {
        "Key": "Current engine version",
        "Value": "5.7.42"
      }
    ]
  }
],
"Category": "security",
"Source": "RDS",
```

```

        "TypeDetection": "***[resource-count] resources** are not running the
latest minor DB engine version",
        "TypeRecommendation": "Upgrade to latest engine version",
        "Impact": "Reduced database performance and data security at risk",
        "AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
        "Links": [
            {
                "Text": "Upgrading an RDS DB instance engine version",
                "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
            },
            {
                "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon Aurora",
                "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
            },
            {
                "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon RDS",
                "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
            }
        ]
    }
]
}

```

Weitere Informationen finden Sie unter [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im RDSAmazon-Benutzerhandbuch und [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im Amazon Aurora-Benutzerhandbuch.

Beispiel 2: Um DB-Empfehlungen mit hohem Schweregrad aufzulisten

Im folgenden `describe-db-recommendations` Beispiel werden DB-Empfehlungen mit hohem Schweregrad in Ihrem AWS Konto aufgeführt.

```

aws rds describe-db-recommendations \
  --filters Name=severity,Values=high

```

Ausgabe:

```
{
  "DBRecommendations": [
    {
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
      "TypeId": "config_recommendation::rds_extended_support",
      "Severity": "high",
      "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
      "Status": "active",
      "CreatedTime": "2024-02-21T23:14:19.392000+00:00",
      "UpdatedTime": "2024-02-21T23:14:19+00:00",
      "Detection": "Your databases will be auto-enrolled to RDS Extended
Support on February 29",
      "Recommendation": "Upgrade your major version before February 29, 2024
to avoid additional charges",
      "Description": "Your PostgreSQL 11 and MySQL 5.7 databases will be
automatically enrolled into RDS Extended Support on February 29, 2024. To avoid
the increase in charges due to RDS Extended Support, we recommend upgrading your
databases to a newer major engine version before February 29, 2024.\nTo learn more
about the RDS Extended Support pricing, refer to the pricing page.",
      "RecommendedActions": [
        {
          "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
          "Parameters": [],
          "ApplyModes": [
            "manual"
          ],
          "Status": "ready",
          "ContextAttributes": []
        }
      ],
      "Category": "cost optimization",
      "Source": "RDS",
      "TypeDetection": "Your database will be auto-enrolled to RDS Extended
Support on February 29",
      "TypeRecommendation": "Upgrade your major version before February 29,
2024 to avoid additional charges",
      "Impact": "Increase in charges due to RDS Extended Support",
      "AdditionalInfo": "With Amazon RDS Extended Support, you can continue
running your database on a major engine version past the RDS end of standard
support date for an additional cost. This paid feature gives you more time to
upgrade to a supported major engine version.\nDuring Extended Support, Amazon RDS
will supply critical CVE patches and bug fixes.",
    }
  ]
}
```

```

    "Links": [
      {
        "Text": "Amazon RDS Extended Support pricing for RDS for MySQL",
        "Url": "https://aws.amazon.com/rds/mysql/pricing/"
      },
      {
        "Text": "Amazon RDS Extended Support for RDS for MySQL and
PostgreSQL databases",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
extended-support.html"
      },
      {
        "Text": "Amazon RDS Extended Support pricing for Amazon Aurora
PostgreSQL",
        "Url": "https://aws.amazon.com/rds/aurora/pricing/"
      },
      {
        "Text": "Amazon RDS Extended Support for Aurora PostgreSQL
databases",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/extended-support.html"
      },
      {
        "Text": "Amazon RDS Extended Support pricing for RDS for
PostgreSQL",
        "Url": "https://aws.amazon.com/rds/postgresql/pricing/"
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im RDSAmazon-Benutzerhandbuch und [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im Amazon Aurora-Benutzerhandbuch.

Beispiel 3: Um DB-Empfehlungen für eine bestimmte DB-Instance aufzulisten

Das folgende `describe-db-recommendations` Beispiel listet alle DB-Empfehlungen für eine angegebene DB-Instance auf.

```

aws rds describe-db-recommendations \
  --filters Name=dbi-resource-id,Values=database-1

```

Ausgabe:

```
{
  "DBRecommendations": [
    {
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
      "TypeId": "config_recommendation::old_minor_version",
      "Severity": "informational",
      "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
      "Status": "active",
      "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
      "UpdatedTime": "2024-02-21T23:14:19+00:00",
      "Detection": "***[resource-name]** is not running the latest minor DB
engine version",
      "Recommendation": "Upgrade to latest engine version",
      "Description": "Your database resources aren't running the latest minor
DB engine version. The latest minor version contains the latest security fixes and
other improvements.",
      "RecommendedActions": [
        {
          "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
          "Operation": "modifyDbInstance",
          "Parameters": [
            {
              "Key": "EngineVersion",
              "Value": "5.7.44"
            },
            {
              "Key": "DBInstanceIdentifier",
              "Value": "database-1"
            }
          ],
          "ApplyModes": [
            "immediately",
            "next-maintenance-window"
          ],
          "Status": "ready",
          "ContextAttributes": [
            {
              "Key": "Recommended value",
              "Value": "5.7.44"
            },
            {
              "Key": "Current engine version",
```

```

        "Value": "5.7.42"
      }
    ]
  },
  "Category": "security",
  "Source": "RDS",
  "TypeDetection": "***[resource-count] resources** are not running the
latest minor DB engine version",
  "TypeRecommendation": "Upgrade to latest engine version",
  "Impact": "Reduced database performance and data security at risk",
  "AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
  "Links": [
    {
      "Text": "Upgrading an RDS DB instance engine version",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
    },
    {
      "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon Aurora",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
    },
    {
      "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon RDS",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
    }
  ]
}
]
}

```

Weitere Informationen finden Sie unter [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im RDSAmazon-Benutzerhandbuch und [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im Amazon Aurora-Benutzerhandbuch.

Beispiel 4: Um alle aktiven DB-Empfehlungen aufzulisten

Das folgende `describe-db-recommendations` Beispiel listet alle aktiven DB-Empfehlungen in Ihrem AWS Konto auf.

```
aws rds describe-db-recommendations \  
  --filters Name=status,Values=active
```

Ausgabe:

```
{  
  "DBRecommendations": [  
    {  
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",  
      "TypeId": "config_recommendation::old_minor_version",  
      "Severity": "informational",  
      "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",  
      "Status": "active",  
      "CreatedTime": "2024-02-21T23:14:19.292000+00:00",  
      "UpdatedTime": "2024-02-21T23:14:19+00:00",  
      "Detection": "***[resource-name]** is not running the latest minor DB  
engine version",  
      "Recommendation": "Upgrade to latest engine version",  
      "Description": "Your database resources aren't running the latest minor  
DB engine version. The latest minor version contains the latest security fixes and  
other improvements.",  
      "RecommendedActions": [  
        {  
          "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",  
          "Operation": "modifyDbInstance",  
          "Parameters": [  
            {  
              "Key": "EngineVersion",  
              "Value": "5.7.44"  
            },  
            {  
              "Key": "DBInstanceIdentifier",  
              "Value": "database-1"  
            }  
          ],  
          "ApplyModes": [  
            "immediately",  
            "next-maintenance-window"  
          ]  
        }  
      ]  
    }  
  ]  
}
```



```
    ],
    "Status": "ready",
    "ContextAttributes": [
      {
        "Key": "Recommended value",
        "Value": "5.7.44"
      },
      {
        "Key": "Current engine version",
        "Value": "5.7.42"
      }
    ]
  }
],
"Category": "security",
"Source": "RDS",
"TypeDetection": "**[resource-count] resources** are not running the
latest minor DB engine version",
"TypeRecommendation": "Upgrade to latest engine version",
"Impact": "Reduced database performance and data security at risk",
"AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
"Links": [
  {
    "Text": "Upgrading an RDS DB instance engine version",
    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
  },
  {
    "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon Aurora",
    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
  },
  {
    "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon RDS",
    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
  }
]
]
```

```
    }
  ]
}
```

Weitere Informationen finden Sie unter [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im RDSAmazon-Benutzerhandbuch und [RDSAmazon-Empfehlungen anzeigen und beantworten](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbRecommendations](#) in der AWS CLI Befehlsreferenz.

describe-db-security-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-security-groups`.

AWS CLI

Um DB-Sicherheitsgruppen aufzulisten

Das folgende `describe-db-security-groups` Beispiel listet DB-Sicherheitsgruppen auf.

```
aws rds describe-db-security-groups
```

Ausgabe:

```
{
  "DBSecurityGroups": [
    {
      "OwnerId": "123456789012",
      "DBSecurityGroupName": "default",
      "DBSecurityGroupDescription": "default",
      "EC2SecurityGroups": [],
      "IPRanges": [],
      "DBSecurityGroupArn": "arn:aws:rds:us-west-1:111122223333:secgrp:default"
    },
    {
      "OwnerId": "123456789012",
      "DBSecurityGroupName": "mysecgroup",
      "DBSecurityGroupDescription": "My Test Security Group",
      "VpcId": "vpc-1234567f",
      "EC2SecurityGroups": [],
      "IPRanges": [],
    }
  ]
}
```

```
        "DBSecurityGroupArn": "arn:aws:rds:us-  
west-1:111122223333:secgrp:mysecgroup"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Verfügbare DB-Sicherheitsgruppen auflisten](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeDbSecurityGroups](#) in der AWS CLI Befehlsreferenz.

describe-db-shard-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-db-shard-groups`.

AWS CLI

Beispiel 1: Um DB-Shard-Gruppen zu beschreiben

Im folgenden `describe-db-shard-groups` Beispiel werden die Details Ihrer DB-Shard-Gruppen abgerufen.

```
aws rds describe-db-shard-groups
```

Ausgabe:

```
{  
  "DBShardGroups": [  
    {  
      "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",  
      "DBShardGroupIdentifier": "limitless-test-shard-grp",  
      "DBClusterIdentifier": "limitless-test-cluster",  
      "MaxACU": 768.0,  
      "ComputeRedundancy": 0,  
      "Status": "available",  
      "PubliclyAccessible": true,  
      "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-  
east-2.rds.amazonaws.com"  
    },  
    {  
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",  
      "DBShardGroupIdentifier": "my-db-shard-group",  

```

```

        "DBClusterIdentifier": "my-sv2-cluster",
        "MaxACU": 768.0,
        "ComputeRedundancy": 0,
        "Status": "available",
        "PubliclyAccessible": false,
        "Endpoint": "my-sv2-cluster.limitless-cekyceexample.us-
east-2.rds.amazonaws.com"
    }
]
}

```

Weitere Informationen finden Sie unter [Amazon Aurora DB Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbShardGroups](#) unter AWS CLI Befehlsreferenz.

describe-db-snapshot-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-db-snapshot-attributes`.

AWS CLI

Um die Attributnamen und -werte für einen DB-Snapshot zu beschreiben

Das folgende `describe-db-snapshot-attributes` Beispiel beschreibt die Attributnamen und -werte für einen DB-Snapshot.

```

aws rds describe-db-snapshot-attributes \
  --db-snapshot-identifier mydbsnapshot

```

Ausgabe:

```

{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012",
          "210987654321"
        ]
      }
    ]
  }
}

```

```
    ]
  }
}
```

Weitere Informationen finden Sie unter [Einen DB-Snapshot teilen](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeDbSnapshotAttributes](#) in der AWS CLI Befehlsreferenz.

describe-db-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-db-snapshots`.

AWS CLI

Beispiel 1: Um einen DB-Snapshot für eine DB-Instance zu beschreiben

Im folgenden `describe-db-snapshots` Beispiel werden die Details eines DB-Snapshots für eine DB-Instance abgerufen.

```
aws rds describe-db-snapshots \
  --db-snapshot-identifizier mydbsnapshot
```

Ausgabe:

```
{
  "DBSnapshots": [
    {
      "DBSnapshotIdentifizier": "mydbsnapshot",
      "DBInstanceIdentifizier": "mysqldb",
      "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
      "Engine": "mysql",
      "AllocatedStorage": 20,
      "Status": "available",
      "Port": 3306,
      "AvailabilityZone": "us-east-1f",
      "VpcId": "vpc-6594f31c",
      "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
      "MasterUsername": "mysqladmin",
      "EngineVersion": "5.6.37",
      "LicenseModel": "general-public-license",
      "SnapshotType": "manual",
      "OptionGroupName": "default:mysql-5-6",
```

```

        "PercentProgress": 100,
        "StorageType": "gp2",
        "Encrypted": false,
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
]
}

```

Weitere Informationen finden Sie unter [Erstellen eines DB-Snapshots](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: Um die Anzahl der manuell erstellten Snapshots zu ermitteln

Im folgenden `describe-db-snapshots` Beispiel wird der `length` Operator in der `--query` Option verwendet, um die Anzahl der manuellen Schnappschüsse zurückzugeben, die in einer bestimmten AWS Region aufgenommen wurden.

```

aws rds describe-db-snapshots \
  --snapshot-type manual \
  --query "length(*[].[DBSnapshots:SnapshotType])" \
  --region eu-central-1

```

Ausgabe:

```
35
```

Weitere Informationen finden Sie unter [Erstellen eines DB-Snapshots](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [D describeDBSnapshots](#) in der AWS CLI Befehlsreferenz.

describe-db-subnet-groups

Das folgende Codebeispiel zeigt, wie man `describe-db-subnet-groups`.

AWS CLI

Um eine DB-Subnetzgruppe zu beschreiben

Im folgenden `describe-db-subnet-groups` Beispiel werden die Details der angegebenen DB-Subnetzgruppe abgerufen.

```
aws rds describe-db-subnet-groups
```

Ausgabe:

```
{
  "DBSubnetGroups": [
    {
      "DBSubnetGroupName": "mydbsubnetgroup",
      "DBSubnetGroupDescription": "My DB Subnet Group",
      "VpcId": "vpc-971c12ee",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-d8c8e7f4",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-718fdc7d",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-cbc8e7e7",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-0ccde220",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        }
      ]
    }
  ]
}
```

```

    ],
    "DBSubnetGroupArn": "arn:aws:rds:us-
east-1:123456789012:subgrp:mydbsubnetgroup"
  }
]
}

```

Weitere Informationen finden Sie unter [Amazon Virtual Private Cloud VPCs und Amazon RDS](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDbSubnetGroups](#) in der AWS CLI Befehlsreferenz.

describe-engine-default-cluster-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-engine-default-cluster-parameters`.

AWS CLI

Um die Standard-Engine und die Systemparameterinformationen für die Aurora-Datenbank-Engine zu beschreiben

Im folgenden `describe-engine-default-cluster-parameters` Beispiel werden die Details der Standard-Engine und die Systemparameterinformationen für Aurora-DB-Cluster mit MySQL 5.7-Kompatibilität abgerufen.

```

aws rds describe-engine-default-cluster-parameters \
  --db-parameter-group-family aurora-mysql5.7

```

Ausgabe:

```

{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "aurora_load_from_s3_role",
        "Description": "IAM role ARN used to load data from AWS S3",
        "Source": "engine-default",
        "ApplyType": "dynamic",
        "DataType": "string",
        "IsModifiable": true,
        "SupportedEngineModes": [

```



```

        "provisioned"
      ]
    },
    ...some output truncated...
  ]
}
}

```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEngineDefaultClusterParameters](#) unter AWS CLI Befehlsreferenz.

describe-engine-default-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-engine-default-parameters`.

AWS CLI

Um die Standard-Engine und die Systemparameterinformationen für die Datenbank-Engine zu beschreiben

Im folgenden `describe-engine-default-parameters` Beispiel werden Details für die Standard-Engine und Systemparameterinformationen für MySQL 5.7 DB-Instances abgerufen.

```

aws rds describe-engine-default-parameters \
  --db-parameter-group-family mysql5.7

```

Ausgabe:

```

{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "allow-suspicious-udfs",
        "Description": "Controls whether user-defined functions that have
only an xxx symbol for the main function can be loaded",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",

```

```

        "IsModifiable": false
      },
      ...some output truncated...
    ]
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeEngineDefaultParameters](#) unter AWS CLI Befehlsreferenz.

describe-event-categories

Das folgende Codebeispiel zeigt die Verwendung `describe-event-categories`.

AWS CLI

Um Ereigniskategorien zu beschreiben

Im folgenden `describe-event-categories` Beispiel werden Details zu den Ereigniskategorien für alle verfügbaren Ereignisquellen abgerufen.

```
aws rds describe-event-categories
```

Ausgabe:

```

{
  "EventCategoriesMapList": [
    {
      "SourceType": "db-instance",
      "EventCategories": [
        "deletion",
        "read replica",
        "failover",
        "restoration",
        "maintenance",
        "low storage",
        "configuration change",
        "backup",
        "creation",
        "availability",
        "recovery",

```

```
        "failure",
        "backtrack",
        "notification"
    ]
},
{
    "SourceType": "db-security-group",
    "EventCategories": [
        "configuration change",
        "failure"
    ]
},
{
    "SourceType": "db-parameter-group",
    "EventCategories": [
        "configuration change"
    ]
},
{
    "SourceType": "db-snapshot",
    "EventCategories": [
        "deletion",
        "creation",
        "restoration",
        "notification"
    ]
},
{
    "SourceType": "db-cluster",
    "EventCategories": [
        "failover",
        "failure",
        "notification"
    ]
},
{
    "SourceType": "db-cluster-snapshot",
    "EventCategories": [
        "backup"
    ]
}
]
```

- API-Einheiten finden Sie unter [DescribeEventCategories](#) AWS CLI-Befehlsreferenz.

describe-event-subscriptions

Das folgende Codebeispiel zeigt die Verwendung `describe-event-subscriptions`.

AWS CLI

Um Veranstaltungsabonnements zu beschreiben

Dieses Beispiel beschreibt alle RDS Amazon-Event-Abonnements für das AWS Girokonto.

```
aws rds describe-event-subscriptions
```

Ausgabe:

```
{
  "EventSubscriptionsList": [
    {
      "EventCategoriesList": [
        "backup",
        "recovery"
      ],
      "Enabled": true,
      "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-
instance-events",
      "Status": "creating",
      "SourceType": "db-instance",
      "CustomerAwsId": "123456789012",
      "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
      "CustSubscriptionId": "my-instance-events",
      "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events"
    },
    ...some output truncated...
  ]
}
```

- API-Einheiten finden Sie [DescribeEventSubscriptions](#) in der AWS CLI Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Um Ereignisse zu beschreiben

Im folgenden `describe-events` Beispiel werden Details zu den Ereignissen abgerufen, die für die angegebene DB-Instance aufgetreten sind.

```
aws rds describe-events \  
  --source-identifier test-instance \  
  --source-type db-instance
```

Ausgabe:

```
{  
  "Events": [  
    {  
      "SourceType": "db-instance",  
      "SourceIdentifier": "test-instance",  
      "EventCategories": [  
        "backup"  
      ],  
      "Message": "Backing up DB instance",  
      "Date": "2018-07-31T23:09:23.983Z",  
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"  
    },  
    {  
      "SourceType": "db-instance",  
      "SourceIdentifier": "test-instance",  
      "EventCategories": [  
        "backup"  
      ],  
      "Message": "Finished DB Instance backup",  
      "Date": "2018-07-31T23:15:13.049Z",  
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [DescribeEvents AWS CLI Befehlsreferenz](#).

describe-export-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-export-tasks`.

AWS CLI

Um Aufgaben zum Exportieren von Snapshots zu beschreiben

Das folgende `describe-export-tasks` Beispiel gibt Informationen über Snapshot-Exporte nach Amazon S3 zurück.

```
aws rds describe-export-tasks
```

Ausgabe:

```
{
  "ExportTasks": [
    {
      "ExportTaskIdentifier": "test-snapshot-export",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:test-
snapshot",
      "SnapshotTime": "2020-03-02T18:26:28.163Z",
      "TaskStartTime": "2020-03-02T18:57:56.896Z",
      "TaskEndTime": "2020-03-02T19:10:31.985Z",
      "S3Bucket": "mybucket",
      "S3Prefix": "",
      "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
      "Status": "COMPLETE",
      "PercentProgress": 100,
      "TotalExtractedDataInGB": 0
    },
    {
      "ExportTaskIdentifier": "my-s3-export",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
test",
      "SnapshotTime": "2020-03-27T20:48:42.023Z",
      "S3Bucket": "mybucket",
      "S3Prefix": "",
      "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
```

```
        "Status": "STARTING",
        "PercentProgress": 0,
        "TotalExtractedDataInGB": 0
    }
]
}
```

Weitere Informationen finden Sie unter [Überwachung von Snapshot-Exporten](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeExportTasks](#) in der AWS CLI Befehlsreferenz.

describe-global-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-global-clusters`.

AWS CLI

Um globale DB-Cluster zu beschreiben

Das folgende `describe-global-clusters` Beispiel listet globale Aurora-DB-Cluster in der aktuellen AWS Region auf.

```
aws rds describe-global-clusters
```

Ausgabe:

```
{
  "GlobalClusters": [
    {
      "GlobalClusterIdentifier": "myglobalcluster",
      "GlobalClusterResourceId": "cluster-f5982077e3b5aabb",
      "GlobalClusterArn": "arn:aws:rds::123456789012:global-cluster:myglobalcluster",
      "Status": "available",
      "Engine": "aurora-mysql",
      "EngineVersion": "5.7.mysql_aurora.2.07.2",
      "StorageEncrypted": false,
      "DeletionProtection": false,
      "GlobalClusterMembers": []
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Verwaltung einer globalen Aurora-Datenbank](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeGlobalClusters](#) unter AWS CLI Befehlsreferenz.

describe-option-group-options

Das folgende Codebeispiel zeigt die Verwendung `describe-option-group-options`.

AWS CLI

Um alle verfügbaren Optionen zu beschreiben

Das folgende `describe-option-group-options` Beispiel listet zwei Optionen für eine Oracle Database 19c-Instance auf.

```
aws rds describe-option-group-options \
  --engine-name oracle-ee \
  --major-engine-version 19 \
  --max-items 2
```

Ausgabe:

```
{
  "OptionGroupOptions": [
    {
      "Name": "APEX",
      "Description": "Oracle Application Express Runtime Environment",
      "EngineName": "oracle-ee",
      "MajorEngineVersion": "19",
      "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
      "PortRequired": false,
      "OptionsDependedOn": [],
      "OptionsConflictsWith": [],
      "Persistent": false,
      "Permanent": false,
      "RequiresAutoMinorEngineVersionUpgrade": false,
      "VpcOnly": false,
      "SupportsOptionVersionDowngrade": false,
      "OptionGroupOptionSettings": [],
      "OptionGroupOptionVersions": [
```



```

        {
            "Version": "19.1.v1",
            "IsDefault": true
        },
        {
            "Version": "19.2.v1",
            "IsDefault": false
        }
    ]
},
{
    "Name": "APEX-DEV",
    "Description": "Oracle Application Express Development Environment",
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "19",
    "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
    "PortRequired": false,
    "OptionsDependedOn": [
        "APEX"
    ],
    "OptionsConflictsWith": [],
    "Persistent": false,
    "Permanent": false,
    "RequiresAutoMinorEngineVersionUpgrade": false,
    "VpcOnly": false,
    "OptionGroupOptionSettings": []
}
],
"NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

Weitere Informationen finden Sie unter [Auflisten der Optionen und Optionseinstellungen für eine Optionsgruppe](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeOptionGroupOptions](#) in der AWS CLI Befehlsreferenz.

describe-option-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-option-groups`.

AWS CLI

Um die verfügbaren Optionsgruppen zu beschreiben

Das folgende `describe-option-groups` Beispiel listet die Optionsgruppen für eine Oracle Database 19c-Instance auf.

```
aws rds describe-option-groups \  
  --engine-name oracle-ee \  
  --major-engine-version 19
```

Ausgabe:

```
{  
  "OptionGroupsList": [  
    {  
      "OptionGroupName": "default:oracle-ee-19",  
      "OptionGroupDescription": "Default option group for oracle-ee 19",  
      "EngineName": "oracle-ee",  
      "MajorEngineVersion": "19",  
      "Options": [],  
      "AllowsVpcAndNonVpcInstanceMemberships": true,  
      "OptionGroupArn": "arn:aws:rds:us-west-1:111122223333:og:default:oracle-  
ee-19"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Auflisten der Optionen und Optionseinstellungen für eine Optionsgruppe](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeOptionGroups](#) in der AWS CLI Befehlsreferenz.

describe-orderable-db-instance-options

Das folgende Codebeispiel zeigt die Verwendung `describe-orderable-db-instance-options`.

AWS CLI

Um bestellbare DB-Instance-Optionen zu beschreiben

Im folgenden `describe-orderable-db-instance-options` Beispiel werden Details zu den bestellbaren Optionen für DB-Instances abgerufen, auf denen die My SQL DB-Engine ausgeführt wird.

```
aws rds describe-orderable-db-instance-options \  

```

```
--engine mysql
```

Ausgabe:

```
{
  "OrderableDBInstanceOptions": [
    {
      "MinStorageSize": 5,
      "ReadReplicaCapable": true,
      "MaxStorageSize": 6144,
      "AvailabilityZones": [
        {
          "Name": "us-east-1a"
        },
        {
          "Name": "us-east-1b"
        },
        {
          "Name": "us-east-1c"
        },
        {
          "Name": "us-east-1d"
        }
      ],
      "SupportsIops": false,
      "AvailableProcessorFeatures": [],
      "MultiAZCapable": true,
      "DBInstanceClass": "db.m1.large",
      "Vpc": true,
      "StorageType": "gp2",
      "LicenseModel": "general-public-license",
      "EngineVersion": "5.5.46",
      "SupportsStorageEncryption": false,
      "SupportsEnhancedMonitoring": true,
      "Engine": "mysql",
      "SupportsIAMDatabaseAuthentication": false,
      "SupportsPerformanceInsights": false
    }
  ]
  ...some output truncated...
}
```

- API-Einheiten finden Sie unter [DescribeOrderableDBInstanceOptions AWS CLI](#) Befehlsreferenz.

describe-pending-maintenance-actions

Das folgende Codebeispiel zeigt die Verwendung `describe-pending-maintenance-actions`.

AWS CLI

Um Ressourcen aufzulisten, für die mindestens eine Wartungsaktion aussteht

Das folgende `describe-pending-maintenance-actions` Beispiel listet die ausstehende Wartungsaktion für eine DB-Instance auf.

```
aws rds describe-pending-maintenance-actions
```

Ausgabe:

```
{
  "PendingMaintenanceActions": [
    {
      "ResourceIdentifier": "arn:aws:rds:us-west-2:123456789012:cluster:global-db1-cl1",
      "PendingMaintenanceActionDetails": [
        {
          "Action": "system-update",
          "Description": "Upgrade to Aurora PostgreSQL 2.4.2"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Wartung einer DB-Instance](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribePendingMaintenanceActions](#) unter AWS CLI Befehlsreferenz.

describe-reserved-db-instances-offerings

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-db-instances-offerings`.

AWS CLI

Um Angebote für reservierte DB-Instances zu beschreiben

Im folgenden `describe-reserved-db-instances-offerings` Beispiel werden Details zu den Optionen für `oracle` reservierte DB-Instances abgerufen.

```
aws rds describe-reserved-db-instances-offerings \
  --product-description oracle
```

Ausgabe:

```
{
  "ReservedDBInstancesOfferings": [
    {
      "CurrencyCode": "USD",
      "UsagePrice": 0.0,
      "ProductDescription": "oracle-se2(li)",
      "ReservedDBInstancesOfferingId": "005bdee3-9ef4-4182-aa0c-58ef7cb6c2f8",
      "MultiAZ": true,
      "DBInstanceClass": "db.m4.xlarge",
      "OfferingType": "Partial Upfront",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.594,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "FixedPrice": 4089.0,
      "Duration": 31536000
    },
    ...some output truncated...
  ]
}
```

- API-Einheiten finden Sie unter [DescribeReservedDbInstancesOfferings AWS CLI-Befehlsreferenz](#).

describe-reserved-db-instances

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-db-instances`.

AWS CLI

Um reservierte DB-Instances zu beschreiben

Im folgenden `describe-reserved-db-instances` Beispiel werden Details zu allen reservierten DB-Instances im aktuellen AWS Konto abgerufen.

```
aws rds describe-reserved-db-instances
```

Ausgabe:

```
{
  "ReservedDBInstances": [
    {
      "ReservedDBInstanceId": "myreservedinstance",
      "ReservedDBInstancesOfferingId": "12ab34cd-59af-4b2c-a660-1abcdef23456",
      "DBInstanceClass": "db.t3.micro",
      "StartTime": "2020-06-01T13:44:21.436Z",
      "Duration": 31536000,
      "FixedPrice": 0.0,
      "UsagePrice": 0.0,
      "CurrencyCode": "USD",
      "DBInstanceCount": 1,
      "ProductDescription": "sqlserver-ex(li)",
      "OfferingType": "No Upfront",
      "MultiAZ": false,
      "State": "payment-pending",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.014,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "ReservedDBInstanceArn": "arn:aws:rds:us-west-2:123456789012:ri:myreservedinstance",
      "LeaseId": "a1b2c3d4-6b69-4a59-be89-5e11aa446666"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Reserved DB Instances for Amazon RDS](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeReservedDbInstances](#) unter AWS CLI Befehlsreferenz.

describe-source-regions

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-source-regions`.

AWS CLI

Um Quellregionen zu beschreiben

Im folgenden `describe-source-regions` Beispiel werden Details zu allen AWS Quellregionen abgerufen. Es zeigt auch, dass automatische Backups nur von USA West (Oregon) in die AWS Zielregion USA Ost (Nord-Virginia) repliziert werden können.

```
aws rds describe-source-regions \  
  --region us-east-1
```

Ausgabe:

```
{  
  "SourceRegions": [  
    {  
      "RegionName": "af-south-1",  
      "Endpoint": "https://rds.af-south-1.amazonaws.com",  
      "Status": "available",  
      "SupportsDBInstanceAutomatedBackupsReplication": false  
    },  
    {  
      "RegionName": "ap-east-1",  
      "Endpoint": "https://rds.ap-east-1.amazonaws.com",  
      "Status": "available",  
      "SupportsDBInstanceAutomatedBackupsReplication": false  
    },  
    {  
      "RegionName": "ap-northeast-1",  
      "Endpoint": "https://rds.ap-northeast-1.amazonaws.com",  
      "Status": "available",  
      "SupportsDBInstanceAutomatedBackupsReplication": true  
    },  
    {
```

```
    "RegionName": "ap-northeast-2",
    "Endpoint": "https://rds.ap-northeast-2.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "ap-northeast-3",
    "Endpoint": "https://rds.ap-northeast-3.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "ap-south-1",
    "Endpoint": "https://rds.ap-south-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "ap-southeast-1",
    "Endpoint": "https://rds.ap-southeast-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "ap-southeast-2",
    "Endpoint": "https://rds.ap-southeast-2.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "ap-southeast-3",
    "Endpoint": "https://rds.ap-southeast-3.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "ca-central-1",
    "Endpoint": "https://rds.ca-central-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-north-1",
    "Endpoint": "https://rds.eu-north-1.amazonaws.com",
```



```
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-south-1",
    "Endpoint": "https://rds.eu-south-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "eu-west-1",
    "Endpoint": "https://rds.eu-west-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-west-2",
    "Endpoint": "https://rds.eu-west-2.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-west-3",
    "Endpoint": "https://rds.eu-west-3.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "me-central-1",
    "Endpoint": "https://rds.me-central-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "me-south-1",
    "Endpoint": "https://rds.me-south-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "sa-east-1",
    "Endpoint": "https://rds.sa-east-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  }
```

```
    },
    {
      "RegionName": "us-east-2",
      "Endpoint": "https://rds.us-east-2.amazonaws.com",
      "Status": "available",
      "SupportsDBInstanceAutomatedBackupsReplication": true
    },
    {
      "RegionName": "us-west-1",
      "Endpoint": "https://rds.us-west-1.amazonaws.com",
      "Status": "available",
      "SupportsDBInstanceAutomatedBackupsReplication": true
    },
    {
      "RegionName": "us-west-2",
      "Endpoint": "https://rds.us-west-2.amazonaws.com",
      "Status": "available",
      "SupportsDBInstanceAutomatedBackupsReplication": true
    }
  ]
}
```

Weitere Informationen [finden Sie unter Informationen zu replizierten Backups](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeSourceRegions AWS CLI Befehlsreferenz](#).

describe-valid-db-instance-modifications

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-valid-db-instance-modifications`.

AWS CLI

Um gültige Änderungen für eine DB-Instance zu beschreiben

Im folgenden `describe-valid-db-instance-modifications` Beispiel werden Details zu den gültigen Änderungen für die angegebene DB-Instance abgerufen.

```
aws rds describe-valid-db-instance-modifications \
  --db-instance-identifier test-instance
```

Ausgabe:

```
{
  "ValidDBInstanceModificationsMessage": {
    "ValidProcessorFeatures": [],
    "Storage": [
      {
        "StorageSize": [
          {
            "Step": 1,
            "To": 20,
            "From": 20
          },
          {
            "Step": 1,
            "To": 6144,
            "From": 22
          }
        ],
        "ProvisionedIops": [
          {
            "Step": 1,
            "To": 0,
            "From": 0
          }
        ],
        "IopsToStorageRatio": [
          {
            "To": 0.0,
            "From": 0.0
          }
        ],
        "StorageType": "gp2"
      },
      {
        "StorageSize": [
          {
            "Step": 1,
            "To": 6144,
            "From": 100
          }
        ],
        "ProvisionedIops": [
          {
            "Step": 1,
```

```
        "To": 40000,
        "From": 1000
      }
    ],
    "IopsToStorageRatio": [
      {
        "To": 50.0,
        "From": 1.0
      }
    ],
    "StorageType": "io1"
  },
  {
    "StorageSize": [
      {
        "Step": 1,
        "To": 20,
        "From": 20
      },
      {
        "Step": 1,
        "To": 3072,
        "From": 22
      }
    ],
    "ProvisionedIops": [
      {
        "Step": 1,
        "To": 0,
        "From": 0
      }
    ],
    "IopsToStorageRatio": [
      {
        "To": 0.0,
        "From": 0.0
      }
    ],
    "StorageType": "magnetic"
  }
]
}
```

- API-Einheiten finden Sie unter [DescribeValidDbInstanceModifications AWS CLI](#) Befehlsreferenz.

download-db-log-file-portion

Das folgende Codebeispiel zeigt, wie man es benutzt `download-db-log-file-portion`.

AWS CLI

Um eine DB-Protokolldatei herunterzuladen

Im folgenden `download-db-log-file-portion` Beispiel wird nur der neueste Teil Ihrer Protokolldatei heruntergeladen und in einer lokalen Datei mit dem Namen `gespeicherttail.txt`.

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier test-instance \  
  --log-file-name log.txt \  
  --output text > tail.txt
```

Um die gesamte Datei herunterzuladen, müssen Sie den `--starting-token 0` Parameter angeben. Im folgenden Beispiel wird die Ausgabe in einer lokalen Datei mit dem Namen `gespeichertfull.txt`.

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier test-instance \  
  --log-file-name log.txt \  
  --starting-token 0 \  
  --output text > full.txt
```

Die gespeicherte Datei kann Leerzeilen enthalten. Sie werden beim Herunterladen am Ende jedes Teils der Protokolldatei angezeigt. Dies verursacht im Allgemeinen keine Probleme bei der Analyse Ihrer Protokolldateien.

- API-Einheiten finden Sie [DownloadDbLogFilePortion](#) in der AWS CLI Befehlsreferenz.

generate-auth-token

Das folgende Codebeispiel zeigt, wie man es benutzt `generate-auth-token`.

AWS CLI

Um ein Authentifizierungstoken zu generieren

Im folgenden `generate-db-auth-token` Beispiel wird ein Authentifizierungstoken zur Verwendung mit der IAM Datenbankauthentifizierung generiert.

```
aws rds generate-db-auth-token \  
  --hostname aurmysql-test.cdgmuiadpid.us-west-2.rds.amazonaws.com \  
  --port 3306 \  
  --region us-east-1 \  
  --username jane_doe
```

Ausgabe:

```
aurmysql-test.cdgmuiadpid.us-west-2.rds.amazonaws.com:3306/?  
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-  
Credential=AKIAIESZCNJ30EXAMPLE%2F20180731%2Fus-east-1%2Frds-db%2Faws4_request&X-  
Amz-Date=20180731T235209Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-  
Signature=5a8753ebEXAMPLEa2c724e5667797EXAMPLE9d6ec6e3f427191fa41aeEXAMPLE
```

- API-Einheiten finden Sie [GenerateAuthToken](#) unter AWS CLI Befehlsreferenz.

`generate-db-auth-token`

Das folgende Codebeispiel zeigt, wie man es benutzt `generate-db-auth-token`.

AWS CLI

Um ein IAM Authentifizierungstoken zu generieren

Im folgenden `generate-db-auth-token` Beispiel wird ein IAM Authentifizierungstoken generiert, um eine Verbindung zu einer Datenbank herzustellen.

```
aws rds generate-db-auth-token \  
  --hostname mydb.123456789012.us-east-1.rds.amazonaws.com \  
  --port 3306 \  
  --region us-east-1 \  
  --username db_user
```

Ausgabe:

```
mydb.123456789012.us-east-1.rds.amazonaws.com:3306/?
Action=connect&DBUser=db_user&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIEXAMPLE%2Fus-east-1%2Frds-db%2Faws4_request&X-Amz-
Date=20210123T011543Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=88987EXAMPLE1EXAMPLE2EXAMPLE3EXAMPLE4EXAMPLE5EXAMPLE6
```

Weitere Informationen finden Sie unter [Herstellen einer Verbindung zu Ihrer DB-Instance mithilfe von IAM Authentifizierung](#) im RDSAmazon-Benutzerhandbuch und [Herstellen einer Verbindung zu Ihrem DB-Cluster mithilfe von IAM Authentifizierung](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [GenerateDbAuthToken](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

So listen Sie Tags auf einer RDS Amazon-Ressource auf

Das folgende `list-tags-for-resource` Beispiel listet alle Tags auf einer DB-Instance auf.

```
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

Ausgabe:

```
{
  "TagList": [
    {
      "Key": "Environment",
      "Value": "test"
    },
    {
      "Key": "Name",
      "Value": "MyDatabase"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging Amazon RDS Resources](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

modify-certificates

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-certificates`.

AWS CLI

Um das standardmäßige TLS Systemzertifikat `SSL/für neue DB-Instances` vorübergehend zu überschreiben

Das folgende `modify-certificates` Beispiel überschreibt vorübergehend das TLS Systemstandardzertifikat `SSL/für neue DB-Instances`.

```
aws rds modify-certificates \  
  --certificate-identifier rds-ca-2019
```

Ausgabe:

```
{  
  "Certificate": {  
    "CertificateIdentifier": "rds-ca-2019",  
    "CertificateType": "CA",  
    "Thumbprint": "EXAMPLE123456789012",  
    "ValidFrom": "2019-09-19T18:16:53Z",  
    "ValidTill": "2024-08-22T17:08:50Z",  
    "CertificateArn": "arn:aws:rds:us-east-1::cert:rds-ca-2019",  
    "CustomerOverride": true,  
    "CustomerOverrideValidTill": "2024-08-22T17:08:50Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Rotation Ihres SSL TLS /-Zertifikats](#) im RDS Amazon-Benutzerhandbuch und [Rotation Ihres SSL TLS /-Zertifikats](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyCertificates](#) in der AWS CLI Befehlsreferenz.

modify-current-db-cluster-capacity

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-current-db-cluster-capacity`.

AWS CLI

So skalieren Sie die Kapazität eines Aurora Serverless DB-Clusters

Im folgenden `modify-current-db-cluster-capacity` Beispiel wird die Kapazität eines Aurora Serverless DB-Clusters auf 8 skaliert.

```
aws rds modify-current-db-cluster-capacity \  
  --db-cluster-identifier mydbcluster \  
  --capacity 8
```

Ausgabe:

```
{  
  "DBClusterIdentifier": "mydbcluster",  
  "PendingCapacity": 8,  
  "CurrentCapacity": 1,  
  "SecondsBeforeTimeout": 300,  
  "TimeoutAction": "ForceApplyCapacityChange"  
}
```

Weitere Informationen finden Sie unter [Manuelles Skalieren der Kapazität des Aurora Serverless v1-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ModifyCurrentDbClusterCapacity AWS CLI Befehlsreferenz](#).

`modify-db-cluster-endpoint`

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-cluster-endpoint`.

AWS CLI

Um einen benutzerdefinierten DB-Cluster-Endpunkt zu ändern

Im folgenden `modify-db-cluster-endpoint` Beispiel wird der angegebene benutzerdefinierte DB-Cluster-Endpunkt geändert.

```
aws rds modify-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --static-members dbinstance1 dbinstance2 dbinstance3
```

Ausgabe:

```
{
  "DBClusterEndpointIdentifier": "mycustomendpoint",
  "DBClusterIdentifier": "mydbcluster",
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
  "Endpoint": "mycustomendpoint.cluster-custom-cnpxample.us-east-1.rds.amazonaws.com",
  "Status": "modifying",
  "EndpointType": "CUSTOM",
  "CustomEndpointType": "READER",
  "StaticMembers": [
    "dbinstance1",
    "dbinstance2",
    "dbinstance3"
  ],
  "ExcludedMembers": [],
  "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:mycustomendpoint"
}
```

Weitere Informationen finden Sie unter [Amazon Aurora Connection Management](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbClusterEndpoint](#) unter AWS CLI Befehlsreferenz.

modify-db-cluster-parameter-group

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-cluster-parameter-group`.

AWS CLI

Um Parameter in einer DB-Cluster-Parametergruppe zu ändern

Im folgenden `modify-db-cluster-parameter-group` Beispiel werden die Werte von Parametern in einer DB-Cluster-Parametergruppe geändert.

```
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name mydbclusterpg \
  --
parameters "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate"
\
"ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Ausgabe:

```
{
  "DBClusterParameterGroupName": "mydbclusterpg"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbClusterParameterGroup](#) unter AWS CLI Befehlsreferenz.

modify-db-cluster-snapshot-attribute

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-cluster-snapshot-attribute`.

AWS CLI

Um ein DB-Cluster-Snapshot-Attribut zu ändern

Im folgenden `modify-db-cluster-snapshot-attribute` Beispiel werden Änderungen am angegebenen DB-Cluster-Snapshot-Attribut vorgenommen.

```
aws rds modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier myclustersnapshot \
  --attribute-name restore \
  --values-to-add 123456789012
```

Ausgabe:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "myclustersnapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012"
        ]
      }
    ]
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Wiederherstellen aus einem DB-Cluster-Snapshot](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbClusterSnapshotAttribute](#) unter AWS CLI Befehlsreferenz.

modify-db-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-cluster`.

AWS CLI

Beispiel 1: Um einen DB-Cluster zu ändern

Im folgenden `modify-db-cluster` Beispiel wird das Masterbenutzerkennwort für den genannten DB-Cluster geändert `cluster-2` und die Aufbewahrungsfrist für Backups auf 14 Tage festgelegt. Der `--apply-immediately` Parameter bewirkt, dass die Änderungen sofort vorgenommen werden, anstatt bis zum nächsten Wartungsfenster zu warten.

```
aws rds modify-db-cluster \  
  --db-cluster-identifier cluster-2 \  
  --backup-retention-period 14 \  
  --master-user-password newpassword99 \  
  --apply-immediately
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "eu-central-1b",  
      "eu-central-1c",  
      "eu-central-1a"  
    ],  
    "BackupRetentionPeriod": 14,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "cluster-2",  
    "DBClusterParameterGroup": "default.aurora5.6",  
    "DBSubnetGroup": "default-vpc-2305ca49",  
    "Status": "available",
```

```
    "EarliestRestorableTime": "2020-06-03T02:07:29.637Z",
    "Endpoint": "cluster-2.cluster-#####.eu-central-1.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-2.cluster-ro-#####.eu-
central-1.rds.amazonaws.com",
    "MultiAZ": false,
    "Engine": "aurora",
    "EngineVersion": "5.6.10a",
    "LatestRestorableTime": "2020-06-04T15:11:25.748Z",
    "Port": 3306,
    "MasterUsername": "admin",
    "PreferredBackupWindow": "01:55-02:25",
    "PreferredMaintenanceWindow": "thu:21:14-thu:21:44",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [
      {
        "DBInstanceIdentifier": "cluster-2-instance-1",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      }
    ],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-20a5c047",
        "Status": "active"
      }
    ],
    "HostedZoneId": "Z1RLNU0EXAMPLE",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:eu-central-1:123456789012:key/
d1bd7c8f-5cdb-49ca-8a62-a1b2c3d4e5f6",
    "DbClusterResourceId": "cluster-AGJ7XI77XVIS6FUXHU1EXAMPLE",
    "DBClusterArn": "arn:aws:rds:eu-central-1:123456789012:cluster:cluster-2",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2020-04-03T14:44:02.764Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": true,
    "CrossAccountClone": false,
    "DomainMemberships": []
  }
```

```
}
```

Weitere Informationen finden Sie unter [Ändern eines Amazon Aurora Aurora-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Beispiel 2: Um eine VPC Sicherheitsgruppe einem DB-Cluster zuzuordnen

Das folgende `modify-db-instance` Beispiel ordnet eine bestimmte VPC Sicherheitsgruppe zu und entfernt DB-Sicherheitsgruppen aus einem DB-Cluster.

```
aws rds modify-db-cluster \  
  --db-cluster-identifier dbName \  
  --vpc-security-group-ids sg-ID
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2b",  
      "us-west-2a"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "dbName",  
    "DBClusterParameterGroup": "default.aurora-mysql8.0",  
    "DBSubnetGroup": "default",  
    "Status": "available",  
    "EarliestRestorableTime": "2024-02-15T01:12:13.966000+00:00",  
    "Endpoint": "dbName.cluster-abcdefghji.us-west-2.rds.amazonaws.com",  
    "ReaderEndpoint": "dbName.cluster-ro-abcdefghji.us-  
west-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-mysql",  
    "EngineVersion": "8.0.mysql_aurora.3.04.1",  
    "LatestRestorableTime": "2024-02-15T02:25:33.696000+00:00",  
    "Port": 3306,  
    "MasterUsername": "admin",  
    "PreferredBackupWindow": "10:59-11:29",  
    "PreferredMaintenanceWindow": "thu:08:54-thu:09:24",  
    "ReadReplicaIdentifiers": [],  
    "DBClusterMembers": [  

```

```
    {
      "DBInstanceIdentifier": "dbName-instance-1",
      "IsClusterWriter": true,
      "DBClusterParameterGroupStatus": "in-sync",
      "PromotionTier": 1
    }
  ],
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sg-ID",
      "Status": "active"
    }
  ],
  ...output omitted...
}
```

Weitere Informationen finden Sie unter [Steuern des Zugriffs mit Sicherheitsgruppen](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbCluster](#) unter AWS CLI Befehlsreferenz.

modify-db-instance

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-instance`.

AWS CLI

Beispiel 1: Um eine DB-Instance zu ändern

Das folgende `modify-db-instance` Beispiel ordnet einer kompatiblen Microsoft SQL Server-DB-Instance eine Optionsgruppe und eine Parametergruppe zu. Der `--apply-immediately` Parameter bewirkt, dass die Options- und Parametergruppen sofort verknüpft werden, anstatt bis zum nächsten Wartungsfenster zu warten.

```
aws rds modify-db-instance \  
  --db-instance-identifier database-2 \  
  --option-group-name test-se-2017 \  
  --db-parameter-group-name test-sqlserver-se-2017 \  
  --apply-immediately
```

Ausgabe:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "database-2",
    "DBInstanceClass": "db.r4.large",
    "Engine": "sqlserver-se",
    "DBInstanceStatus": "available",

    ...output omitted...

    "DBParameterGroups": [
      {
        "DBParameterGroupName": "test-sqlserver-se-2017",
        "ParameterApplyStatus": "applying"
      }
    ],
    "AvailabilityZone": "us-west-2d",

    ...output omitted...

    "MultiAZ": true,
    "EngineVersion": "14.00.3281.6.v1",
    "AutoMinorVersionUpgrade": false,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "license-included",
    "OptionGroupMemberships": [
      {
        "OptionGroupName": "test-se-2017",
        "Status": "pending-apply"
      }
    ],
    "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
    "SecondaryAvailabilityZone": "us-west-2c",
    "PubliclyAccessible": true,
    "StorageType": "gp2",

    ...output omitted...

    "DeletionProtection": false,
    "AssociatedRoles": [],
    "MaxAllocatedStorage": 1000
  }
}
```


Weitere Informationen finden Sie unter [Ändern einer RDS Amazon-DB-Instance](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: So ordnen Sie eine VPC Sicherheitsgruppe einer DB-Instance zu

Im folgenden `modify-db-instance` Beispiel wird eine bestimmte VPC Sicherheitsgruppe zugeordnet und DB-Sicherheitsgruppen aus einer DB-Instance entfernt:

```
aws rds modify-db-instance \  
  --db-instance-identifizier dbName \  
  --vpc-security-group-ids sg-ID
```

Ausgabe:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifizier": "dbName",  
    "DBInstanceClass": "db.t3.micro",  
    "Engine": "mysql",  
    "DBInstanceStatus": "available",  
    "MasterUsername": "admin",  
    "Endpoint": {  
      "Address": "dbName.abcdefghijkl.us-west-2.rds.amazonaws.com",  
      "Port": 3306,  
      "HostedZoneId": "ABCDEFGHIJK1234"  
    },  
    "AllocatedStorage": 20,  
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",  
    "PreferredBackupWindow": "11:57-12:27",  
    "BackupRetentionPeriod": 7,  
    "DBSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-ID",  
        "Status": "active"  
      }  
    ],  
    "... output omitted ..."  
    "MultiAZ": false,  
    "EngineVersion": "8.0.35",  
    "AutoMinorVersionUpgrade": true,  
    "ReadReplicaDBInstanceIdentifiziers": [],  
    "LicenseModel": "general-public-license",
```

```
... output ommited ...  
}  
}
```

Weitere Informationen finden Sie unter [Zugriffskontrolle mit Sicherheitsgruppen](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [M odifyDBInstance](#) in der AWS CLI Befehlsreferenz.

modify-db-parameter-group

Das folgende Codebeispiel zeigt, wie man `modify-db-parameter-group`.

AWS CLI

Um eine DB-Parametergruppe zu ändern

Im folgenden `modify-db-parameter-group` Beispiel wird der Wert des `clr enabled` Parameters in einer DB-Parametergruppe geändert. Der `--apply-immediately` Parameter bewirkt, dass die DB-Parametergruppe sofort geändert wird, anstatt bis zum nächsten Wartungsfenster zu warten.

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name test-sqlserver-se-2017 \  
  --parameters "ParameterName='clr  
enabled',ParameterValue=1,ApplyMethod=immediate"
```

Ausgabe:

```
{  
  "DBParameterGroupName": "test-sqlserver-se-2017"  
}
```

Weitere Informationen finden Sie unter [Ändern von Parametern in einer DB-Parametergruppe](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [M odifyDBParameter Group](#) in der AWS CLI Befehlsreferenz.

modify-db-proxy-endpoint

Das folgende Codebeispiel zeigt die Verwendung `modify-db-proxy-endpoint`.

AWS CLI

Um einen DB-Proxyendpunkt für eine RDS Datenbank zu ändern

Im folgenden `modify-db-proxy-endpoint` Beispiel wird ein DB-Proxyendpunkt so geändert, dass das Lese-Timeout auf 65 Sekunden festgelegt wird.

```
aws rds modify-db-proxy-endpoint \  
  --db-proxy-endpoint-name proxyEndpoint \  
  --cli-read-timeout 65
```

Ausgabe:

```
{  
  "DBProxyEndpoint":  
    {  
      "DBProxyEndpointName": "proxyEndpoint",  
      "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-  
endpoint:prx-endpoint-0123a01b12345c0ab",  
      "DBProxyName": "proxyExample",  
      "Status": "available",  
      "VpcId": "vpc-1234567",  
      "VpcSecurityGroupIds": [  
        "sg-1234"  
      ],  
      "VpcSubnetIds": [  
        "subnetgroup1",  
        "subnetgroup2"  
      ],  
      "Endpoint": "proxyEndpoint.endpoint.proxyExample-ab0cd1efghij.us-  
east-1.rds.amazonaws.com",  
      "CreateDate": "2023-04-05T16:09:33.452000+00:00",  
      "TargetRole": "READ_WRITE",  
      "IsDefault": "false"  
    }  
}
```

Weitere Informationen finden Sie unter [Ändern eines Proxy-Endpunkts](#) im RDS Amazon-Benutzerhandbuch und [Ändern eines Proxy-Endpunkts](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbProxyEndpoint](#) in der AWS CLI Befehlsreferenz.

modify-db-proxy-target-group

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-proxy-target-group`.

AWS CLI

Um einen DB-Proxyendpunkt zu ändern

Im folgenden `modify-db-proxy-target-group` Beispiel wird eine DB-Proxy-Zielgruppe dahingehend geändert, dass die maximale Anzahl an Verbindungen auf 80 Prozent und die maximale Anzahl inaktiver Verbindungen auf 10 Prozent festgelegt wird.

```
aws rds modify-db-proxy-target-group \
  --target-group-name default \
  --db-proxy-name proxyExample \
  --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10
```

Ausgabe:

```
{
  "DBProxyTargetGroup":
    {
      "DBProxyName": "proxyExample",
      "TargetGroupName": "default",
      "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
      "IsDefault": true,
      "Status": "available",
      "ConnectionPoolConfig": {
        "MaxConnectionsPercent": 80,
        "MaxIdleConnectionsPercent": 10,
        "ConnectionBorrowTimeout": 120,
        "SessionPinningFilters": []
      },
      "CreateDate": "2023-05-02T18:41:19.495000+00:00",
      "UpdateDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

Weitere Informationen finden Sie unter [Ändern eines RDS Proxys](#) im RDS Amazon-Benutzerhandbuch und [Ändern eines RDS Proxys](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbProxyTargetGroup](#) in der AWS CLI Befehlsreferenz.

modify-db-proxy

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-proxy`.

AWS CLI

Um einen DB-Proxy für eine RDS Datenbank zu ändern

Im folgenden `modify-db-proxy` Beispiel wird ein DB-Proxy mit dem Namen `proxyExample` Require SSL für seine Verbindungen geändert.

```
aws rds modify-db-proxy \  
  --db-proxy-name proxyExample \  
  --require-tls
```

Ausgabe:

```
{  
  "DBProxy":  
    {  
      "DBProxyName": "proxyExample",  
      "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-  
proxy:prx-0123a01b12345c0ab",  
      "Status": "modifying"  
      "EngineFamily": "PostgreSQL",  
      "VpcId": "sg-1234567",  
      "VpcSecurityGroupIds": [  
        "sg-1234"  
      ],  
      "VpcSubnetIds": [  
        "subnetgroup1",  
        "subnetgroup2"  
      ],  
      "Auth": "[  
        {  
          "Description": "proxydescription1",  
          "AuthScheme": "SECRETS",  
          "SecretArn": "arn:aws:secretsmanager:us-  
west-2:123456789123:secret:proxysecret1-Abcd1e",  
          "IAMAuth": "DISABLED"  
        }  
      ]",  
      "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
```

```

    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "RequireTLS": true,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdateDate": "2023-04-13T01:49:38.568000+00:00"
  }
}

```

Weitere Informationen finden Sie unter [Ändern eines RDS Proxys](#) im RDSAmazon-Benutzerhandbuch und [Erstellen eines RDS Proxys](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbProxy](#) in der AWS CLI Befehlsreferenz.

modify-db-shard-group

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-shard-group`.

AWS CLI

Beispiel 1: Um eine DB-Shard-Gruppe zu ändern

Im folgenden `modify-db-shard-group` Beispiel wird die maximale Kapazität einer DB-Shard-Gruppe geändert.

```

aws rds modify-db-shard-group \
  --db-shard-group-identifier my-db-shard-group \
  --max-acu 1000

```

Ausgabe:

```

{
  "DBShardGroups": [
    {
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
      "DBShardGroupIdentifier": "my-db-shard-group",
      "DBClusterIdentifier": "my-sv2-cluster",
      "MaxACU": 768.0,
      "ComputeRedundancy": 0,
      "Status": "available",
      "PubliclyAccessible": false,
      "Endpoint": "my-sv2-cluster.limitless-cekyceexample.us-east-2.rds.amazonaws.com"
    }
  ]
}

```

```
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Amazon Aurora DB Clusters](#) im Amazon Aurora Benutzerhandbuch.

Beispiel 2: Um Ihre DB-Shard-Gruppen zu beschreiben

Im folgenden `describe-db-shard-groups` Beispiel werden die Details Ihrer DB-Shard-Gruppen abgerufen, nachdem Sie den Befehl ausgeführt haben. `modify-db-shard-group` Die maximale Kapazität der DB-Shard-Gruppe `my-db-shard-group` beträgt jetzt 1000 Aurora-Kapazitätseinheiten (ACUs).

```
aws rds describe-db-shard-groups
```

Ausgabe:

```
{  
  "DBShardGroups": [  
    {  
      "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",  
      "DBShardGroupIdentifier": "limitless-test-shard-grp",  
      "DBClusterIdentifier": "limitless-test-cluster",  
      "MaxACU": 768.0,  
      "ComputeRedundancy": 0,  
      "Status": "available",  
      "PubliclyAccessible": true,  
      "Endpoint": "limitless-test-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"  
    },  
    {  
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",  
      "DBShardGroupIdentifier": "my-db-shard-group",  
      "DBClusterIdentifier": "my-sv2-cluster",  
      "MaxACU": 1000.0,  
      "ComputeRedundancy": 0,  
      "Status": "available",  
      "PubliclyAccessible": false,  
      "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"  
    }  
  ]  
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Amazon Aurora DB Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbShardGroup](#) unter AWS CLI Befehlsreferenz.

modify-db-snapshot-attribute

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-snapshot-attribute`.

AWS CLI

Beispiel 1: Um zwei AWS Konten die Wiederherstellung eines DB-Snapshots zu ermöglichen

Im folgenden `modify-db-snapshot-attribute` Beispiel wird zwei AWS Konten mit den Kennungen 111122223333 und die Erlaubnis erteilt 444455556666, den DB-Snapshot mit dem Namen `mydbsnapshot` wiederherzustellen.

```
aws rds modify-db-snapshot-attribute \
  --db-snapshot-identifier mydbsnapshot \
  --attribute-name restore \
  --values-to-add {"111122223333","444455556666"}
```

Ausgabe:

```
{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "111122223333",
          "444455556666"
        ]
      }
    ]
  }
}
```


Weitere Informationen finden Sie unter [Einen Snapshot teilen](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: Um zu verhindern, dass ein AWS Konto einen DB-Snapshot wiederherstellt

Im folgenden `modify-db-snapshot-attribute` Beispiel wird einem bestimmten AWS Konto die Erlaubnis entzogen, den genannten DB-Snapshot wiederherzustellen `mydbsnapshot`. Bei der Angabe eines einzelnen Kontos darf die Konto-ID nicht von Anführungszeichen oder geschweiften Klammern umgeben werden.

```
aws rds modify-db-snapshot-attribute \  
  --db-snapshot-identifizier mydbsnapshot \  
  --attribute-name restore \  
  --values-to-remove 444455556666
```

Ausgabe:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifizier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "111122223333"  
        ]  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Einen Snapshot teilen](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [ModifyDbSnapshotAttribute](#) in der AWS CLI Befehlsreferenz.

modify-db-snapshot-attributes

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-snapshot-attributes`.

AWS CLI

Um ein DB-Snapshot-Attribut zu ändern

Das folgende `modify-db-snapshot-attribute` Beispiel erlaubt zwei AWS Konto-IDs, 111122223333 und 444455556666, den genannten `mydbsnapshot` DB-Snapshot wiederherzustellen.

```
aws rds modify-db-snapshot-attribute \  
  --db-snapshot-identifizier mydbsnapshot \  
  --attribute-name restore \  
  --values-to-add '["111122223333", "444455556666"]'
```

Ausgabe:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifizier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "111122223333",  
          "444455556666"  
        ]  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Einen Snapshot teilen](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDbSnapshotAttributes](#) in der AWS CLI Befehlsreferenz.

modify-db-snapshot

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-snapshot`.

AWS CLI

Um einen DB-Snapshot zu ändern

Im folgenden `modify-db-snapshot` Beispiel wird ein PostgreSQL 10.6-Snapshot mit dem Namen `db5-snapshot-upg-test` auf PostgreSQL 11.7 aktualisiert. Die neue DB-Engine-Version wird angezeigt, nachdem das Upgrade des Snapshots abgeschlossen wurde und sein Status verfügbar ist.

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifizier db5-snapshot-upg-test \  
  --engine-version 11.7
```

Ausgabe:

```
{  
  "DBSnapshot": {  
    "DBSnapshotIdentifizier": "db5-snapshot-upg-test",  
    "DBInstanceIdentifizier": "database-5",  
    "SnapshotCreateTime": "2020-03-27T20:49:17.092Z",  
    "Engine": "postgres",  
    "AllocatedStorage": 20,  
    "Status": "upgrading",  
    "Port": 5432,  
    "AvailabilityZone": "us-west-2a",  
    "VpcId": "vpc-2ff27557",  
    "InstanceCreateTime": "2020-03-27T19:59:04.735Z",  
    "MasterUsername": "postgres",  
    "EngineVersion": "10.6",  
    "LicenseModel": "postgresql-license",  
    "SnapshotType": "manual",  
    "OptionGroupName": "default:postgres-11",  
    "PercentProgress": 100,  
    "StorageType": "gp2",  
    "Encrypted": false,  
    "DBSnapshotArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-upg-test",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "ProcessorFeatures": [],  
    "DbiResourceId": "db-GJMF75LM42IL6BTFRE4UZJ5YM4"  
  }  
}
```

Weitere Informationen finden Sie unter [Upgraden eines SQL Postgre-DB-Snapshots](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [ModifyDbSnapshot](#) in der AWS CLI Befehlsreferenz.

modify-db-subnet-group

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-db-subnet-group`.

AWS CLI

Um eine DB-Subnetzgruppe zu ändern

Im folgenden `modify-db-subnet-group` Beispiel wird der DB-Subnetzgruppe namens ein Subnetz mit der ID `subnet-08e41f9e230222222` hinzugefügt. `mysubnetgroup` Um die vorhandenen Subnetze in der Subnetzgruppe beizubehalten, fügen Sie deren AS-Werte in die `IDs` Option ein. `--subnet-ids` Stellen Sie sicher, dass Subnetze mit mindestens zwei verschiedenen Availability Zones in der DB-Subnetzgruppe vorhanden sind.

```
aws rds modify-db-subnet-group \  
  --db-subnet-group-name mysubnetgroup \  
  --subnet-ids  
  '["subnet-0a1dc4e1a6f123456", "subnet-070dd7ecb3aaaaaaa", "subnet-00f5b198bc0abcdef", "subnet-
```

Ausgabe:

```
{  
  "DBSubnetGroup": {  
    "DBSubnetGroupName": "mysubnetgroup",  
    "DBSubnetGroupDescription": "test DB subnet group",  
    "VpcId": "vpc-0f08e7610a1b2c3d4",  
    "SubnetGroupStatus": "Complete",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-08e41f9e230222222",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        },  
        "SubnetStatus": "Active"  
      },  
      {  
        "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2b"  
        },  
        "SubnetStatus": "Active"  
      },  
      {  
        "SubnetIdentifier": "subnet-00f5b198bc0abcdef",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2d"  
        }  
      }  
    ]  
  }  
}
```

```

    },
    "SubnetStatus": "Active"
  },
  {
    "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
    "SubnetAvailabilityZone": {
      "Name": "us-west-2b"
    },
    "SubnetStatus": "Active"
  }
],
"DBSubnetGroupArn": "arn:aws:rds:us-
west-2:534026745191:subgrp:mysubnetgroup"
}
}

```

Weitere Informationen finden Sie unter [Schritt 3: Erstellen einer DB-Subnetzgruppe](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [ModifyDbSubnetGroup AWS CLI](#) Befehlsreferenz.

modify-event-subscription

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-event-subscription`.

AWS CLI

Um ein Event-Abonnement zu ändern

Im folgenden `modify-event-subscription` Beispiel wird das angegebene Event-Abonnement deaktiviert, sodass es keine Benachrichtigungen mehr zum angegebenen Amazon Simple Notification Service-Thema veröffentlicht.

```

aws rds modify-event-subscription \
  --subscription-name my-instance-events \
  --no-enabled

```

Ausgabe:

```

{
  "EventSubscription": {
    "EventCategoriesList": [

```

```

        "backup",
        "recovery"
    ],
    "CustomerAwsId": "123456789012",
    "SourceType": "db-instance",
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-
events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
    "CustSubscriptionId": "my-instance-events",
    "Status": "modifying",
    "Enabled": false
    }
}

```

- API-Einheiten finden Sie [ModifyEventSubscription](#) in der AWS CLI Befehlsreferenz.

modify-global-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `modify-global-cluster`.

AWS CLI

Um einen globalen DB-Cluster zu ändern

Das folgende `modify-global-cluster` Beispiel aktiviert den Löschschutz für einen Aurora My SQL -kompatiblen globalen DB-Cluster.

```

aws rds modify-global-cluster \
  --global-cluster-identifier myglobalcluster \
  --deletion-protection

```

Ausgabe:

```

{
  "GlobalCluster": {
    "GlobalClusterIdentifier": "myglobalcluster",
    "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
    "Status": "available",
    "Engine": "aurora-mysql",
  }
}

```

```
    "EngineVersion": "5.7.mysql_aurora.2.07.2",
    "StorageEncrypted": false,
    "DeletionProtection": true,
    "GlobalClusterMembers": []
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung einer globalen Aurora-Datenbank](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyGlobalCluster](#) unter AWS CLI Befehlsreferenz.

promote-read-replica-db-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `promote-read-replica-db-cluster`.

AWS CLI

Um einen DB-Cluster hochzustufen, lesen Sie [Replica](#)

Im folgenden `promote-read-replica-db-cluster` Beispiel wird die angegebene Read Replica zu einem eigenständigen DB-Cluster heraufgestuft.

```
aws rds promote-read-replica-db-cluster \
  --db-cluster-identifier mydbcluster-1
```

Ausgabe:

```
{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1c"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "",
    "DBClusterIdentifier": "mydbcluster-1",
    ...some output truncated...
  }
}
```

Weitere Informationen finden Sie unter [Heraufstufen einer Read Replica zu einem DB-Cluster](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie unter [PromoteReadReplicaDbCluster AWS CLI](#) Befehlsreferenz.

promote-read-replica

Das folgende Codebeispiel zeigt, wie man es benutzt `promote-read-replica`.

AWS CLI

Um ein Read Replica zu bewerben

Im folgenden `promote-read-replica` Beispiel wird die angegebene Read Replica zu einer eigenständigen DB-Instance heraufgestuft.

```
aws rds promote-read-replica \  
  --db-instance-identifier test-instance-repl
```

Ausgabe:

```
{  
  "DBInstance": {  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",  
    "StorageType": "standard",  
    "ReadReplicaSourceDBInstanceIdentifier": "test-instance",  
    "DBInstanceStatus": "modifying",  
    ...some output truncated...  
  }  
}
```

- API-Einheiten finden Sie unter [PromoteReadReplica AWS CLI](#) Befehlsreferenz.

purchase-reserved-db-instance

Das folgende Codebeispiel zeigt, wie man es benutzt `purchase-reserved-db-instance`.

AWS CLI

Um ein reserviertes DB-Instance-Angebot zu erwerben

Im folgenden `purchase-reserved-db-instances-offering` Beispiel wird ein reserviertes DB-Instance-Angebot erworben. Die `reserved-db-instances-offering-id` muss eine gültige Angebots-ID sein, wie sie vom `describe-reserved-db-instances-offering` Befehl zurückgegeben wird.

```
aws rds purchase-reserved-db-instances -offering -- ID reserved-db-instances-offering
438012d3-4a52-4cc7-b2e3-8dff72e0e706
```

- API-Einheiten finden [PurchaseReservedDbInstance](#) Sie AWS CLI in der Befehlsreferenz.

`purchase-reserved-db-instances-offerings`

Das folgende Codebeispiel zeigt, wie man es benutzt `purchase-reserved-db-instances-offerings`.

AWS CLI

Beispiel 1: Um eine reservierte DB-Instance zum Kauf zu finden

Das folgende `describe-reserved-db-instances-offerings` Beispiel listet die verfügbaren reservierten My SQL DB-Instances mit der Instance-Klasse `db.t2.micro` und einer Dauer von einem Jahr auf. Die Angebots-ID ist für den Kauf einer reservierten DB-Instance erforderlich.

```
aws rds describe-reserved-db-instances-offerings \
  --product-description mysql \
  --db-instance-class db.t2.micro \
  --duration 1
```

Ausgabe:

```
{
  "ReservedDBInstancesOfferings": [
    {
      "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
      "DBInstanceClass": "db.t2.micro",
      "Duration": 31536000,
      "FixedPrice": 51.0,
      "UsagePrice": 0.0,
      "CurrencyCode": "USD",
      "ProductDescription": "mysql",
      "OfferingType": "Partial Upfront",
```

```

        "MultiAZ": false,
        "RecurringCharges": [
            {
                "RecurringChargeAmount": 0.006,
                "RecurringChargeFrequency": "Hourly"
            }
        ]
    },
    ... some output truncated ...
]
}

```

Weitere Informationen finden Sie unter [Reserved DB Instances for Amazon RDS](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: Um eine reservierte DB-Instance zu erwerben

Das folgende `purchase-reserved-db-instances-offering` Beispiel zeigt, wie Sie das reservierte DB-Instance-Angebot aus dem vorherigen Beispiel kaufen können.

```
aws rds purchase-reserved-db-instances -offering --id reserved-db-instances-offering
8ba30be1-b9ec-447f-8f23-6114e3f4c7b4
```

Ausgabe:

```

{
  "ReservedDBInstance": {
    "ReservedDBInstanceId": "ri-2020-06-29-16-54-57-670",
    "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
    "DBInstanceClass": "db.t2.micro",
    "StartTime": "2020-06-29T16:54:57.670Z",
    "Duration": 31536000,
    "FixedPrice": 51.0,
    "UsagePrice": 0.0,
    "CurrencyCode": "USD",
    "DBInstanceCount": 1,
    "ProductDescription": "mysql",
    "OfferingType": "Partial Upfront",
    "MultiAZ": false,
    "State": "payment-pending",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": 0.006,

```

```

        "RecurringChargeFrequency": "Hourly"
      }
    ],
    "ReservedDBInstanceArn": "arn:aws:rds:us-
west-2:123456789012:ri:ri-2020-06-29-16-54-57-670"
  }
}

```

Weitere Informationen finden Sie unter [Reserved DB Instances for Amazon RDS](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [PurchaseReservedDbInstancesOfferings](#) unter AWS CLI Befehlsreferenz.

reboot-db-instance

Das folgende Codebeispiel zeigt, wie man es benutzt `reboot-db-instance`.

AWS CLI

Um eine DB-Instance neu zu starten

Das folgende `reboot-db-instance` Beispiel startet einen Neustart der angegebenen DB-Instance.

```

aws rds reboot-db-instance \
  --db-instance-identifier test-mysql-instance

```

Ausgabe:

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "test-mysql-instance",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "rebooting",
    "MasterUsername": "admin",
    "Endpoint": {
      "Address": "test-mysql-instance.#####.us-
west-2.rds.amazonaws.com",
      "Port": 3306,
      "HostedZoneId": "Z1PVIF0EXAMPLE"
    }
  }
}

```

```
    },  
    ... output omitted...  
  }  
}
```

Weitere Informationen finden Sie unter [Rebooting a DB Instance](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie unter [R ebootDBInstance](#) in der AWS CLI Befehlsreferenz.

reboot-db-shard-group

Das folgende Codebeispiel zeigt, wie man `reboot-db-shard-group`.

AWS CLI

Beispiel 1: Um eine DB-Shard-Gruppe neu zu starten

Im folgenden `reboot-db-shard-group` Beispiel wird eine DB-Shard-Gruppe neu gestartet.

```
aws rds reboot-db-shard-group \  
  --db-shard-group-identifier my-db-shard-group
```

Ausgabe:

```
{  
  "DBShardGroups": [  
    {  
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",  
      "DBShardGroupIdentifier": "my-db-shard-group",  
      "DBClusterIdentifier": "my-sv2-cluster",  
      "MaxACU": 1000.0,  
      "ComputeRedundancy": 0,  
      "Status": "available",  
      "PubliclyAccessible": false,  
      "Endpoint": "my-sv2-cluster.limitless-cekyceexample.us-  
east-2.rds.amazonaws.com"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Neustart eines Amazon Aurora Aurora-DB-Clusters oder einer Amazon Aurora Aurora-DB-Instance](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Beispiel 2: Um Ihre DB-Shard-Gruppen zu beschreiben

Im folgenden `describe-db-shard-groups` Beispiel werden die Details Ihrer DB-Shard-Gruppen abgerufen, nachdem Sie den Befehl ausgeführt haben. `reboot-db-shard-group` Die DB-Shard-Gruppe `my-db-shard-group` wird jetzt neu gestartet.

```
aws rds describe-db-shard-groups
```

Ausgabe:

```
{
  "DBShardGroups": [
    {
      "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
      "DBShardGroupIdentifier": "limitless-test-shard-grp",
      "DBClusterIdentifier": "limitless-test-cluster",
      "MaxACU": 768.0,
      "ComputeRedundancy": 0,
      "Status": "available",
      "PubliclyAccessible": true,
      "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-east-2.rds.amazonaws.com"
    },
    {
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
      "DBShardGroupIdentifier": "my-db-shard-group",
      "DBClusterIdentifier": "my-sv2-cluster",
      "MaxACU": 1000.0,
      "ComputeRedundancy": 0,
      "Status": "rebooting",
      "PubliclyAccessible": false,
      "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-east-2.rds.amazonaws.com"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Neustart eines Amazon Aurora Aurora-DB-Clusters oder einer Amazon Aurora Aurora-DB-Instance](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [RebootDbShardGroup](#) in der AWS CLI Befehlsreferenz.

register-db-proxy-targets

Das folgende Codebeispiel zeigt, wie man es benutzt `register-db-proxy-targets`.

AWS CLI

Um einen DB-Proxy bei einer Datenbank zu registrieren

Das folgende `register-db-proxy-targets` Beispiel erstellt die Zuordnung zwischen einer Datenbank und einem Proxy.

```
aws rds register-db-proxy-targets \  
  --db-proxy-name proxyExample \  
  --db-cluster-identifiers database-5
```

Ausgabe:

```
{  
  "DBProxyTargets": [  
    {  
      "RdsResourceId": "database-5",  
      "Port": 3306,  
      "Type": "TRACKED_CLUSTER",  
      "TargetHealth": {  
        "State": "REGISTERING"  
      }  
    },  
    {  
      "Endpoint": "database-5instance-1.ab0cd1efghij.us-east-1.rds.amazonaws.com",  
      "RdsResourceId": "database-5",  
      "Port": 3306,  
      "Type": "RDS_INSTANCE",  
      "TargetHealth": {  
        "State": "REGISTERING"  
      }  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Erstellen eines RDS Proxys](#) im RDSAmazon-Benutzerhandbuch und [Erstellen eines RDS Proxys](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterDbProxyTargets](#) in der AWS CLI Befehlsreferenz.

remove-from-global-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-from-global-cluster`.

AWS CLI

Um einen sekundären Aurora-Cluster von einem globalen Aurora-Datenbankcluster zu trennen

Im folgenden `remove-from-global-cluster` Beispiel wird ein sekundärer Aurora-Cluster von einem globalen Aurora-Datenbankcluster getrennt. Der Cluster wechselt von einem schreibgeschützten zu einem eigenständigen Cluster mit Lese- und Schreibfunktion.

```
aws rds remove-from-global-cluster \
  --region us-west-2 \
  --global-cluster-identifizier myglobalcluster \
  --db-cluster-identifizier arn:aws:rds:us-west-2:123456789012:cluster:DB-1
```

Ausgabe:

```
{
  "GlobalCluster": {
    "GlobalClusterIdentifizier": "myglobalcluster",
    "GlobalClusterResourceId": "cluster-abc123def456gh",
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-
cluster:myglobalcluster",
    "Status": "available",
    "Engine": "aurora-postgresql",
    "EngineVersion": "10.11",
    "StorageEncrypted": true,
    "DeletionProtection": false,
    "GlobalClusterMembers": [
      {
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:js-
global-cluster",
        "Readers": [
          "arn:aws:rds:us-west-2:123456789012:cluster:DB-1"
        ]
      }
    ]
  }
}
```

```

        "IsWriter": true
      },
      {
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:DB-1",
        "Readers": [],
        "IsWriter": false,
        "GlobalWriteForwardingStatus": "disabled"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Entfernen eines Clusters aus einer globalen Amazon Aurora Aurora-Datenbank](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [RemoveFromGlobalCluster](#) unter AWS CLI Befehlsreferenz.

remove-option-from-option-group

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-option-from-option-group`.

AWS CLI

Um eine Option aus einer Optionsgruppe zu löschen

Im folgenden `remove-option-from-option-group` Beispiel wird die `OEM` Option von `myoptiongroup` entfernt.

```

aws rds remove-option-from-option-group \
  --option-group-name myoptiongroup \
  --options OEM \
  --apply-immediately

```

Ausgabe:

```

{
  "OptionGroup": {
    "OptionGroupName": "myoptiongroup",
    "OptionGroupDescription": "Test",
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "19",
    "Options": [],

```



```
"AllowsVpcAndNonVpcInstanceMemberships": true,  
  "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"  
}
```

Weitere Informationen finden Sie unter [Entfernen einer Option aus einer Optionsgruppe](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [RemoveOptionFromOptionGroup](#) unter AWS CLI Befehlsreferenz.

remove-role-from-db-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-role-from-db-cluster`.

AWS CLI

So trennen Sie die Zuordnung einer AWS Identity and Access Management (IAM) -Rolle zu einem DB-Cluster

Im folgenden `remove-role-from-db-cluster` Beispiel wird eine Rolle aus einem DB-Cluster entfernt.

```
aws rds remove-role-from-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Zuordnen einer IAM Rolle zu einem Amazon Aurora MySQL DB-Cluster](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [RemoveRoleFromDbCluster](#) in der AWS CLI Befehlsreferenz.

remove-role-from-db-instance

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-role-from-db-instance`.

AWS CLI

So trennen Sie die Zuordnung einer AWS Identity and Access Management (IAM) -Rolle zu einer DB-Instance

Im folgenden `remove-role-from-db-instance` Beispiel wird die angegebene Rolle `rds-s3-integration-role` aus einer Oracle-DB-Instance mit dem Namen `test-instance` entfernt.

```
aws rds remove-role-from-db-instance \  
  --db-instance-identifier test-instance \  
  --feature-name S3_INTEGRATION \  
  --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deaktivierung der RDS SQL Serverintegration mit S3](#) im RDS Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [RemoveRoleFromDbInstance](#) in der AWS CLI Befehlsreferenz.

remove-source-identifier-from-subscription

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-source-identifier-from-subscription`.

AWS CLI

Um eine Quell-ID aus einem Abonnement zu entfernen

Im folgenden `remove-source-identifier` Beispiel wird die angegebene Quell-ID aus einem vorhandenen Abonnement entfernt.

```
aws rds remove-source-identifier-from-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

Ausgabe:

```
{  
  "EventSubscription": {  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-  
events",  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ],  
  },  
}
```

```

    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
    "Status": "modifying",
    "CustSubscriptionId": "my-instance-events",
    "CustomerAwsId": "123456789012",
    "SourceIdsList": [
        "test-instance"
    ],
    "SourceType": "db-instance",
    "Enabled": false
}
}

```

- API-Einheiten finden Sie [RemoveSourceIdentifierFromSubscription](#) unter AWS CLI Befehlsreferenz.

remove-tags-from-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `remove-tags-from-resource`.

AWS CLI

Um Tags aus einer Ressource zu entfernen

Im folgenden `remove-tags-from-resource` Beispiel werden Tags aus einer Ressource entfernt.

```

aws rds remove-tags-from-resource \
  --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \
  --tag-keys Name Environment

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging RDS Amazon-Ressourcen](#) im RDS Amazon-Benutzerhandbuch und [Tagging RDS Amazon-Ressourcen im Amazon Aurora-Benutzerhandbuch](#).

- API-Einheiten finden Sie [RemoveTagsFromResource](#) in der AWS CLI Befehlsreferenz.

reset-db-cluster-parameter-group

Das folgende Codebeispiel zeigt, wie man es benutzt `reset-db-cluster-parameter-group`.

AWS CLI

Beispiel 1: Um alle Parameter auf ihre Standardwerte zurückzusetzen

Im folgenden `reset-db-cluster-parameter-group` Beispiel werden alle Parameterwerte in einer vom Kunden erstellten DB-Cluster-Parametergruppe auf ihre Standardwerte zurückgesetzt.

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclpg \  
  --reset-all-parameters
```

Ausgabe:

```
{  
  "DBClusterParameterGroupName": "mydbclpg"  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Benutzerhandbuch.

Beispiel 2: Um bestimmte Parameter auf ihre Standardwerte zurückzusetzen

Im folgenden `reset-db-cluster-parameter-group` Beispiel werden die Parameterwerte für bestimmte Parameter auf ihre Standardwerte in einer vom Kunden erstellten DB-Cluster-Parametergruppe zurückgesetzt.

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclpgy \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
  "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Ausgabe:

```
{  
  "DBClusterParameterGroupName": "mydbclpg"  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [ResetDbClusterParameterGroup](#) unter AWS CLI Befehlsreferenz.

reset-db-parameter-group

Das folgende Codebeispiel zeigt, wie man es benutzt `reset-db-parameter-group`.

AWS CLI

Beispiel 1: Um alle Parameter auf ihre Standardwerte zurückzusetzen

Im folgenden `reset-db-parameter-group` Beispiel werden alle Parameterwerte in einer vom Kunden erstellten DB-Parametergruppe auf ihre Standardwerte zurückgesetzt.

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --reset-all-parameters
```

Ausgabe:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen](#) im RDS Amazon-Benutzerhandbuch und [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora-Benutzerhandbuch.

Beispiel 2: Um bestimmte Parameter auf ihre Standardwerte zurückzusetzen

Im folgenden `reset-db-parameter-group` Beispiel werden die Parameterwerte für bestimmte Parameter auf ihre Standardwerte in einer vom Kunden erstellten DB-Parametergruppe zurückgesetzt.

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
               "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Ausgabe:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit DB-Parametergruppen](#) im RDSAmazon-Benutzerhandbuch und [Arbeiten mit DB-Parametergruppen und DB-Cluster-Parametergruppen](#) im Amazon Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [ResetDbParameterGroup](#) in der AWS CLI Befehlsreferenz.

restore-db-cluster-from-s3

Das folgende Codebeispiel zeigt, wie man es benutzt `restore-db-cluster-from-s3`.

AWS CLI

So stellen Sie einen Amazon Aurora DB-Cluster von Amazon S3 wieder her

Das folgende `restore-db-cluster-from-s3` Beispiel stellt einen mit Amazon Aurora MySQL Version 5.7 kompatiblen DB-Cluster aus einer MySQL 5.7 DB-Backup-Datei in Amazon S3 wieder her.

```
aws rds restore-db-cluster-from-s3 \  
  --db-cluster-identifier cluster-s3-restore \  
  --engine aurora-mysql \  
  --master-username admin \  
  --master-user-password mypassword \  
  --s3-bucket-name mybucket \  
  --s3-prefix test-backup \  
  --s3-ingestion-role-arn arn:aws:iam::123456789012:role/service-role/TestBackup \  
  --source-engine mysql \  
  --source-engine-version 5.7.28
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 1,
```

```

    "DBClusterIdentifier": "cluster-s3-restore",
    "DBClusterParameterGroup": "default.aurora-mysql5.7",
    "DBSubnetGroup": "default",
    "Status": "creating",
    "Endpoint": "cluster-s3-restore.cluster-co3xyzabc123.us-
west-2.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-s3-restore.cluster-ro-co3xyzabc123.us-
west-2.rds.amazonaws.com",
    "MultiAZ": false,
    "Engine": "aurora-mysql",
    "EngineVersion": "5.7.12",
    "Port": 3306,
    "MasterUsername": "admin",
    "PreferredBackupWindow": "11:15-11:45",
    "PreferredMaintenanceWindow": "thu:12:19-thu:12:49",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-#####",
        "Status": "active"
      }
    ],
    "HostedZoneId": "Z1PVIIF0EXAMPLE",
    "StorageEncrypted": false,
    "DbClusterResourceId": "cluster-SU5THYQQH0WCXZZDGXREXAMPLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:cluster-s3-
restore",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2020-07-27T14:22:08.095Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": false,
    "CrossAccountClone": false,
    "DomainMemberships": []
  }
}

```

Weitere Informationen finden Sie unter [Migrieren von Daten aus My SQL mithilfe eines Amazon S3 S3-Buckets](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie unter [RestoreDbClusterFromS3](#) in der AWS CLI Befehlsreferenz.

restore-db-cluster-from-snapshot

Das folgende Codebeispiel zeigt die Verwendung `restore-db-cluster-from-snapshot`.

AWS CLI

Um einen DB-Cluster aus einem Snapshot wiederherzustellen

Im Folgenden wird ein Aurora SQL Postgre-DB-Cluster, der mit Postgre SQL Version 10.7 kompatibel ist, aus einem DB-Cluster-Snapshot mit dem Namen `restore-db-cluster-from-snapshot` wiederhergestellt. `test-instance-snapshot`

```
aws rds restore-db-cluster-from-snapshot \  
  --db-cluster-identifizier newdbcluster \  
  --snapshot-identifizier test-instance-snapshot \  
  --engine aurora-postgresql \  
  --engine-version 10.7
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DatabaseName": "",  
    "DBClusterIdentifizier": "newdbcluster",  
    "DBClusterParameterGroup": "default.aurora-postgresql10",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "newdbcluster.cluster-#####.us-west-2.rds.amazonaws.com",  
    "ReaderEndpoint": "newdbcluster.cluster-ro-#####.us-  
west-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-postgresql",  
    "EngineVersion": "10.7",  
    "Port": 5432,  
    "MasterUsername": "postgres",
```



```

    "PreferredBackupWindow": "09:33-10:03",
    "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-#####",
        "Status": "active"
      }
    ],
    "HostedZoneId": "Z1PVIF0EXAMPLE",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
    "DbClusterResourceId": "cluster-5DSB5IFQDDUVAWOUWM1EXAMPLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:newdbcluster",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2020-06-05T15:06:58.634Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": false,
    "CrossAccountClone": false,
    "DomainMemberships": []
  }
}

```

Weitere Informationen finden Sie unter [Wiederherstellen aus einem DB-Cluster-Snapshot](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [RestoreDbClusterFromSnapshot](#) unter AWS CLI Befehlsreferenz.

restore-db-cluster-to-point-in-time

Das folgende Codebeispiel zeigt, wie man es benutzt `restore-db-cluster-to-point-in-time`.

AWS CLI

Um einen DB-Cluster auf einen bestimmten Zeitpunkt wiederherzustellen

Im folgenden `restore-db-cluster-to-point-in-time` Beispiel wird der angegebene DB-Cluster `database-4` auf den spätestmöglichen Zeitpunkt wiederhergestellt. Bei Verwendung

copy-on-write als Wiederherstellungstyp wird der neue DB-Cluster als Klon des Quell-DB-Clusters wiederhergestellt.

```
aws rds restore-db-cluster-to-point-in-time \  
  --source-db-cluster-identifler database-4 \  
  --db-cluster-identifler sample-cluster-clone \  
  --restore-type copy-on-write \  
  --use-latest-restorable-time
```

Ausgabe:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "sample-cluster-clone",  
    "DBClusterParameterGroup": "default.aurora-postgresql10",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "sample-cluster-clone.cluster-#####.us-  
west-2.rds.amazonaws.com",  
    "ReaderEndpoint": "sample-cluster-clone.cluster-ro-#####.us-  
west-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-postgresql",  
    "EngineVersion": "10.7",  
    "Port": 5432,  
    "MasterUsername": "postgres",  
    "PreferredBackupWindow": "09:33-10:03",  
    "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",  
    "ReadReplicaIdentifiers": [],  
    "DBClusterMembers": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-#####",  
        "Status": "active"  
      }  
    ]  
  }  
}
```

```
    ],
    "HostedZoneId": "Z1PVIIF0EXAMPLE",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
    "DbClusterResourceId": "cluster-BIZ77GDSA2XBSTNPFW1EXAMPLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
clone",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "CloneGroupId": "8d19331a-099a-45a4-b4aa-11aa22bb33cc44dd",
    "ClusterCreateTime": "2020-03-10T19:57:38.967Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": false,
    "CrossAccountClone": false
  }
}
```

Weitere Informationen finden Sie unter [Wiederherstellen eines DB-Clusters zu einem bestimmten Zeitpunkt](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [RestoreDbClusterToPointInTime](#) unter AWS CLI Befehlsreferenz.

restore-db-instance-from-db-snapshot

Das folgende Codebeispiel zeigt, wie man es benutzt `restore-db-instance-from-db-snapshot`.

AWS CLI

Um eine DB-Instance aus einem DB-Snapshot wiederherzustellen

Im folgenden `restore-db-instance-from-db-snapshot` Beispiel wird aus dem angegebenen DB-Snapshot eine neue `db.t3.small` DB-Instance `db7-new-instance` mit dem Namen der DB-Instance-Klasse erstellt. Die Quell-DB-Instance, von der der Snapshot erstellt wurde, verwendet eine veraltete DB-Instance-Klasse, sodass Sie sie nicht aktualisieren können.

```
aws rds restore-db-instance-from-db-snapshot \
  --db-instance-identifizier db7-new-instance \
  --db-snapshot-identifizier db7-test-snapshot \
```

```
--db-instance-class db.t3.small
```

Ausgabe:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "db7-new-instance",
    "DBInstanceClass": "db.t3.small",
    "Engine": "mysql",
    "DBInstanceStatus": "creating",

    ...output omitted...

    "PreferredMaintenanceWindow": "mon:07:37-mon:08:07",
    "PendingModifiedValues": {},
    "MultiAZ": false,
    "EngineVersion": "5.7.22",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",

    ...output omitted...

    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:db7-new-instance",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": []
  }
}
```

Weitere Informationen finden Sie unter [Aus einem DB-Snapshot wiederherstellen](#) im RDSAmazon-Benutzerhandbuch.

- API-Einheiten finden Sie [RestoreDbInstanceFromDbSnapshot](#) unter AWS CLI Befehlsreferenz.

restore-db-instance-from-s3

Das folgende Codebeispiel zeigt, wie man es benutzt `restore-db-instance-from-s3`.

AWS CLI

So stellen Sie eine DB-Instance aus einem Backup in Amazon S3 wieder her

Das folgende `restore-db-instance-from-s3` Beispiel erstellt eine neue DB-Instance, die `restored-test-instance` anhand eines vorhandenen Backups im `my-backups` S3-Bucket benannt wird.

```
aws rds restore-db-instance-from-s3 \  
  --db-instance-identifier restored-test-instance \  
  --allocated-storage 250 --db-instance-class db.m4.large --engine mysql \  
  --master-username master --master-user-password secret99 \  
  --s3-bucket-name my-backups --s3-ingestion-role-  
arn arn:aws:iam::123456789012:role/my-role \  
  --source-engine mysql --source-engine-version 5.6.27
```

- API-Einheiten finden Sie unter [RestoreDbInstanceFromS3](#) in der AWS CLI Befehlsreferenz.

`restore-db-instance-to-point-in-time`

Das folgende Codebeispiel zeigt die Verwendung `restore-db-instance-to-point-in-time`.

AWS CLI

Beispiel 1: Um eine DB-Instance auf einen bestimmten Zeitpunkt zurückzusetzen

Im folgenden `restore-db-instance-to-point-in-time` Beispiel wird `test-instance` zum angegebenen Zeitpunkt eine neue DB-Instance mit dem Namen `restored-test-instance` wiederhergestellt.

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-identifier test-instance \  
  --target-db-instance restored-test-instance \  
  --restore-time 2018-07-30T23:45:00.000Z
```

Ausgabe:

```
{  
  "DBInstance": {  
    "AllocatedStorage": 20,
```

```

    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-
instance",
    "DBInstanceStatus": "creating",
    "DBInstanceIdentifier": "restored-test-instance",
    ...some output omitted...
  }
}

```

Weitere Informationen finden Sie unter [Wiederherstellen einer DB-Instance zu einem bestimmten Zeitpunkt](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: So stellen Sie eine DB-Instance aus einem replizierten Backup zu einem bestimmten Zeitpunkt wieder her

Im folgenden `restore-db-instance-to-point-in-time` Beispiel wird eine Oracle-DB-Instance aus einem replizierten automatisierten Backup zum angegebenen Zeitpunkt wiederhergestellt.

```

aws rds restore-db-instance-to-point-in-time \
  --source-db-instance-automated-backups-arn "arn:aws:rds:us-
west-2:123456789012:auto-backup:ab-jkib2gfg5rv7replzadausbrktni2bn4example" \
  --target-db-instance-identifier myorclinstance-from-replicated-backup \
  --restore-time 2020-12-08T18:45:00.000Z

```

Ausgabe:

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "myorclinstance-from-replicated-backup",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "oracle-se2",
    "DBInstanceStatus": "creating",
    "MasterUsername": "admin",
    "DBName": "ORCL",
    "AllocatedStorage": 20,
    "PreferredBackupWindow": "07:45-08:15",
    "BackupRetentionPeriod": 14,
    ... some output omitted ...
    "DbiResourceId": "db-KGLXG75BGVWIKQT7NQ4EXAMPLE",
    "CACertificateIdentifier": "rds-ca-2019",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
  }
}

```

```

    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:myorclinstance-from-
replicated-backup",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": [],
    "TagList": []
  }
}

```

Weitere Informationen finden Sie unter [Wiederherstellung zu einem bestimmten Zeitpunkt aus einem replizierten Backup](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [RestoreDbInstanceToPointInTime](#) in der AWS CLI Befehlsreferenz.

start-activity-stream

Das folgende Codebeispiel zeigt, wie man es benutzt `start-activity-stream`.

AWS CLI

Um einen Datenbank-Aktivitätsstream zu starten

Das folgende `start-activity-stream` Beispiel startet einen asynchronen Aktivitätsstream zur Überwachung eines Aurora-Clusters mit dem Namen `my-pg-cluster`.

```

aws rds start-activity-stream \
  --region us-east-1 \
  --mode async \
  --kms-key-id arn:aws:kms:us-east-1:123456789012:key/a12c345d-6ef7-890g-
h123-456i789jk0l1 \
  --resource-arn arn:aws:rds:us-east-1:123456789012:cluster:my-pg-cluster \
  --apply-immediately

```

Ausgabe:

```

{
  "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2N0PQ3R4S",
  "Status": "starting",
  "Mode": "async",
}

```

```
"ApplyImmediately": true
}
```

Weitere Informationen finden Sie unter [Starten eines Datenbank-Aktivitätsstreams](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [StartActivityStream](#) unter AWS CLI Befehlsreferenz.

start-db-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `start-db-cluster`.

AWS CLI

Um einen DB-Cluster zu starten

Im folgenden `start-db-cluster` Beispiel werden ein DB-Cluster und seine DB-Instances gestartet.

```
aws rds start-db-cluster \
  --db-cluster-identifier mydbcluster
```

Ausgabe:

```
{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1e",
      "us-east-1b"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydb",
    "DBClusterIdentifier": "mydbcluster",
    "...some output truncated..."
  }
}
```

Weitere Informationen finden Sie unter [Stoppen und Starten eines Amazon Aurora Aurora-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [StartDbCluster](#) unter AWS CLI Befehlsreferenz.

start-db-instance-automated-backups-replication

Das folgende Codebeispiel zeigt, wie man es benutzt `start-db-instance-automated-backups-replication`.

AWS CLI

Um regionsübergreifende automatische Backups zu aktivieren

Das folgende `start-db-instance-automated-backups-replication` Beispiel repliziert automatisierte Backups von einer DB-Instance in der Region USA Ost (Nord-Virginia) nach USA West (Oregon). Die Aufbewahrungsfrist für Backups beträgt 14 Tage.

```
aws rds start-db-instance-automated-backups-replication \  
  --region us-west-2 \  
  --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db" \  
  --backup-retention-period 14
```

Ausgabe:

```
{  
  "DBInstanceAutomatedBackup": {  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",  
    "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",  
    "Region": "us-east-1",  
    "DBInstanceIdentifier": "new-orcl-db",  
    "RestoreWindow": {},  
    "AllocatedStorage": 20,  
    "Status": "pending",  
    "Port": 1521,  
    "InstanceCreateTime": "2020-12-04T15:28:31Z",  
    "MasterUsername": "admin",  
    "Engine": "oracle-se2",  
    "EngineVersion": "12.1.0.2.v21",  
    "LicenseModel": "bring-your-own-license",  
    "OptionGroupName": "default:oracle-se2-12-1",  
    "Encrypted": false,  
    "StorageType": "gp2",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "BackupRetentionPeriod": 14,  
    "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfg5rv7replzadabrktni2bn4example"  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Aktivieren regionsübergreifender automatisierter Backups](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [StartDbInstanceAutomatedBackupsReplication](#) in der AWS CLI Befehlsreferenz.

start-db-instance

Das folgende Codebeispiel zeigt, wie man es benutzt `start-db-instance`.

AWS CLI

Um eine DB-Instance zu starten

Das folgende `start-db-instance` Beispiel startet die angegebene DB-Instance.

```
aws rds start-db-instance \  
  --db-instance-identifizier test-instance
```

Ausgabe:

```
{  
  "DBInstance": {  
    "DBInstanceStatus": "starting",  
    ...some output truncated...  
  }  
}
```

- APIEinzelheiten finden Sie [StartDbInstance](#) unter AWS CLI Befehlsreferenz.

start-export-task

Das folgende Codebeispiel zeigt, wie man es benutzt `start-export-task`.

AWS CLI

Um einen Snapshot nach Amazon S3 zu exportieren

Das folgende `start-export-task` Beispiel exportiert einen DB-Snapshot mit dem Namen `db5-snapshot-test` in den Amazon S3 S3-Bucket mit dem Namen `mybucket`.

```
aws rds start-export-task \  
  --export-task-identifizier my-s3-export \  
  --source-arn arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test \  
  --s3-bucket-name mybucket \  
  --iam-role-arn arn:aws:iam::123456789012:role/service-role/ExportRole \  
  --kms-key-id arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-  
  aabbccddeeff
```

Ausgabe:

```
{  
  "ExportTaskIdentifizier": "my-s3-export",  
  "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test",  
  "SnapshotTime": "2020-03-27T20:48:42.023Z",  
  "S3Bucket": "mybucket",  
  "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",  
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-  
  aabbccddeeff",  
  "Status": "STARTING",  
  "PercentProgress": 0,  
  "TotalExtractedDataInGB": 0  
}
```

Weitere Informationen finden Sie unter [Exportieren eines Snapshots in einen Amazon S3 S3-Bucket](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [StartExportTask](#) in der AWS CLI Befehlsreferenz.

stop-activity-stream

Das folgende Codebeispiel zeigt, wie man es benutzt stop-activity-stream.

AWS CLI

Um einen Datenbank-Aktivitätsstream zu stoppen

Das folgende stop-activity-stream Beispiel stoppt einen Aktivitätsstream in einem Aurora-Cluster mit dem Namen my-pg-cluster.

```
aws rds stop-activity-stream \  
  --region us-east-1 \  
  --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \  
  \
```

```
--apply-immediately
```

Ausgabe:

```
{
  "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk011",
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2N0PQ3R4S",
  "Status": "stopping"
}
```

Weitere Informationen finden Sie unter [Stoppen eines Activity Streams](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie [StopActivityStream](#) in der AWS CLI Befehlsreferenz.

stop-db-cluster

Das folgende Codebeispiel zeigt, wie man es benutzt `stop-db-cluster`.

AWS CLI

Um einen DB-Cluster zu stoppen

Das folgende `stop-db-cluster` Beispiel stoppt einen DB-Cluster und seine DB-Instances.

```
aws rds stop-db-cluster \  
  --db-cluster-identifizier mydbcluster
```

Ausgabe:

```
{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1e",
      "us-east-1b"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydb",
```

```

    "DBClusterIdentifier": "mydbcluster",
    ...some output truncated...
  }
}

```

Weitere Informationen finden Sie unter [Stoppen und Starten eines Amazon Aurora Aurora-DB-Clusters](#) im Amazon Aurora Benutzerhandbuch.

- API-Einheiten finden Sie [StopDbCluster](#) unter AWS CLI Befehlsreferenz.

stop-db-instance-automated-backups-replication

Das folgende Codebeispiel zeigt, wie man es benutzt `stop-db-instance-automated-backups-replication`.

AWS CLI

Um die Replikation automatisierter Backups zu beenden

Mit dem Folgenden `stop-db-instance-automated-backups-replication` wird die Replikation automatisierter Backups in die Region USA West (Oregon) beendet. Replizierte Backups werden gemäß dem festgelegten Aufbewahrungszeitraum für Backups aufbewahrt.

```

aws rds stop-db-instance-automated-backups-replication \
  --region us-west-2 \
  --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db"

```

Ausgabe:

```

{
  "DBInstanceAutomatedBackup": {
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
    "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
    "Region": "us-east-1",
    "DBInstanceIdentifier": "new-orcl-db",
    "RestoreWindow": {
      "EarliestTime": "2020-12-04T23:13:21.030Z",
      "LatestTime": "2020-12-07T19:59:57Z"
    },
    "AllocatedStorage": 20,
    "Status": "replicating",
    "Port": 1521,
  }
}

```

```
"InstanceCreateTime": "2020-12-04T15:28:31Z",
"MasterUsername": "admin",
"Engine": "oracle-se2",
"EngineVersion": "12.1.0.2.v21",
"LicenseModel": "bring-your-own-license",
"OptionGroupName": "default:oracle-se2-12-1",
"Encrypted": false,
"StorageType": "gp2",
"IAMDatabaseAuthenticationEnabled": false,
"BackupRetentionPeriod": 7,
"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadabrktni2bn4example"
}
}
```

Weitere Informationen finden Sie unter [Stoppen der automatisierten Backup-Replikation](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie [StopDbInstanceAutomatedBackupsReplication](#) in der AWS CLI Befehlsreferenz.

stop-db-instance

Das folgende Codebeispiel zeigt, wie man es benutzt `stop-db-instance`.

AWS CLI

Um eine DB-Instance zu stoppen

Das folgende `stop-db-instance` Beispiel stoppt die angegebene DB-Instance.

```
aws rds stop-db-instance \
  --db-instance-identifier test-instance
```

Ausgabe:

```
{
  "DBInstance": {
    "DBInstanceStatus": "stopping",
    ...some output truncated...
  }
}
```

- API-Einheiten finden Sie [StopDbInstance](#) unter AWS CLI Befehlsreferenz.

switchover-blue-green-deployment

Das folgende Codebeispiel zeigt, wie man es benutzt `switchover-blue-green-deployment`.

AWS CLI

Beispiel 1: Um eine blaue/grüne Bereitstellung für eine RDS DB-Instance zu ändern

Im folgenden `switchover-blue-green-deployment` Beispiel wird die angegebene grüne Umgebung als neue Produktionsumgebung beworben.

```
aws rds switchover-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \  
  --switchover-timeout 300
```

Ausgabe:

```
{  
  "BlueGreenDeployment": {  
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",  
    "BlueGreenDeploymentName": "bgd-cli-test-instance",  
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",  
    "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-  
blhile",  
    "SwitchoverDetails": [  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-green-blhile",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-1",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-  
instance-replica-1-green-k5fv7u",  
        "Status": "AVAILABLE"  
      },  
      {
```

```

        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ggsh8m",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-o2vwm0",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
    }
],
"Status": "SWITCHOVER_IN_PROGRESS",
"CreateTime": "2022-02-25T22:33:22.225000+00:00"
}
}

```

Weitere Informationen finden Sie unter [Umstellung auf eine blaue/grüne Bereitstellung](#) im RDSAmazon-Benutzerhandbuch.

Beispiel 2: Um eine blaue/grüne Bereitstellung für einen Aurora My SQL DB-Cluster zu bewerben

Im folgenden `switchover-blue-green-deployment` Beispiel wird die angegebene grüne Umgebung als neue Produktionsumgebung beworben.

```
aws rds switchover-blue-green-deployment \  
  --blue-green-deployment-identifizier bgd-wi89nwzglccsfake \  
  --switchover-timeout 300
```

Ausgabe:

```
{  
  "BlueGreenDeployment": {  
    "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",  
    "BlueGreenDeploymentName": "my-blue-green-deployment",  
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-  
cluster",  
    "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-  
cluster-green-3ud8z6",  
    "SwitchoverDetails": [  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-  
aurora-mysql-cluster",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-  
aurora-mysql-cluster-green-3ud8z6",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-1",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-1-green-bvxc73",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-2",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-2-green-7wc4ie",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-3",
```

```
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-p4xxkz",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-np1ik1",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-miszlf",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],
"Status": "SWITCHOVER_IN_PROGRESS",
"CreateTime": "2022-02-25T22:38:49.522000+00:00"
}
}
```

Weitere Informationen finden Sie unter [Umstellung auf eine blaue/grüne Bereitstellung](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- API-Einheiten finden Sie unter [SwitchoverBlueGreenDeployment AWS CLI Befehlsreferenz](#).

Beispiele für Amazon RDS Data Service mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon RDS Data Service Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-execute-statement

Das folgende Codebeispiel zeigt die Verwendung `batch-execute-statement`.

AWS CLI

Um eine SQL Batch-Anweisung auszuführen

Im folgenden `batch-execute-statement` Beispiel wird eine SQL Batch-Anweisung über ein Datenarray mit einem Parametersatz ausgeführt.

```
aws rds-data batch-execute-statement \
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
  --database "mydb" \
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
  --sql "insert into mytable values (:id, :val)" \
  --parameter-sets "[[{"name": "id", "value": {"longValue": 1}}, {"name":
  "val", "value": {"stringValue": "ValueOne"}}],
```

```
[{"name": "id", "value": {"longValue": 2}}, {"name": "val", "value": {"stringValue": "ValueTwo"}}, {"name": "id", "value": {"longValue": 3}}, {"name": "val", "value": {"stringValue": "ValueThree"}}]
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Using the Data API for Aurora Serverless](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [BatchExecuteStatement AWS CLI](#) Befehlsreferenz.

begin-transaction

Das folgende Codebeispiel zeigt die Verwendung begin-transaction.

AWS CLI

Um eine SQL Transaktion zu starten

Das folgende begin-transaction Beispiel startet eine SQL Transaktion.

```
aws rds-data begin-transaction \  
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
  --database "mydb" \  
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret"
```

Ausgabe:

```
{  
  "transactionId": "ABC1234567890xyz"  
}
```

Weitere Informationen finden Sie unter [Using the Data API for Aurora Serverless](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [BeginTransaction AWS CLI](#) Befehlsreferenz.

commit-transaction

Das folgende Codebeispiel zeigt die Verwendung commit-transaction.

AWS CLI

Um eine SQL Transaktion festzuschreiben

Im folgenden `commit-transaction` Beispiel wird die angegebene SQL Transaktion beendet und die Änderungen, die Sie als Teil der Transaktion vorgenommen haben, festgeschrieben.

```
aws rds-data commit-transaction \  
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \  
  --transaction-id "ABC1234567890xyz"
```

Ausgabe:

```
{  
  "transactionStatus": "Transaction Committed"  
}
```

Weitere Informationen finden Sie unter [Using the Data API for Aurora Serverless](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [CommitTransaction AWS CLIBefehlsreferenz](#).

execute-statement

Das folgende Codebeispiel zeigt die Verwendung `execute-statement`.

AWS CLI

Beispiel 1: Um eine SQL Anweisung auszuführen, die Teil einer Transaktion ist

Im folgenden `execute-statement` Beispiel wird eine SQL Anweisung ausgeführt, die Teil einer Transaktion ist.

```
aws rds-data execute-statement \  
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
  --database "mydb" \  
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \  
  --sql "update mytable set quantity=5 where id=201" \  
  --transaction-id "ABC1234567890xyz"
```

Ausgabe:

```
{
  "numberOfRecordsUpdated": 1
}
```

Beispiel 2: Um eine SQL Anweisung mit Parametern auszuführen

Im folgenden `execute-statement` Beispiel wird eine SQL Anweisung mit Parametern ausgeführt.

```
aws rds-data execute-statement \
  --resource-arn "arn:aws:rds:us-east-1:123456789012:cluster:mydbcluster" \
  --database "mydb" \
  --secret-arn "arn:aws:secretsmanager:us-east-1:123456789012:secret:mysecret" \
  --sql "insert into mytable values (:id, :val)" \
  --parameters "[{\"name\": \"id\", \"value\": {\"longValue\": 1}}, {\"name\": \"val\", \"value\": {\"stringValue\": \"value1\"}}]"
```

Ausgabe:

```
{
  "numberOfRecordsUpdated": 1
}
```

Weitere Informationen finden Sie unter [Using the Data API for Aurora Serverless](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [ExecuteStatement AWS CLI](#) Befehlsreferenz.

rollback-transaction

Das folgende Codebeispiel zeigt die Verwendung `rollback-transaction`.

AWS CLI

Um eine SQL Transaktion rückgängig zu machen

Im folgenden `rollback-transaction` Beispiel wird die angegebene SQL Transaktion rückgängig gemacht.

```
aws rds-data rollback-transaction \
```

```
--resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
--secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \  
--transaction-id "ABC1234567890xyz"
```

Ausgabe:

```
{  
  "transactionStatus": "Rollback Complete"  
}
```

Weitere Informationen finden Sie unter [Using the Data API for Aurora Serverless](#) im RDSAmazon-Benutzerhandbuch.

- APIEinzelheiten finden Sie unter [RollbackTransaction AWS CLI](#) Befehlsreferenz.

Beispiele für Amazon RDS Performance Insights mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon RDS Performance Insights Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-dimension-keys

Das folgende Codebeispiel zeigt die Verwendung `describe-dimension-keys`.

AWS CLI

Um Dimensionsschlüssel zu beschreiben

In diesem Beispiel werden die Namen aller Warteereignisse abgefragt. Die Daten werden nach dem Namen des Ereignisses und den Aggregatwerten dieser Ereignisse über den angegebenen Zeitraum zusammengefasst.

Befehl:

```
aws pi describe-dimension-keys --service-type RDS --identifier db-LKCG0BK26374TPTDFX0IWWCPPM --start-time 1527026400 --end-time 1527080400 --metric db.load.avg --group-by '{"Group": "db.wait_event"}
```

Ausgabe:

```
{
  "AlignedEndTime": 1.5270804E9,
  "AlignedStartTime": 1.5270264E9,
  "Keys": [
    {
      "Dimensions": {"db.wait_event.name": "wait/synch/mutex/innodb/aurora_lock_thread_slot_futex"},
      "Total": 0.05906906851195666
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/io/aurora_redo_log_flush"},
      "Total": 0.015824722186149193
    },
    {
      "Dimensions": {"db.wait_event.name": "CPU"},
      "Total": 0.008014396230265477
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/io/aurora_respond_to_client"},
      "Total": 0.0036361612526204477
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/io/table/sql/handler"},
      "Total": 0.0019108398419382965
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/synch/cond/mysys/my_thread_var::suspend"},
      "Total": 8.533847837782684E-4
    },
  ],
}
```



```

    {
      "Dimensions": {"db.wait_event.name": "wait/io/file/csv/data"},
      "Total": 6.864181956477376E-4
    },
    {
      "Dimensions": {"db.wait_event.name": "Unknown"},
      "Total": 3.895887056379051E-4
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/synch/mutex/sql/
FILE_AS_TABLE::LOCK_shim_lists"},
      "Total": 3.710368625122906E-5
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/lock/table/sql/handler"},
      "Total": 0
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeDimensionKeys](#) unter AWS CLI Befehlsreferenz.

get-resource-metrics

Das folgende Codebeispiel zeigt die Verwendung `get-resource-metrics`.

AWS CLI

Um Ressourcenmetriken abzurufen

In diesem Beispiel werden Datenpunkte für die Dimensionsgruppe `db.wait_event` und für die Dimension `db.wait_event.name` innerhalb dieser Gruppe angefordert. In der Antwort werden die relevanten Datenpunkte nach der angeforderten Dimension (`db.wait_event.name`) gruppiert.

Befehl:

```

aws pi get-resource-metrics --service-type RDS --identifier db-
LKCGBK26374TPTDFX0IWVCPM --start-time 1527026400 --end-time 1527080400 --period-
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json

```

Die Argumente für `--metric-queries` werden in einer Datei gespeichert, `JSON metric-queries.json`. Hier ist der Inhalt dieser Datei:

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": {
      "Group": "db.wait_event"
    }
  }
]
```

Ausgabe:

```
{
  "AlignedEndTime": 1.5270804E9,
  "AlignedStartTime": 1.5270264E9,
  "Identifier": "db-LKCG0BK26374TPTDFX0IWVCPM",
  "MetricList": [
    {
      "Key": {
        "Metric": "db.load.avg"
      },
      "DataPoints": [
        {
          "Timestamp": 1527026700.0,
          "Value": 1.3533333333333333
        },
        {
          "Timestamp": 1527027000.0,
          "Value": 0.88
        },
        <...remaining output omitted...>
      ]
    },
    {
      "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
          "db.wait_event.name": "wait/synch/mutex/innodb/aurora_lock_thread_slot_futex"
        }
      },
      "DataPoints": [
        {
          "Timestamp": 1527026700.0,
```

```

        "Value": 0.8566666666666667
      },
      {
        "Timestamp": 1527027000.0,
        "Value": 0.8633333333333333
      },
      <...remaining output omitted...>
    ],
  },
  <...remaining output omitted...>
]
}

```

- API-Einheiten finden Sie [GetResourceMetrics](#) in der AWS CLI Befehlsreferenz.

Amazon Redshift Redshift-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Redshift Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-reserved-node-exchange

Das folgende Codebeispiel zeigt die Verwendung `accept-reserved-node-exchange`.

AWS CLI

Um den Austausch reservierter Knoten zu akzeptieren

Das folgende `accept-reserved-node-exchange` Beispiel akzeptiert den Austausch eines DC1 reservierten Knotens gegen einen DC2 reservierten Knoten.

```
aws redshift accept-reserved-node-exchange /
--reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE /
--target-reserved-node-offering-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

Ausgabe:

```
{
  "ExchangedReservedNode": {
    "ReservedNodeId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
    "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
    "NodeType": "dc2.large",
    "StartTime": "2019-12-06T21:17:26Z",
    "Duration": 31536000,
    "FixedPrice": 0.0,
    "UsagePrice": 0.0,
    "CurrencyCode": "USD",
    "NodeCount": 1,
    "State": "exchanging",
    "OfferingType": "All Upfront",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": 0.0,
        "RecurringChargeFrequency": "Hourly"
      }
    ],
    "ReservedNodeOfferingType": "Regular"
  }
}
```

Weitere Informationen finden Sie unter [Upgrading Reserved Nodes With AWS CLI](#) the im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [AcceptReservedNodeExchange AWS CLI](#) Befehlsreferenz.

authorize-cluster-security-group-ingress

Das folgende Codebeispiel zeigt die Verwendung `authorize-cluster-security-group-ingress`.

AWS CLI

Ein GroupThis Beispiel für die Autorisierung des Zugriffs auf eine EC2 Sicherheitsgruppe autorisiert den Zugriff auf eine benannte EC2 Amazon-Sicherheitsgruppe.Befehl:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

Durch die Autorisierung des Zugriffs auf ein CIDR rangeThis Beispiel wird der Zugriff auf einen Bereich autorisiert.Befehl: CIDR

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --cidrip 192.168.100.100/32
```

- APIEinzelheiten finden Sie in der Befehlsreferenz [AuthorizeClusterSecurityGroupIngress](#).AWS CLI

authorize-snapshot-access

Das folgende Codebeispiel zeigt die Verwendungauthorize-snapshot-access.

AWS CLI

Ein AWS Konto zur Wiederherstellung autorisieren Ein SnapshotThis Beispiel autorisiert das AWS Konto444455556666, den Snapshot wiederherzustellen. my-snapshot-id Standardmäßig erfolgt die Ausgabe im Format.COMMAND: JSON

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

Ergebnis:

```
{
  "Snapshot": {
    "Status": "available",
    "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
    "EstimatedSecondsToCompletion": 0,
    "AvailabilityZone": "us-east-1a",
    "ClusterVersion": "1.0",
```

```

    "MasterUsername": "adminuser",
    "Encrypted": false,
    "OwnerAccount": "111122223333",
    "BackupProgressInMegabytes": 11.0,
    "ElapsedTimeInSeconds": 0,
    "DBName": "dev",
    "CurrentBackupRateInMegabytesPerSecond": 0.1534,
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "ActualIncrementalBackupSizeInMegabytes"; 11.0,
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "mycluster",
    "TotalBackupSizeInMegabytes": 20.0,
    "Port": 5439,
    "NumberOfNodes": 2,
    "SnapshotIdentifier": "my-snapshot-id"
  }
}

```

- API-Einheiten finden Sie [AuthorizeSnapshotAccess](#) in der AWS CLI Befehlsreferenz.

batch-delete-cluster-snapshots

Das folgende Codebeispiel zeigt die Verwendung `batch-delete-cluster-snapshots`.

AWS CLI

Um eine Reihe von Cluster-Snapshots zu löschen

Im folgenden `batch-delete-cluster-snapshots` Beispiel wird ein Satz manueller Cluster-Snapshots gelöscht.

```

aws redshift batch-delete-cluster-snapshots \
  --
  identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-

```

Ausgabe:

```

{
  "Resources": [
    "mycluster-2019-11-06-14-12",
    "mycluster-2019-11-06-14-20"
  ]
}

```

```
]
}
```

Weitere Informationen finden Sie unter [Amazon Redshift Snapshots](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [BatchDeleteClusterSnapshots](#).AWS CLI

batch-modify-cluster-snapshots

Das folgende Codebeispiel zeigt die Verwendung `batch-modify-cluster-snapshots`.

AWS CLI

Um eine Reihe von Cluster-Snapshots zu ändern

Im folgenden `batch-modify-cluster-snapshots` Beispiel werden die Einstellungen für eine Reihe von Cluster-Snapshots geändert.

```
aws redshift batch-modify-cluster-snapshots \
  --snapshot-identifier-list mycluster-2019-11-06-16-31 mycluster-2019-11-06-16-32 \
  --manual-snapshot-retention-period 30
```

Ausgabe:

```
{
  "Resources": [
    "mycluster-2019-11-06-16-31",
    "mycluster-2019-11-06-16-32"
  ],
  "Errors": [],
  "ResponseMetadata": {
    "RequestId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "x-amzn-requestid": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
      "content-type": "text/xml",
      "content-length": "480",
      "date": "Sat, 07 Dec 2019 00:36:09 GMT",
      "connection": "keep-alive"
    }
  },
}
```

```
    "RetryAttempts": 0
  }
}
```

Weitere Informationen finden Sie unter [Amazon Redshift Snapshots](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [BatchModifyClusterSnapshots](#).AWS CLI

cancel-resize

Das folgende Codebeispiel zeigt die Verwendung `cancel-resize`.

AWS CLI

Um die Größenänderung eines Clusters abubrechen

Im folgenden `cancel-resize` Beispiel wird ein klassischer Vorgang zur Größenänderung für einen Cluster abgebrochen.

```
aws redshift cancel-resize \  
  --cluster-identifier mycluster
```

Ausgabe:

```
{  
  "TargetNodeType": "dc2.large",  
  "TargetNumberOfNodes": 2,  
  "TargetClusterType": "multi-node",  
  "Status": "CANCELLING",  
  "ResizeType": "ClassicResize",  
  "TargetEncryptionType": "NONE"  
}
```

Weitere Informationen finden Sie unter [Resizing Clusters in Amazon Redshift](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [CancelResize](#) in der AWS CLI Befehlsreferenz.

copy-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `copy-cluster-snapshot`.

AWS CLI

Das VersionsThis Beispiel Get a Description of All Cluster gibt eine Beschreibung aller Clusterversionen zurück. Standardmäßig erfolgt die Ausgabe im JSON Format.Command:

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

Ergebnis:

```
{
  "Snapshot": {
    "Status": "available",
    "SnapshotCreateTime": "2013-01-22T19:27:58.931Z",
    "AvailabilityZone": "us-east-1c",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "DBName": "dev",
    "ClusterCreateTime": "2013-01-22T19:23:59.368Z",
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "examplecluster",
    "Port": 5439,
    "NumberOfNodes": "2",
    "SnapshotIdentifier": "my-saved-snapshot-copy"
  },
  "ResponseMetadata": {
    "RequestId": "3b279691-64e3-11e2-bec0-17624ad140dd"
  }
}
```

- APIEinzelheiten finden Sie [CopyClusterSnapshot](#) in der AWS CLI Befehlsreferenz.

create-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendungcreate-cluster-parameter-group.

AWS CLI

Das GroupThis Beispiel „Cluster-Parameter erstellen“ erstellt eine neue Cluster-Parametergruppe.Command:

```
aws redshift create-cluster-parameter-group --parameter-group-name
myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
first cluster parameter group"
```

Ergebnis:

```
{
  "ClusterParameterGroup": {
    "ParameterGroupFamily": "redshift-1.0",
    "Description": "My first cluster parameter group",
    "ParameterGroupName": "myclusterparametergroup"
  },
  "ResponseMetadata": {
    "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
  }
}
```

- API-Einheiten finden Sie [CreateClusterParameterGroup](#) in der AWS CLI Befehlsreferenz.

create-cluster-security-group

Das folgende Codebeispiel zeigt die Verwendung `create-cluster-security-group`.

AWS CLI

Durch das Erstellen eines Group-Beispiels für Clustersicherheit wird eine neue Clustersicherheitsgruppe erstellt. Die Ausgabe erfolgt standardmäßig im JSON-Format. Command:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group"
```

Ergebnis:

```
{
  "create_cluster_security_group_response": {
    "create_cluster_security_group_result": {
      "cluster_security_group": {
        "description": "This is my cluster security group",
        "owner_id": "300454760768",
        "cluster_security_group_name": "mysecuritygroup",

```

```

        "ec2_security_groups": \[],
        "ip_ranges": \[]
    }
},
"response_metadata": {
    "request_id": "5df486a0-343a-11e2-b0d8-d15d0ef48549"
}
}
}

```

Mit der Option können Sie dieselben Informationen auch im Textformat abrufen. Befehl `--output text`:

`--output text`Option.Befehl:

Option.Befehl:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group" --output text
```

Ergebnis:

```
This is my cluster security group 300454760768 mysecuritygroup
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- API-Einheiten finden Sie [CreateClusterSecurityGroup](#) in der AWS CLI Befehlsreferenz.

create-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `create-cluster-snapshot`.

AWS CLI

Das Snapshot-Beispiel „Cluster erstellen“ erstellt einen neuen Cluster-Snapshot. Standardmäßig hat die Ausgabe das JSON-Format. **COMMAND:**

```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-
identifier my-snapshot-id
```

Ergebnis:

```
{
  "Snapshot": {
    "Status": "creating",
    "SnapshotCreateTime": "2013-01-22T22:20:33.548Z",
    "AvailabilityZone": "us-east-1a",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "DBName": "dev",
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "mycluster",
    "Port": 5439,
    "NumberOfNodes": "2",
    "SnapshotIdentifier": "my-snapshot-id"
  },
  "ResponseMetadata": {
    "RequestId": "f024d1a5-64e1-11e2-88c5-53eb05787dfb"
  }
}
```

- API-Einheiten finden Sie [CreateClusterSnapshot](#) in der AWS CLI Befehlsreferenz.

create-cluster-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `create-cluster-subnet-group`.

AWS CLI

Das GroupThis Beispiel „Cluster-Subnetz erstellen“ erstellt eine neue Cluster-Subnetzgruppe.

Befehl:

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
--description "My subnet group" --subnet-ids subnet-763fdd1c
```

Ergebnis:

```
{
  "ClusterSubnetGroup": {
    "Subnets": [
      {
        "SubnetStatus": "Active",
```

```

        "SubnetIdentifier": "subnet-763fdd1c",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
        }
    } ],
    "VpcId": "vpc-7e3fdd14",
    "SubnetGroupStatus": "Complete",
    "Description": "My subnet group",
    "ClusterSubnetGroupName": "mysubnetgroup"
},
"ResponseMetadata": {
    "RequestId": "500b8ce2-698f-11e2-9790-fd67517fb6fd"
}
}

```

- API-Einheiten finden Sie in der Befehlsreferenz [CreateClusterSubnetGroup](#).AWS CLI

create-cluster

Das folgende Codebeispiel zeigt die Verwendung `create-cluster`.

AWS CLI

Das Parameterset-Beispiel `Create a Cluster with Minimal` erstellt einen Cluster mit einem minimalen Parametersatz. Standardmäßig erfolgt die Ausgabe im JSON-Format. **COMMAND:**

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

Ergebnis:

```

{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],

```

```

    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      } ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": \[],
    "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "creating",
    "ClusterIdentifier": "mycluster",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {
      "MasterUserPassword": "\*****"
    }
  },
  "ResponseMetadata": {
    "RequestId": "7cf4bcfc-64dd-11e2-bea9-49e0ce183f07"
  }
}

```

- API-Einheiten finden Sie [CreateCluster](#) in der AWS CLI Befehlsreferenz.

create-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-event-subscription`.

AWS CLI

Um ein Benachrichtigungsabonnement für ein Ereignis zu erstellen

Im folgenden `create-event-subscription` Beispiel wird ein Abonnement für Ereignisbenachrichtigungen erstellt.

```

aws redshift create-event-subscription \
  --subscription-name mysubscription \
  --sns-topic-arn arn:aws:sns:us-west-2:123456789012:MySNStopic \
  --source-type cluster \
  --source-ids mycluster

```

Ausgabe:

```
{
  "EventSubscription": {
    "CustomerAwsId": "123456789012",
    "CustSubscriptionId": "mysubscription",
    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
    "Status": "active",
    "SubscriptionCreationTime": "2019-12-09T20:05:19.365Z",
    "SourceType": "cluster",
    "SourceIdsList": [
      "mycluster"
    ],
    "EventCategoriesList": [],
    "Severity": "INFO",
    "Enabled": true,
    "Tags": []
  }
}
```

Weitere Informationen finden Sie unter [Abonnieren von Amazon Redshift-Ereignisbenachrichtigungen](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [CreateEventSubscription](#) in AWS CLI der Befehlsreferenz.

create-hsm-client-certificate

Das folgende Codebeispiel zeigt die Verwendung `create-hsm-client-certificate`.

AWS CLI

Um ein HSM Client-Zertifikat zu erstellen

Im folgenden `create-hsm-client-certificate` Beispiel wird ein HSM Client-Zertifikat generiert, mit dem ein Cluster eine Verbindung zu einem herstellen kann HSM.

```
aws redshift create-hsm-client-certificate \
  --hsm-client-certificate-identifier myhsmclientcert
```

Ausgabe:

```
{
  "HsmClientCertificate": {
```

```

    "HsmClientCertificateIdentifizier": "myhsmclientcert",
    "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----
MIICiEXAMPLECQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTEXAMPLEEwDgYDVQQHEwDTZWf0dGxLMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25EXAMPLEIwEAYDVQQDEw1UZXR0Q21sYWxHZAAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD
EXAMPLETZWF0dGxLMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAEXAMPLEEw1UZXR0Q21sYWxHZAAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKEXAMPLEAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uU5fwfEvySwTc2XADZ4nB+BLyGVIk6EXAMPLE3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugEXAMPLEzZswY6786m86gpE
Ibb30hjZncvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEEXAMPLEEAtCu4
nUhVvXyUEXAMPLEh8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLE10ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
  }
}

```

Weitere Informationen finden Sie unter [Amazon Redshift API Permissions Reference](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [CreateHsmClientCertificate AWS CLI Befehlsreferenz](#).

create-hsm-configuration

Das folgende Codebeispiel zeigt die Verwendung `create-hsm-configuration`.

AWS CLI

Um eine HSM Konfiguration zu erstellen

Im folgenden `create-hsm-configuration` Beispiel wird die angegebene HSM Konfiguration erstellt, die Informationen enthält, die ein Cluster benötigt, um Datenbankverschlüsselungsschlüssel in einem Hardware-Sicherheitsmodul (HSM) zu speichern und zu verwenden.

```

aws redshift create-hsm-configuration /
  --hsm-configuration-identifizier myhsmconnection
  --description "My HSM connection"
  --hsm-ip-address 192.0.2.09

```



```
--hsm-partition-name myhsmpartition /
--hsm-partition-password A1b2c3d4 /
--hsm-server-public-certificate myhsmclientcert
```

Ausgabe:

```
{
  "HsmConfiguration": {
    "HsmConfigurationIdentifier": "myhsmconnection",
    "Description": "My HSM connection",
    "HsmIpAddress": "192.0.2.09",
    "HsmPartitionName": "myhsmpartition",
    "Tags": []
  }
}
```

- API-Einheiten finden Sie [CreateHsmConfiguration](#) unter AWS CLI Befehlsreferenz.

create-snapshot-copy-grant

Das folgende Codebeispiel zeigt die Verwendung `create-snapshot-copy-grant`.

AWS CLI

Um einen Snapshot Copy Grant zu erstellen

Im folgenden `create-snapshot-copy-grant` Beispiel wird ein Snapshot-Kopierzuschuss erstellt und kopierte Snapshots in einer AWS Zielregion verschlüsselt.

```
aws redshift create-snapshot-copy-grant \
  --snapshot-copy-grant-name mynsnapshotcopygrantname
```

Ausgabe:

```
{
  "SnapshotCopyGrant": {
    "SnapshotCopyGrantName": "mynsnapshotcopygrantname",
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/bPxRfih3yCo8nvbEXAMPLEKEY",
    "Tags": []
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Amazon Redshift Database Encryption](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [CreateSnapshotCopyGrant AWS CLI Befehlsreferenz](#).

create-snapshot-schedule

Das folgende Codebeispiel zeigt die Verwendung `create-snapshot-schedule`.

AWS CLI

Um einen Snapshot-Zeitplan zu erstellen

Im folgenden `create-snapshot-schedule` Beispiel wird ein Snapshot-Zeitplan mit der angegebenen Beschreibung und einer Rate im Abstand von 12 Stunden erstellt.

```
aws redshift create-snapshot-schedule \  
  --schedule-definitions "rate(12 hours)" \  
  --schedule-identifier mysnapshotschedule \  
  --schedule-description "My schedule description"
```

Ausgabe:

```
{  
  "ScheduleDefinitions": [  
    "rate(12 hours)"  
  ],  
  "ScheduleIdentifier": "mysnapshotschedule",  
  "ScheduleDescription": "My schedule description",  
  "Tags": []  
}
```

Weitere Informationen finden Sie unter [Automated Snapshot Schedules](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [CreateSnapshotSchedule AWS CLI Befehlsreferenz](#).

create-tags

Das folgende Codebeispiel zeigt die Verwendung `create-tags`.

AWS CLI

Um Tags für einen Cluster zu erstellen

Im folgenden `create-tags` Beispiel wird das angegebene Tag-Schlüssel/Wert-Paar zum angegebenen Cluster hinzugefügt.

```
aws redshift create-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tags "Key"="mytags","Value"="tag1"
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Tagging Resources in Amazon Redshift](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [CreateTags](#) in AWS CLI der Befehlsreferenz.

delete-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster-parameter-group`.

AWS CLI

Das GroupThis Beispiel „Einen Cluster-Parameter löschen“ löscht eine Cluster-Parametergruppe. Command:

```
aws redshift delete-cluster-parameter-group --parameter-group-name  
myclusterparametergroup
```

- API-Einheiten finden Sie [DeleteClusterParameterGroup](#) in AWS CLI der Befehlsreferenz.

delete-cluster-security-group

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster-security-group`.

AWS CLI

Beim Löschen eines GroupThis Clustersicherheitsbeispiels wird eine Cluster-Sicherheitsgruppe gelöscht. Befehl:

```
aws redshift delete-cluster-security-group --cluster-security-group-name
mysecuritygroup
```

- API-Einheiten finden Sie [DeleteClusterSecurityGroup](#) in AWS CLI der Befehlsreferenz.

delete-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster-snapshot`.

AWS CLI

Das Snapshot-Beispiel zum Löschen eines Clusters löscht einen Cluster-Snapshot. Command:

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- API-Einheiten finden Sie in der Befehlsreferenz [DeleteClusterSnapshot](#). AWS CLI

delete-cluster-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster-subnet-group`.

AWS CLI

Das Group-Beispiel „Ein Cluster-Subnetz löschen“ löscht eine Cluster-Subnetzgruppe. Befehl:

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

Ergebnis:

```
{
  "ResponseMetadata": {
    "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
  }
}
```

- API-Einheiten finden Sie in der Befehlsreferenz. [DeleteClusterSubnetGroup](#) AWS CLI

delete-cluster

Das folgende Codebeispiel zeigt die Verwendung `delete-cluster`.

AWS CLI

Das SnapshotThis Beispiel „Cluster ohne endgültigen Cluster löschen“ löscht einen Cluster und erzwingt das Löschen von Daten, sodass kein endgültiger Cluster-Snapshot erstellt wird. Befehl:

```
aws redshift delete-cluster --cluster-identifizier mycluster --skip-final-cluster-snapshot
```

Das SnapshotThis Beispiel „Cluster löschen, einen finalen Cluster zulassen“ löscht einen Cluster, gibt aber einen endgültigen Cluster-Snapshot an. Befehl:

```
aws redshift delete-cluster --cluster-identifizier mycluster --final-cluster-snapshot-identifizier myfinalsnapshot
```

- API-Einheiten finden Sie in der Befehlsreferenz [DeleteCluster](#). AWS CLI

delete-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `delete-event-subscription`.

AWS CLI

Um ein Event-Abonnement zu löschen

Im folgenden `delete-event-subscription` Beispiel wird das angegebene Abonnement für Ereignisbenachrichtigungen gelöscht.

```
aws redshift delete-event-subscription \
  --subscription-name mysubscription
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Abonnieren von Amazon Redshift-Ereignisbenachrichtigungen](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [DeleteEventSubscription](#) in AWS CLI der Befehlsreferenz.

delete-hsm-client-certificate

Das folgende Codebeispiel zeigt die Verwendung `delete-hsm-client-certificate`.

AWS CLI

Um das HSM Client-Zertifikat zu löschen

Im folgenden `delete-hsm-client-certificate` Beispiel wird ein HSM Client-Zertifikat gelöscht.

```
aws redshift delete-hsm-client-certificate \  
  --hsm-client-certificate-identifier myhsmclientcert
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Amazon Redshift API Permissions Reference](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DeleteHsmClientCertificate AWS CLI Befehlsreferenz](#).

delete-hsm-configuration

Das folgende Codebeispiel zeigt die Verwendung `delete-hsm-configuration`.

AWS CLI

Um eine HSM Konfiguration zu löschen

Im folgenden `delete-hsm-configuration` Beispiel wird die angegebene HSM Konfiguration aus dem aktuellen AWS Konto gelöscht.

```
aws redshift delete-hsm-configuration \  
  --hsm-configuration-identifier myhsmconnection
```

Dieser Befehl erzeugt keine Ausgabe.

- API-Einheiten finden Sie [DeleteHsmConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-scheduled-action

Das folgende Codebeispiel zeigt die Verwendung `delete-scheduled-action`.

AWS CLI

Um eine geplante Aktion zu löschen

Im folgenden `delete-scheduled-action` Beispiel wird die angegebene geplante Aktion gelöscht.

```
aws redshift delete-scheduled-action \  
  --scheduled-action-name myscheduledaction
```

Dieser Befehl erzeugt keine Ausgabe.

- API-Einheiten finden Sie unter [DeleteScheduledAction AWS CLI Befehlsreferenz](#).

`delete-snapshot-copy-grant`

Das folgende Codebeispiel zeigt die Verwendung `delete-snapshot-copy-grant`.

AWS CLI

Um Snapshot Copy Grant zu löschen

Im folgenden `delete-snapshot-copy-grant` Beispiel wird der angegebene Snapshot-Kopierzuschuss gelöscht.

```
aws redshift delete-snapshot-copy-grant \  
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Amazon Redshift Database Encryption](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DeleteSnapshotCopyGrant AWS CLI Befehlsreferenz](#).

`delete-snapshot-schedule`

Das folgende Codebeispiel zeigt die Verwendung `delete-snapshot-schedule`.

AWS CLI

Um den Snapshot-Zeitplan zu löschen

Im folgenden `delete-snapshot-schedule` Beispiel wird der angegebene Snapshot-Zeitplan gelöscht. Sie müssen die Cluster-Zuordnung aufheben, bevor Sie den Zeitplan löschen können.

```
aws redshift delete-snapshot-schedule \  
  --schedule-identifizier mysnapshotschedule
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Automated Snapshot Schedules](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DeleteSnapshotSchedule AWS CLI](#) Befehlsreferenz.

delete-tags

Das folgende Codebeispiel zeigt die Verwendung `delete-tags`.

AWS CLI

Um Tags aus einem Cluster zu löschen

Im folgenden `delete-tags` Beispiel werden die Tags mit den angegebenen Schlüsselnamen aus dem angegebenen Cluster gelöscht.

```
aws redshift delete-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tag-keys "clustertagkey" "clustertagvalue"
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Tagging Resources in Amazon Redshift](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [DeleteTags](#) in AWS CLI der Befehlsreferenz.

describe-account-attributes

Das folgende Codebeispiel zeigt die Verwendung `describe-account-attributes`.

AWS CLI

Um die Attribute eines AWS Kontos zu beschreiben

Im folgenden `describe-account-attributes` Beispiel werden die Attribute angezeigt, die dem anrufenden AWS Konto zugeordnet sind.

```
aws redshift describe-account-attributes
```

Ausgabe:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "max-defer-maintenance-duration",
      "AttributeValues": [
        {
          "AttributeValue": "45"
        }
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeAccountAttributes](#) unter AWS CLI Befehlsreferenz.

`describe-cluster-db-revisions`

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-db-revisions`.

AWS CLI

Um DB-Revisionen für einen Cluster zu beschreiben

Im folgenden `describe-cluster-db-revisions` Beispiel werden die Details eines Arrays von `ClusterDbRevision` Objekten für den angegebenen Cluster angezeigt.

```
aws redshift describe-cluster-db-revisions \
  --cluster-identifier mycluster
```

Ausgabe:

```
{
  "ClusterDbRevisions": [
    {
```

```

        "ClusterIdentifizier": "mycluster",
        "CurrentDatabaseRevision": "11420",
        "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",
        "RevisionTargets": []
    }
]
}

```

- API-Einheiten finden Sie [DescribeClusterDbRevisions](#) unter AWS CLI Befehlsreferenz.

describe-cluster-parameter-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-parameter-groups`.

AWS CLI

Das GroupsThis Beispiel „Beschreibung aller Clusterparameter abrufen“ gibt eine Beschreibung aller Cluster-Parametergruppen für das Konto mit Spaltenüberschriften zurück. Standardmäßig erfolgt die Ausgabe im JSON Format. Command:

```
aws redshift describe-cluster-parameter-groups
```

Ergebnis:

```

{
  "ParameterGroups": [
    {
      "ParameterGroupFamily": "redshift-1.0",
      "Description": "My first cluster parameter group",
      "ParameterGroupName": "myclusterparametergroup"
    } ],
  "ResponseMetadata": {
    "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
  }
}

```

Mit der Option können Sie dieselben Informationen auch im Textformat abrufen. Befehl `--output text`:

```
--output textOption.Befehl:
```

```
Option.Befehl:
```

```
aws redshift describe-cluster-parameter-groups --output text
```

Ergebnis:

```
redshift-1.0      My first cluster parameter group      myclusterparametergroup
RESPONSEMETADATA  9e665a36-64cc-11e2-8f7d-3b939af52818
```

- API-Einheiten finden Sie [DescribeClusterParameterGroups](#) in der AWS CLI Befehlsreferenz.

describe-cluster-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-parameters`.

AWS CLI

Das GroupThis Beispiel „Parameter für einen angegebenen Clusterparameter abrufen“ ruft die Parameter für die benannte Parametergruppe ab. Die Ausgabe erfolgt standardmäßig im JSON-Format. **COMMAND:**

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup
```

Ergebnis:

```
{
  "Parameters": [
    {
      "Description": "Sets the display format for date and time values.",
      "DataType": "string",
      "IsModifiable": true,
      "Source": "engine-default",
      "ParameterValue": "ISO, MDY",
      "ParameterName": "datestyle"
    },
    {
      "Description": "Sets the number of digits displayed for floating-point
values",
      "DataType": "integer",
      "IsModifiable": true,
      "AllowedValues": "-15-2",
      "Source": "engine-default",
```

```

        "ParameterValue": "0",
        "ParameterName": "extra_float_digits"
    },
    (...remaining output omitted...)
]
}

```

Mit der Option können Sie dieselben Informationen auch im Textformat abrufen. Befehl `--output text`:

`--output text`Option.Befehl:

Option.Befehl:

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup --output text
```

Ergebnis:

```

RESPONSEMETADATA    cdac40aa-64cc-11e2-9e70-918437dd236d
Sets the display format for date and time values.    string True    engine-default
ISO, MDY    datestyle
Sets the number of digits displayed for floating-point values    integer True
-15-2    engine-default 0    extra_float_digits
This parameter applies a user-defined label to a group of queries that are run
during the same session..    string True    engine-default default query_group
require ssl for all databaseconnections    boolean True    true,false    engine-
default false    require_ssl
Sets the schema search order for names that are not schema-qualified.    string
True    engine-default $user, public    search_path
Aborts any statement that takes over the specified number of milliseconds.    integer
True    engine-default 0    statement_timeout
wlm json configuration    string True    engine-default
\ [{"query_concurrency":5}]    wlm_json_configuration

```

- API-Einheiten finden Sie [DescribeClusterParameters](#) in der AWS CLI Befehlsreferenz.

describe-cluster-security-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-security-groups`.

AWS CLI

Das GroupsThis Beispiel Get a Description of All Cluster Security gibt eine Beschreibung aller Clustersicherheitsgruppen für das Konto zurück. Die Ausgabe erfolgt standardmäßig im JSON Format.COMMAND:

```
aws redshift describe-cluster-security-groups
```

Ergebnis:

```
{
  "ClusterSecurityGroups": [
    {
      "OwnerId": "100447751468",
      "Description": "default",
      "ClusterSecurityGroupName": "default",
      "EC2SecurityGroups": \[],
      "IPRanges": [
        {
          "Status": "authorized",
          "CIDRIP": "0.0.0.0/0"
        }
      ]
    },
    {
      "OwnerId": "100447751468",
      "Description": "This is my cluster security group",
      "ClusterSecurityGroupName": "mysecuritygroup",
      "EC2SecurityGroups": \[],
      "IPRanges": \[]
    },
    (...remaining output omitted...)
  ]
}
```

- APIEinzelheiten finden Sie [DescribeClusterSecurityGroups](#) in der AWS CLI Befehlsreferenz.

describe-cluster-snapshots

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-snapshots`.

AWS CLI

Das `aws redshift describe-cluster-snapshots` gibt eine Beschreibung aller Cluster-Snapshots für das Konto zurück. Standardmäßig erfolgt die Ausgabe im JSON Format.

```
aws redshift describe-cluster-snapshots
```

Ergebnis:

```
{
  "Snapshots": [
    {
      "Status": "available",
      "SnapshotCreateTime": "2013-07-17T22:02:22.852Z",
      "EstimatedSecondsToCompletion": -1,
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "Encrypted": false,
      "OwnerAccount": "111122223333",
      "BackupProgressInMegabytes": 20.0,
      "ElapsedTimeInSeconds": 0,
      "DBName": "dev",
      "CurrentBackupRateInMegabytesPerSecond": 0.0,
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "ActualIncrementalBackupSizeInMegabytes": 20.0,
      "SnapshotType": "automated",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "Port": 5439,
      "TotalBackupSizeInMegabytes": 20.0,
      "NumberOfNodes": "2",
      "SnapshotIdentifier": "cm:mycluster-2013-01-22-22-04-18"
    },
    {
      "EstimatedSecondsToCompletion": 0,
      "OwnerAccount": "111122223333",
      "CurrentBackupRateInMegabytesPerSecond": 0.1534,
      "ActualIncrementalBackupSizeInMegabytes": 11.0,
      "NumberOfNodes": "2",
      "Status": "available",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",

```

```

    "AccountsWithRestoreAccess": [
      {
        "AccountID": "444455556666"
      } ],
    "TotalBackupSizeInMegabytes": 20.0,
    "DBName": "dev",
    "BackupProgressInMegabytes": 11.0,
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "ElapsedTimeInSeconds": 0,
    "ClusterIdentifier": "mycluster",
    "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
    "AvailabilityZone": "us-east-1a",
    "NodeType": "dw.hs1.xlarge",
    "Encrypted": false,
    "SnapshotType": "manual",
    "Port": 5439,
    "SnapshotIdentifier": "my-snapshot-id"
  } ]
}
(...remaining output omitted...)

```

- API-Einheiten finden Sie [DescribeClusterSnapshots](#) in der AWS CLI Befehlsreferenz.

describe-cluster-subnet-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-subnet-groups`.

AWS CLI

Das GroupsThis Beispiel Get a Description of All Cluster Subnet gibt eine Beschreibung aller Cluster-Subnetzgruppen zurück. Standardmäßig erfolgt die Ausgabe im Format.COMMAND: JSON

```
aws redshift describe-cluster-subnet-groups
```

Ergebnis:

```

{
  "ClusterSubnetGroups": [
    {
      "Subnets": [
        {

```

```

        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-763fdd1c",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
        }
    },
    "VpcId": "vpc-7e3fdd14",
    "SubnetGroupStatus": "Complete",
    "Description": "My subnet group",
    "ClusterSubnetGroupName": "mysubnetgroup"
}
],
"ResponseMetadata": {
    "RequestId": "37fa8c89-6990-11e2-8f75-ab4018764c77"
}
}

```

- API-Einheiten finden Sie [DescribeClusterSubnetGroups](#) in der AWS CLI Befehlsreferenz.

describe-cluster-tracks

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-tracks`.

AWS CLI

Um Cluster-Tracks zu beschreiben

Im folgenden `describe-cluster-tracks` Beispiel werden Details zu den verfügbaren Wartungspfaden angezeigt.

```
aws redshift describe-cluster-tracks \
  --maintenance-track-name current
```

Ausgabe:

```
{
  "MaintenanceTracks": [
    {
      "MaintenanceTrackName": "current",
      "DatabaseVersion": "1.0.11420",
      "UpdateTargets": [

```



```

    {
      "MaintenanceTrackName": "preview_features",
      "DatabaseVersion": "1.0.11746",
      "SupportedOperations": [
        {
          "OperationName": "restore-from-cluster-snapshot"
        }
      ]
    },
    {
      "MaintenanceTrackName": "trailing",
      "DatabaseVersion": "1.0.11116",
      "SupportedOperations": [
        {
          "OperationName": "restore-from-cluster-snapshot"
        },
        {
          "OperationName": "modify-cluster"
        }
      ]
    }
  ]
}

```

Weitere Informationen finden Sie unter [Choosing Cluster Maintenance Tracks](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeClusterTracks AWS CLI Befehlsreferenz](#).

describe-cluster-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-cluster-versions`.

AWS CLI

Das VersionsThis Beispiel Get a Description of All Cluster gibt eine Beschreibung aller Clusterversionen zurück. Standardmäßig erfolgt die Ausgabe im JSON Format. Command:

```
aws redshift describe-cluster-versions
```

Ergebnis:

```
{
  "ClusterVersions": [
    {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
    } ],
  "ResponseMetadata": {
    "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
  }
}
```

- API-Einheiten finden Sie [DescribeClusterVersions](#) in der AWS CLI Befehlsreferenz.

describe-clusters

Das folgende Codebeispiel zeigt die Verwendung `describe-clusters`.

AWS CLI

Das `ClustersThis` Beispiel „Get a Description of All“ gibt eine Beschreibung aller Cluster für das Konto zurück. Standardmäßig erfolgt die Ausgabe im JSON-Format. Command:

```
aws redshift describe-clusters
```

Ergebnis:

```
{
  "Clusters": [
    {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
        "Port": 5439,
        "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
        {
          "ParameterApplyStatus": "in-sync",

```

```

        "ParameterGroupName": "default.redshift-1.0"
    } ],
    "ClusterSecurityGroups": [
        {
            "Status": "active",
            "ClusterSecurityGroupName": "default"
        } ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": \[],
    "AvailabilityZone": "us-east-1a",
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "available",
    "ClusterIdentifier": "mycluster",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {}
} ],
"ResponseMetadata": {
    "RequestId": "65b71cac-64df-11e2-8f5b-e90bd6c77476"
}
}

```

Mit der Option können Sie dieselben Informationen auch im Textformat abrufen. Befehl `--output text`:

`--output text`Option.Befehl:

Option.Befehl:

```
aws redshift describe-clusters --output text
```

Ergebnis:

```

dw.hs1.xlarge      1.0      true      adminuser      True      us-east-1a
2013-01-22T21:59:29.559Z      sat:03:30-sat:04:00      1      available
mycluster         dev      2
ENDPOINT          5439    mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com
in-sync           default.redshift-1.0
active            default
PENDINGMODIFIEDVALUES

```

```
RESPONSEMETADATA    934281a8-64df-11e2-b07c-f7fbdd006c67
```

- API-Einheiten finden Sie [DescribeClusters](#) in der AWS CLI Befehlsreferenz.

describe-default-cluster-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-default-cluster-parameters`.

AWS CLI

Das ParametersThis Beispiel „Beschreibung des Standard-Clusters abrufen“ gibt eine Beschreibung der Standard-Clusterparameter für die `redshift-1.0` Familie zurück. Standardmäßig erfolgt die Ausgabe im JSON Format. Command:

```
aws redshift describe-default-cluster-parameters --parameter-group-family
redshift-1.0
```

Ergebnis:

```
{
  "DefaultClusterParameters": {
    "ParameterGroupFamily": "redshift-1.0",
    "Parameters": [
      {
        "Description": "Sets the display format for date and time values.",
        "DataType": "string",
        "IsModifiable": true,
        "Source": "engine-default",
        "ParameterValue": "ISO, MDY",
        "ParameterName": "datestyle"
      },
      {
        "Description": "Sets the number of digits displayed for floating-point
values",
        "DataType": "integer",
        "IsModifiable": true,
        "AllowedValues": "-15-2",
        "Source": "engine-default",
        "ParameterValue": "0",
        "ParameterName": "extra_float_digits"
      },
      (...remaining output omitted...)
    ]
  }
}
```

```
    ]
  }
}
```

Verwenden Sie den Befehl, um eine Liste der gültigen Parametergruppenfamilien anzuzeigen.
describe-cluster-parameter-groups

describe-cluster-parameter-groupsBefehl.

Befehl.

- API-Einheiten finden Sie [DescribeDefaultClusterParameters](#) in der AWS CLI Befehlsreferenz.

describe-event-categories

Das folgende Codebeispiel zeigt die Verwendung **describe-event-categories**.

AWS CLI

Um Ereigniskategorien für einen Cluster zu beschreiben

Im folgenden **describe-event-categories** Beispiel werden Details zu den Ereigniskategorien für einen Cluster angezeigt.

```
aws redshift describe-event-categories \
  --source-type cluster
```

Ausgabe:

```
{
  "EventCategoriesMapList": [
    {
      "SourceType": "cluster",
      "Events": [
        {
          "EventId": "REDSHIFT-EVENT-2000",
          "EventCategories": [
            "management"
          ],
          "EventDescription": "Cluster <cluster name> created at <time in
UTC>.",
          "Severity": "INFO"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "EventId": "REDSHIFT-EVENT-2001",
      "EventCategories": [
        "management"
      ],
      "EventDescription": "Cluster <cluster name> deleted at <time in
UTC>.",
      "Severity": "INFO"
    },
    {
      "EventId": "REDSHIFT-EVENT-3625",
      "EventCategories": [
        "monitoring"
      ],
      "EventDescription": "The cluster <cluster name> can't be resumed
with its previous elastic network interface <ENI id>. We will allocate a new
elastic network interface and associate it with the cluster node.",
      "Severity": "INFO"
    }
  ]
}
]
}
}

```

- API-Einheiten finden Sie [DescribeEventCategories](#) unter AWS CLI Befehlsreferenz.

describe-event-subscriptions

Das folgende Codebeispiel zeigt die Verwendung `describe-event-subscriptions`.

AWS CLI

Um Veranstaltungsabonnements zu beschreiben

Im folgenden `describe-event-subscriptions` Beispiel werden Abonnements für Ereignisbenachrichtigungen für das angegebene Abonnement angezeigt.

```
aws redshift describe-event-subscriptions \
  --subscription-name mysubscription
```

Ausgabe:

```
{
  "EventSubscriptionsList": [
    {
      "CustomerAwsId": "123456789012",
      "CustSubscriptionId": "mysubscription",
      "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
      "Status": "active",
      "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",
      "SourceIdsList": [],
      "EventCategoriesList": [
        "management"
      ],
      "Severity": "ERROR",
      "Enabled": true,
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Abonnieren von Amazon Redshift-Ereignisbenachrichtigungen](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [DescribeEventSubscriptions](#) in AWS CLI der Befehlsreferenz.

describe-events

Das folgende Codebeispiel zeigt die Verwendung `describe-events`.

AWS CLI

Alle Ereignisse beschreiben. In diesem Beispiel werden alle Ereignisse zurückgegeben. Standardmäßig hat die Ausgabe das Format. Befehl: JSON

```
aws redshift describe-events
```

Ergebnis:

```
{
  "Events": [
    {
      "Date": "2013-01-22T19:17:03.640Z",
```

```

    "SourceIdentifier": "myclusterparametergroup",
    "Message": "Cluster parameter group myclusterparametergroup has been
created.",
    "SourceType": "cluster-parameter-group"
  } ],
  "ResponseMetadata": {
    "RequestId": "9f056111-64c9-11e2-9390-ff04f2c1e638"
  }
}

```

Mit der Option können Sie dieselben Informationen auch im Textformat abrufen. Befehl `--output text`:

`--output text`Option.Befehl:

Option.Befehl:

```
aws redshift describe-events --output text
```

Ergebnis:

```

2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
myclusterparametergroup has been created.    cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17

```

- API-Einheiten finden Sie [DescribeEvents](#) in der AWS CLI Befehlsreferenz.

describe-hsm-client-certificates

Das folgende Codebeispiel zeigt die Verwendung `describe-hsm-client-certificates`.

AWS CLI

Um HSM Client-Zertifikate zu beschreiben

Im folgenden `describe-hsm-client-certificates` Beispiel werden Details für das angegebene HSM Client-Zertifikat angezeigt.

```
aws redshift describe-hsm-client-certificates \
  --hsm-client-certificate-identifier myhsmClientcert
```


Ausgabe:

```
{
  "HsmClientCertificates": [
    {
      "HsmClientCertificateIdentifier": "myhsmclientcert",
      "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\n
EXAMPLECAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC0lBTSBDb25zEXAMPLEwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN
EXAMPLE0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sEXAMPLEdBGkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEEXAMPLEMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY67EXAMPLEE
EXAMPLEZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9EXAMPLE6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEEXAMPLEBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
      "Tags": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Amazon Redshift API Permissions Reference](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeHsmClientCertificates AWS CLI Befehlsreferenz](#).

describe-hsm-configurations

Das folgende Codebeispiel zeigt die Verwendung `describe-hsm-configurations`.

AWS CLI

Um HSM Konfigurationen zu beschreiben

Im folgenden `describe-hsm-configurations` Beispiel werden Details zu den verfügbaren HSM Konfigurationen für das anrufende AWS Konto angezeigt.

```
aws redshift describe-hsm-configurations /
```

```
--hsm-configuration-identifizier myhsmconnection
```

Ausgabe:

```
{
  "HsmConfigurations": [
    {
      "HsmConfigurationIdentifizier": "myhsmconnection",
      "Description": "My HSM connection",
      "HsmIpAddress": "192.0.2.09",
      "HsmPartitionName": "myhsmpartition",
      "Tags": []
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeHsmConfigurations](#) in der AWS CLI Befehlsreferenz.

describe-logging-status

Das folgende Codebeispiel zeigt die Verwendung `describe-logging-status`.

AWS CLI

Um den Protokollierungsstatus für einen Cluster zu beschreiben

Im folgenden `describe-logging-status` Beispiel wird angezeigt, ob Informationen, wie Abfragen und Verbindungsversuche, für einen Cluster protokolliert werden.

```
aws redshift describe-logging-status \
  --cluster-identifizier mycluster
```

Ausgabe:

```
{
  "LoggingEnabled": false
}
```

Weitere Informationen finden Sie unter [Database Audit Logging](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeLoggingStatus AWS CLI-Befehlsreferenz](#).

describe-node-configuration-options

Das folgende Codebeispiel zeigt die Verwendung `describe-node-configuration-options`.

AWS CLI

Um die Konfigurationsoptionen für Knoten zu beschreiben

Im folgenden `describe-node-configuration-options` Beispiel werden die Eigenschaften möglicher Knotenkonfigurationen wie Knotentyp, Anzahl der Knoten und Festplattennutzung für den angegebenen Cluster-Snapshot angezeigt.

```
aws redshift describe-node-configuration-options \
  --action-type restore-cluster \
  --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

Ausgabe:

```
{
  "NodeConfigurationOptionList": [
    {
      "NodeType": "dc2.large",
      "NumberOfNodes": 2,
      "EstimatedDiskUtilizationPercent": 19.61
    },
    {
      "NodeType": "dc2.large",
      "NumberOfNodes": 4,
      "EstimatedDiskUtilizationPercent": 9.96
    },
    {
      "NodeType": "ds2.xlarge",
      "NumberOfNodes": 2,
      "EstimatedDiskUtilizationPercent": 1.53
    },
    {
      "NodeType": "ds2.xlarge",
      "NumberOfNodes": 4,
      "EstimatedDiskUtilizationPercent": 0.78
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Purchasing Amazon Redshift Reserved Nodes](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeNodeConfigurationOptions AWS CLI Befehlsreferenz](#).

describe-orderable-cluster-options

Das folgende Codebeispiel zeigt die Verwendung `describe-orderable-cluster-options`.

AWS CLI

Das OptionsThis Beispiel Describing All Orderable Cluster gibt Beschreibungen aller bestellbaren Cluster-Optionen zurück. Standardmäßig erfolgt die Ausgabe im Format.COMMAND: JSON

```
aws redshift describe-orderable-cluster-options
```

Ergebnis:

```
{
  "OrderableClusterOptions": [
    {
      "NodeType": "dw.hs1.8xlarge",
      "AvailabilityZones": [
        { "Name": "us-east-1a" },
        { "Name": "us-east-1b" },
        { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "multi-node"
    },
    {
      "NodeType": "dw.hs1.xlarge",
      "AvailabilityZones": [
        { "Name": "us-east-1a" },
        { "Name": "us-east-1b" },
        { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "multi-node"
    },
    {
      "NodeType": "dw.hs1.xlarge",
```

```

    "AvailabilityZones": [
      { "Name": "us-east-1a" },
      { "Name": "us-east-1b" },
      { "Name": "us-east-1c" } ],
    "ClusterVersion": "1.0",
    "ClusterType": "single-node"
  } ],
  "ResponseMetadata": {
    "RequestId": "f6000035-64cb-11e2-9135-ff82df53a51a"
  }
}

```

Mit der Option können Sie dieselben Informationen auch im Textformat abrufen. Befehl `--output text`:

`--output text`Option.Befehl:

Option.Befehl:

```
aws redshift describe-orderable-cluster-options --output text
```

Ergebnis:

```

dw.hs1.8xlarge      1.0      multi-node
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge      1.0      multi-node
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge      1.0      single-node
us-east-1a
us-east-1b
us-east-1c
RESPONSEMETADATA  e648696b-64cb-11e2-bec0-17624ad140dd

```

- API-Einheiten finden Sie [DescribeOrderableClusterOptions](#) in der AWS CLI Befehlsreferenz.

describe-reserved-node-offerings

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-node-offerings`.

AWS CLI

Das OfferingsThis Beispiel Describe Reserved Node zeigt alle Angebote für reservierte Knoten, die käuflich erworben werden können.Command:

```
aws redshift describe-reserved-node-offerings
```

Ergebnis:

```
{
  "ReservedNodeOfferings": [
    {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.xlarge",
      "UsagePrice": "",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": "",
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "Duration": 31536000,
      "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
    },
    {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.8xlarge",
      "UsagePrice": "",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": "",
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "Duration": 31536000,
      "ReservedNodeOfferingId": "e5a2ff3b-352d-4a9c-ad7d-373c4cab5dd2"
    },
    ...remaining output omitted...
  ],
  "ResponseMetadata": {
    "RequestId": "8b1a1a43-75ff-11e2-9666-e142fe91ddd1"
  }
}
```

Wenn Sie ein Angebot für reservierte Knoten erwerben möchten, können Sie `purchase-reserved-node-offering` mit einem gültigen Kennwort anrufen. `ReservedNodeOfferingId`

`purchase-reserved-node-offering` mit einem gültigen `ReservedNodeOfferingId`.

unter Verwendung eines gültigen `ReservedNodeOfferingId`.

`ReservedNodeOfferingId`.

- API-Einheiten finden Sie [DescribeReservedNodeOfferings](#) in der AWS CLI Befehlsreferenz.

describe-reserved-nodes

Das folgende Codebeispiel zeigt die Verwendung `describe-reserved-nodes`.

AWS CLI

Das NodesThis Beispiel Describe Reserved zeigt ein Angebot für reservierte Knoten, das gekauft wurde. Befehl:

```
aws redshift describe-reserved-nodes
```

Ergebnis:

```
{
  "ResponseMetadata": {
    "RequestId": "bc29ce2e-7600-11e2-9949-4b361e7420b7"
  },
  "ReservedNodes": [
    {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.xlarge",
      "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
      "UsagePrice": "",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": "",
          "RecurringChargeFrequency": "Hourly"
        }
      ]
    }
  ]
}
```

```
    } ],  
    "NodeCount": 1,  
    "State": "payment-pending",  
    "StartTime": "2013-02-13T17:08:39.051Z",  
    "Duration": 31536000,  
    "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"  
  }  
]  
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [DescribeReservedNodes](#). AWS CLI

describe-resize

Das folgende Codebeispiel zeigt die Verwendung `describe-resize`.

AWS CLI

Das `ResizeThis` Beschreibungsbeispiel beschreibt die letzte Größenänderung eines Clusters. Die Anfrage betraf 3 Knoten des Typs `dw.hs1.8xlarge`. Command:

```
aws redshift describe-resize --cluster-identifier mycluster
```

Ergebnis:

```
{  
  "Status": "NONE",  
  "TargetClusterType": "multi-node",  
  "TargetNodeType": "dw.hs1.8xlarge",  
  "ResponseMetadata": {  
    "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"  
  },  
  "TargetNumberOfNodes": "3"  
}
```

- API-Einheiten finden Sie [DescribeResize](#) in der AWS CLI Befehlsreferenz.

describe-scheduled-actions

Das folgende Codebeispiel zeigt die Verwendung `describe-scheduled-actions`.

AWS CLI

Um geplante Aktionen zu beschreiben

Im folgenden `describe-scheduled-actions` Beispiel werden Details zu allen derzeit geplanten Aktionen angezeigt.

```
aws redshift describe-scheduled-actions
```

Ausgabe:

```
{
  "ScheduledActions": [
    {
      "ScheduledActionName": "resizecluster",
      "TargetAction": {
        "ResizeCluster": {
          "ClusterIdentifier": "mycluster",
          "NumberOfNodes": 4,
          "Classic": false
        }
      },
      "Schedule": "at(2019-12-10T00:07:00)",
      "IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",
      "State": "ACTIVE",
      "NextInvocations": [
        "2019-12-10T00:07:00Z"
      ]
    }
  ]
}
```

- API-Einheiten finden Sie [DescribeScheduledActions](#) unter AWS CLI Befehlsreferenz.

`describe-snapshot-copy-grants`

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshot-copy-grants`.

AWS CLI

Um Snapshot Copy Grants zu beschreiben

Im folgenden `describe-snapshot-copy-grants` Beispiel werden Details für die angegebene Zuweisung von Cluster-Snapshot-Kopien angezeigt.

```
aws redshift describe-snapshot-copy-grants \  
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

Ausgabe:

```
{  
  "SnapshotCopyGrants": [  
    {  
      "SnapshotCopyGrantName": "mysnapshotcopygrantname",  
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/  
bPxRfih3yCo8nvbEXAMPLEKEY",  
      "Tags": []  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Amazon Redshift Database Encryption](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeSnapshotCopyGrants AWS CLI Befehlsreferenz](#).

`describe-snapshot-schedules`

Das folgende Codebeispiel zeigt die Verwendung `describe-snapshot-schedules`.

AWS CLI

Um Snapshot-Zeitpläne zu beschreiben

Im folgenden `describe-snapshot-schedules` Beispiel werden Details für den angegebenen Cluster-Snapshot-Zeitplan angezeigt.

```
aws redshift describe-snapshot-schedules \  
  --cluster-identifier mycluster \  
  --schedule-identifier mysnapshotschedule
```

Ausgabe:

```
{
  "SnapshotSchedules": [
    {
      "ScheduleDefinitions": [
        "rate(12 hours)"
      ],
      "ScheduleIdentifier": "mysnapshotschedule",
      "ScheduleDescription": "My schedule description",
      "Tags": [],
      "AssociatedClusterCount": 1,
      "AssociatedClusters": [
        {
          "ClusterIdentifier": "mycluster",
          "ScheduleAssociationState": "ACTIVE"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [Automated Snapshot Schedules](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeSnapshotSchedules AWS CLI](#) Befehlsreferenz.

describe-storage

Das folgende Codebeispiel zeigt die Verwendung `describe-storage`.

AWS CLI

Um Speicher zu beschreiben

Im folgenden `describe-storage` Beispiel werden Details zum Backup-Speicher und zu den vorläufigen Speichergrößen für das Konto angezeigt.

```
aws redshift describe-storage
```

Ausgabe:

```
{
```

```
"TotalBackupSizeInMegaBytes": 193149.0,  
"TotalProvisionedStorageInMegaBytes": 655360.0  
}
```

Weitere Informationen finden Sie unter [Managing Snapshot Storage](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [DescribeStorage](#) in der AWS CLI Befehlsreferenz.

describe-table-restore-status

Das folgende Codebeispiel zeigt die Verwendung `describe-table-restore-status`.

AWS CLI

Um den Status von Anfragen zur Tabellenwiederherstellung aus einem Cluster-Snapshot zu beschreiben

Im folgenden `describe-table-restore-status` Beispiel werden Details zu Anfragen zur Tabellenwiederherstellung angezeigt, die für den angegebenen Cluster gestellt wurden.

```
aws redshift describe-table-restore-status /  
--cluster-identifier mycluster
```

Ausgabe:

```
{  
  "TableRestoreStatusDetails": [  
    {  
      "TableRestoreRequestId": "z1116630-0e80-46f4-ba86-bd9670411ebd",  
      "Status": "IN_PROGRESS",  
      "RequestTime": "2019-12-27T18:22:12.257Z",  
      "ClusterIdentifier": "mycluster",  
      "SnapshotIdentifier": "mysnapshotid",  
      "SourceDatabaseName": "dev",  
      "SourceSchemaName": "public",  
      "SourceTableName": "mytable",  
      "TargetDatabaseName": "dev",  
      "TargetSchemaName": "public",  
      "NewTableName": "mytable-clone"  
    }  
  ]  
}
```

```
}
```

Weitere Informationen finden Sie unter [Wiederherstellen einer Tabelle aus einem Snapshot](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DescribeTableRestoreStatus AWS CLI Befehlsreferenz](#).

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Um Tags zu beschreiben

Im folgenden `describe-tags` Beispiel werden die Ressourcen angezeigt, die der angegebene Cluster den angegebenen Tag-Namen und -Werten zugeordnet hat.

```
aws redshift describe-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tag-keys clustertagkey \  
  --tag-values clustertagvalue
```

Ausgabe:

```
{  
  "TaggedResources": [  
    {  
      "Tag": {  
        "Key": "clustertagkey",  
        "Value": "clustertagvalue"  
      },  
      "ResourceName": "arn:aws:redshift:us-  
west-2:123456789012:cluster:mycluster",  
      "ResourceType": "cluster"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Tagging Resources in Amazon Redshift](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [DescribeTags](#) in AWS CLI der Befehlsreferenz.

disable-snapshot-copy

Das folgende Codebeispiel zeigt die Verwendung `disable-snapshot-copy`.

AWS CLI

Um die Snapshot-Kopie für einen Cluster zu deaktivieren

Im folgenden `disable-snapshot-copy` Beispiel wird die automatische Kopie eines Snapshots für den angegebenen Cluster deaktiviert.

```
aws redshift disable-snapshot-copy \  
  --cluster-identifier mycluster
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "available",  
    "ClusterAvailabilityStatus": "Available",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-i9b431cd",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {
```

```
        "ParameterGroupName": "default.redshift-1.0",
        "ParameterApplyStatus": "in-sync"
    }
],
"ClusterSubnetGroupName": "default",
"VpcId": "vpc-b1fel7t9",
"AvailabilityZone": "us-west-2f",
"PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
"PendingModifiedValues": {
    "NodeType": "dc2.large",
    "NumberOfNodes": 2,
    "ClusterType": "multi-node"
},
"ClusterVersion": "1.0",
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"Tags": [
    {
        "Key": "mytags",
        "Value": "tag1"
    }
],
"EnhancedVpcRouting": false,
"IamRoles": [
    {
        "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
        "ApplyStatus": "in-sync"
    }
],
"MaintenanceTrackName": "current",
"DeferredMaintenanceWindows": [],
"ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
"ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
"NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
}
}
```

Weitere Informationen finden Sie unter [Kopieren von Snapshots in eine andere AWS Region](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [DisableSnapshotCopy AWS CLI Befehlsreferenz](#).

enable-snapshot-copy

Das folgende Codebeispiel zeigt die Verwendung von `enable-snapshot-copy`.

AWS CLI

Um die Snapshot-Kopie für einen Cluster zu aktivieren

Das folgende `enable-snapshot-copy` Beispiel aktiviert das automatische Kopieren eines Snapshots für den angegebenen Cluster.

```
aws redshift enable-snapshot-copy \  
  --cluster-identifier mycluster \  
  --destination-region us-west-1
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "available",  
    "ClusterAvailabilityStatus": "Available",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-f4c731cd",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ]  
  }  
}
```



```
    }
  ],
  "ClusterSubnetGroupName": "default",
  "VpcId": "vpc-b1ael7t9",
  "AvailabilityZone": "us-west-2f",
  "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
  "PendingModifiedValues": {
    "NodeType": "dc2.large",
    "NumberOfNodes": 2,
    "ClusterType": "multi-node"
  },
  "ClusterVersion": "1.0",
  "AllowVersionUpgrade": true,
  "NumberOfNodes": 4,
  "PubliclyAccessible": false,
  "Encrypted": false,
  "ClusterSnapshotCopyStatus": {
    "DestinationRegion": "us-west-1",
    "RetentionPeriod": 7,
    "ManualSnapshotRetentionPeriod": -1
  },
  "Tags": [
    {
      "Key": "mytags",
      "Value": "tag1"
    }
  ],
  "EnhancedVpcRouting": false,
  "IamRoles": [
    {
      "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
      "ApplyStatus": "in-sync"
    }
  ],
  "MaintenanceTrackName": "current",
  "DeferredMaintenanceWindows": [],
  "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
  "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
  "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
}
}
```

Weitere Informationen finden Sie unter [Kopieren von Snapshots in eine andere AWS Region](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [EnableSnapshotCopy AWS CLI](#) Befehlsreferenz.

get-cluster-credentials

Das folgende Codebeispiel zeigt die Verwendung `get-cluster-credentials`.

AWS CLI

Um Cluster-Anmeldeinformationen für ein AWS Konto abzurufen

Im folgenden `get-cluster-credentials` Beispiel werden temporäre Anmeldeinformationen abgerufen, die den Zugriff auf eine Amazon Redshift Redshift-Datenbank ermöglichen.

```
aws redshift get-cluster-credentials \  
  --db-user adminuser --db-name dev \  
  --cluster-identifizier mycluster
```

Ausgabe:

```
{  
  "DbUser": "IAM:adminuser",  
  "DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx109d6UmpkBtvEqFly/  
EXAMPLE==",  
  "Expiration": "2019-12-10T17:25:05.770Z"  
}
```

Weitere Informationen finden Sie unter [Generieren von IAM Datenbankanmeldedaten mit Amazon Redshift CLI oder API](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [GetClusterCredentials AWS CLI](#) Befehlsreferenz.

get-reserved-node-exchange-offerings

Das folgende Codebeispiel zeigt die Verwendung `get-reserved-node-exchange-offerings`.

AWS CLI

Um Angebote für den Austausch reservierter Knoten zu erhalten

Im folgenden `get-reserved-node-exchange-offerings` Beispiel wird ein Array abgerufen `DC2ReservedNodeOfferings`, das dem angegebenen DC1 reservierten Knoten entspricht.

```
aws redshift get-reserved-node-exchange-offerings \  
--reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

Ausgabe:

```
{  
  "ReservedNodeOfferings": [  
    {  
      "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",  
      "NodeType": "dc2.large",  
      "Duration": 31536000,  
      "FixedPrice": 0.0,  
      "UsagePrice": 0.0,  
      "CurrencyCode": "USD",  
      "OfferingType": "All Upfront",  
      "RecurringCharges": [  
        {  
          "RecurringChargeAmount": 0.0,  
          "RecurringChargeFrequency": "Hourly"  
        }  
      ],  
      "ReservedNodeOfferingType": "Regular"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Upgrading Reserved Nodes With AWS CLI](#) the im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [GetReservedNodeExchangeOfferings AWS CLI](#) Befehlsreferenz.

modify-cluster-iam-roles

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-iam-roles`.

AWS CLI

Um die IAM Rolle für einen Cluster zu ändern

Im folgenden `modify-cluster-iam-roles` Beispiel wird die angegebene AWS IAM Rolle aus dem angegebenen Cluster entfernt.

```
aws redshift modify-cluster-iam-roles \  
  --cluster-identifier mycluster \  
  --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "available",  
    "ClusterAvailabilityStatus": "Available",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-f9b731sd",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "ClusterSubnetGroupName": "default",  
    "VpcId": "vpc-b2fal7t9",  
    "AvailabilityZone": "us-west-2f",  
    "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",  
    "PendingModifiedValues": {  
      "NodeType": "dc2.large",
```

```
        "NumberOfNodes": 2,
        "ClusterType": "multi-node"
    },
    "ClusterVersion": "1.0",
    "AllowVersionUpgrade": true,
    "NumberOfNodes": 4,
    "PubliclyAccessible": false,
    "Encrypted": false,
    "ClusterSnapshotCopyStatus": {
        "DestinationRegion": "us-west-1",
        "RetentionPeriod": 7,
        "ManualSnapshotRetentionPeriod": -1
    },
    "Tags": [
        {
            "Key": "mytags",
            "Value": "tag1"
        }
    ],
    "EnhancedVpcRouting": false,
    "IamRoles": [],
    "MaintenanceTrackName": "current",
    "DeferredMaintenanceWindows": [],
    "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
    "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
    "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
}
}
```

Weitere Informationen finden Sie unter [Using Identity-Based Policies \(IAM Policies\) for Amazon Redshift](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie in der Befehlsreferenz [ModifyClusterIamRoles](#).AWS CLI

modify-cluster-maintenance

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-maintenance`.

AWS CLI

Um die Cluster-Wartung zu ändern

Im folgenden `modify-cluster-maintenance` Beispiel wird die Wartung des angegebenen Clusters um 30 Tage verschoben.

```
aws redshift modify-cluster-maintenance \  
  --cluster-identifizier mycluster \  
  --defer-maintenance \  
  --defer-maintenance-duration 30
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterIdentifizier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "available",  
    "ClusterAvailabilityStatus": "Available",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-a1a123ab",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "ClusterSubnetGroupName": "default",  
    "VpcId": "vpc-b1ael7t9",  
    "AvailabilityZone": "us-west-2f",  
    "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",  
    "PendingModifiedValues": {
```

```
    "NodeType": "dc2.large",
    "NumberOfNodes": 2,
    "ClusterType": "multi-node"
  },
  "ClusterVersion": "1.0",
  "AllowVersionUpgrade": true,
  "NumberOfNodes": 4,
  "PubliclyAccessible": false,
  "Encrypted": false,
  "ClusterSnapshotCopyStatus": {
    "DestinationRegion": "us-west-1",
    "RetentionPeriod": 7,
    "ManualSnapshotRetentionPeriod": -1
  },
  "Tags": [
    {
      "Key": "mytags",
      "Value": "tag1"
    }
  ],
  "EnhancedVpcRouting": false,
  "IamRoles": [],
  "MaintenanceTrackName": "current",
  "DeferredMaintenanceWindows": [
    {
      "DeferMaintenanceIdentifier": "dfm-mUdVIffFcT1B4SGhw6fyF",
      "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
      "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
    }
  ],
  "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
  "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
  "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
}
}
```

Weitere Informationen finden Sie unter [Cluster-Wartung](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [ModifyClusterMaintenance](#) in der AWS CLI Befehlsreferenz.

modify-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-parameter-group`.

AWS CLI

Ändern Sie einen Parameter in einer Parametergruppe

Im folgenden `modify-cluster-parameter-group` Beispiel wird der Parameter `wlm_json_configuration` für die Arbeitslastverwaltung geändert. Es akzeptiert die Parameter aus einer Datei, die den unten gezeigten Inhalt enthält. JSON

```
aws redshift modify-cluster-parameter-group \  
  --parameter-group-name myclusterparametergroup \  
  --parameters file://modify_pg.json
```

Inhalt von `modify_pg.json`:

```
[  
  {  
    "ParameterName": "wlm_json_configuration",  
    "ParameterValue": "[{\\"user_group\\":\\"example_user_group1\\",\\"query_group\\":  
  \\"example_query_group1\\", \\"query_concurrency\\":7},{\\"query_concurrency\\":5}]"  
  }  
]
```

Ausgabe:

```
{  
  "ParameterGroupStatus": "Your parameter group has been updated but changes won't  
  get applied until you reboot the associated Clusters.",  
  "ParameterGroupName": "myclusterparametergroup",  
  "ResponseMetadata": {  
    "RequestId": "09974cc0-64cd-11e2-bea9-49e0ce183f07"  
  }  
}
```

- API-Einheiten finden Sie [ModifyClusterParameterGroup](#) in der AWS CLI Befehlsreferenz.

modify-cluster-snapshot-schedule

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-snapshot-schedule`.

AWS CLI

Um den Cluster-Snapshot-Zeitplan zu ändern

Im folgenden `modify-cluster-snapshot-schedule` Beispiel wird der angegebene Snapshot-Zeitplan aus dem angegebenen Cluster entfernt.

```
aws redshift modify-cluster-snapshot-schedule \  
  --cluster-identifizier mycluster \  
  --schedule-identifizier mysnapshotschedule \  
  --disassociate-schedule
```

Dieser Befehl erzeugt keine Ausgabe.

Weitere Informationen finden Sie unter [Automated Snapshot Schedules](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [ModifyClusterSnapshotSchedule AWS CLI Befehlsreferenz](#).

modify-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-snapshot`.

AWS CLI

Um den Cluster-Snapshot zu ändern

Im folgenden `modify-cluster-snapshot` Beispiel wird die Einstellung für den manuellen Aufbewahrungszeitraum für den angegebenen Cluster-Snapshot auf den Wert von 10 Tagen festgelegt.

```
aws redshift modify-cluster-snapshot \  
  --snapshot-identifizier mycluster-2019-11-06-16-32 \  
  --manual-snapshot-retention-period 10
```

Ausgabe:

```
{  
  "Snapshot": {  
    "SnapshotIdentifizier": "mycluster-2019-11-06-16-32",  
    "ClusterIdentifizier": "mycluster",
```

```
"SnapshotCreateTime": "2019-12-07T00:34:05.633Z",
"Status": "available",
"Port": 5439,
"AvailabilityZone": "us-west-2f",
"ClusterCreateTime": "2019-12-05T18:44:36.991Z",
"MasterUsername": "adminuser",
"ClusterVersion": "1.0",
"SnapshotType": "manual",
"NodeType": "dc2.large",
"NumberOfNodes": 2,
"DBName": "dev",
"VpcId": "vpc-b1cel7t9",
"Encrypted": false,
"EncryptedWithHSM": false,
"OwnerAccount": "123456789012",
"TotalBackupSizeInMegaBytes": 64384.0,
"ActualIncrementalBackupSizeInMegaBytes": 24.0,
"BackupProgressInMegaBytes": 24.0,
"CurrentBackupRateInMegaBytesPerSecond": 13.0011,
"EstimatedSecondsToCompletion": 0,
"ElapsedTimeInSeconds": 1,
"Tags": [
  {
    "Key": "mytagkey",
    "Value": "mytagvalue"
  }
],
"EnhancedVpcRouting": false,
"MaintenanceTrackName": "current",
"ManualSnapshotRetentionPeriod": 10,
"ManualSnapshotRemainingDays": 6,
"SnapshotRetentionStartTime": "2019-12-07T00:34:07.479Z"
}
}
```

Weitere Informationen finden Sie unter [Amazon Redshift Snapshots](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [ModifyClusterSnapshot](#).AWS CLI

modify-cluster-subnet-group

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster-subnet-group`.

AWS CLI

Das GroupThis Beispiel „Subnetze in einem Cluster-Subnetz ändern“ zeigt, wie die Liste der Subnetze in einer Cache-Subnetzgruppe geändert wird. Standardmäßig erfolgt die Ausgabe im Format.COMMAND: JSON

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
--subnet-ids subnet-763fdd1 subnet-ac830e9
```

Ergebnis:

```
{
  "ClusterSubnetGroup":
  {
    "Subnets": [
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-763fdd1c",
        "SubnetAvailabilityZone":
          { "Name": "us-east-1a" }
      },
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-ac830e9",
        "SubnetAvailabilityZone":
          { "Name": "us-east-1b" }
      }
    ],
    "VpcId": "vpc-7e3fdd14",
    "SubnetGroupStatus": "Complete",
    "Description": "My subnet group",
    "ClusterSubnetGroupName": "mysubnetgroup"
  },
  "ResponseMetadata": {
    "RequestId": "8da93e89-8372-f936-93a8-873918938197a"
  }
}
```

- APIEinzelheiten finden Sie [ModifyClusterSubnetGroup](#) in der AWS CLI Befehlsreferenz.

modify-cluster

Das folgende Codebeispiel zeigt die Verwendung `modify-cluster`.

AWS CLI

Die Zuordnung einer Sicherheitsgruppe zu einem Cluster. Das folgende Beispiel zeigt, wie eine Cluster-Sicherheitsgruppe mit dem angegebenen Cluster verknüpft wird. **Command:**

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups mysecuritygroup
```

Das Wartungsfenster ändern für Cluster. Das folgende Beispiel zeigt, wie das bevorzugte wöchentliche Wartungsfenster für einen Cluster so geändert werden kann, dass es mindestens vier Stunden dauert und sonntags um 23:15 Uhr beginnt und montags um 3:15 Uhr endet. **Befehl:**

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-window Sun:23:15-Mon:03:15
```

Das Master-Passwort ändern. Das folgende Beispiel zeigt, wie das Master-Passwort für einen Cluster geändert wird. **Befehl:**

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password A1b2c3d4
```

- API-Einheiten finden Sie [ModifyCluster](#) in AWS CLI der Befehlsreferenz.

modify-event-subscription

Das folgende Codebeispiel zeigt die Verwendung `modify-event-subscription`.

AWS CLI

Um das Event-Abonnement zu ändern

Im folgenden `modify-event-subscription` Beispiel wird das angegebene Abonnement für Ereignisbenachrichtigungen deaktiviert.

```
aws redshift modify-event-subscription \  
  --subscription-name mysubscription \  
  --no-enabled
```

Ausgabe:

```
{
```

```

"EventSubscription": {
  "CustomerAwsId": "123456789012",
  "CustSubscriptionId": "mysubscription",
  "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNSStopic",
  "Status": "active",
  "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",
  "SourceIdsList": [],
  "EventCategoriesList": [
    "management"
  ],
  "Severity": "ERROR",
  "Enabled": false,
  "Tags": []
}
}

```

Weitere Informationen finden Sie unter [Abonnieren von Amazon Redshift-Ereignisbenachrichtigungen](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [ModifyEventSubscription](#) in AWS CLI der Befehlsreferenz.

modify-scheduled-action

Das folgende Codebeispiel zeigt die Verwendung `modify-scheduled-action`.

AWS CLI

Um eine geplante Aktion zu ändern

Im folgenden `modify-scheduled-action` Beispiel wird der angegebenen vorhandenen geplanten Aktion eine Beschreibung hinzugefügt.

```

aws redshift modify-scheduled-action \
  --scheduled-action-name myscheduledaction \
  --scheduled-action-description "My scheduled action"

```

Ausgabe:

```

{
  "ScheduledActionName": "myscheduledaction",
  "TargetAction": {
    "ResizeCluster": {
      "ClusterIdentifier": "mycluster",

```

```

        "NumberOfNodes": 2,
        "Classic": false
    }
},
"Schedule": "at(2019-12-25T00:00:00)",
"IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",
"ScheduledActionDescription": "My scheduled action",
"State": "ACTIVE",
"NextInvocations": [
    "2019-12-25T00:00:00Z"
]
}

```

- API-Einheiten finden Sie [ModifyScheduledAction](#) unter AWS CLI Befehlsreferenz.

modify-snapshot-copy-retention-period

Das folgende Codebeispiel zeigt die Verwendung `modify-snapshot-copy-retention-period`.

AWS CLI

Um den Aufbewahrungszeitraum für Snapshot-Kopien zu ändern

Im folgenden `modify-snapshot-copy-retention-period` Beispiel wird die Anzahl der Tage geändert, für die Snapshots für den angegebenen Cluster in der AWS Zielregion aufbewahrt werden, nachdem sie aus der AWS Quellregion kopiert wurden.

```

aws redshift modify-snapshot-copy-retention-period \
  --cluster-identifier mycluster \
  --retention-period 15

```

Ausgabe:

```

{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "ClusterStatus": "available",
    "ClusterAvailabilityStatus": "Available",
    "MasterUsername": "adminuser",
    "DBName": "dev",
    "Endpoint": {

```

```
    "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
    "Port": 5439
  },
  "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
  "AutomatedSnapshotRetentionPeriod": 3,
  "ManualSnapshotRetentionPeriod": -1,
  "ClusterSecurityGroups": [],
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sh-a1a123ab",
      "Status": "active"
    }
  ],
  "ClusterParameterGroups": [
    {
      "ParameterGroupName": "default.redshift-1.0",
      "ParameterApplyStatus": "in-sync"
    }
  ],
  "ClusterSubnetGroupName": "default",
  "VpcId": "vpc-b1fet7t9",
  "AvailabilityZone": "us-west-2f",
  "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
  "PendingModifiedValues": {
    "NodeType": "dc2.large",
    "NumberOfNodes": 2,
    "ClusterType": "multi-node"
  },
  "ClusterVersion": "1.0",
  "AllowVersionUpgrade": true,
  "NumberOfNodes": 4,
  "PubliclyAccessible": false,
  "Encrypted": false,
  "ClusterSnapshotCopyStatus": {
    "DestinationRegion": "us-west-1",
    "RetentionPeriod": 15,
    "ManualSnapshotRetentionPeriod": -1
  },
  "Tags": [
    {
      "Key": "mytags",
      "Value": "tag1"
    }
  ],
],
```

```

    "EnhancedVpcRouting": false,
    "IamRoles": [],
    "MaintenanceTrackName": "current",
    "DeferredMaintenanceWindows": [
      {
        "DeferMaintenanceIdentifier": "dfm-mUdVSfDcT1F4SGhw6fyF",
        "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
        "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
      }
    ],
    "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
  }
}

```

Weitere Informationen finden Sie unter [Snapshot Schedule Format](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [ModifySnapshotCopyRetentionPeriod AWS CLI Befehlsreferenz](#).

modify-snapshot-schedule

Das folgende Codebeispiel zeigt die Verwendung `modify-snapshot-schedule`.

AWS CLI

Um den Snapshot-Zeitplan zu ändern

Im folgenden `modify-snapshot-schedule` Beispiel wird die Rate des angegebenen Snapshot-Zeitplans auf alle 10 Stunden geändert.

```

aws redshift modify-snapshot-schedule \
  --schedule-identifier mynapshotschedule \
  --schedule-definitions "rate(10 hours)"

```

Ausgabe:

```

{
  "ScheduleDefinitions": [
    "rate(10 hours)"
  ],
  "ScheduleIdentifier": "mynapshotschedule",
  "ScheduleDescription": "My schedule description",
}

```



```
"Tags": []
}
```

Weitere Informationen finden Sie unter [Snapshot Schedule Format](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [ModifySnapshotSchedule AWS CLI Befehlsreferenz](#).

purchase-reserved-node-offering

Das folgende Codebeispiel zeigt die Verwendung `purchase-reserved-node-offering`.

AWS CLI

Das NodeThis Beispiel Purchase a Reserved zeigt, wie Sie ein Angebot für reservierte Knoten erwerben. Das `reserved-node-offering-id` wird durch Aufrufen von `describe-reserved-node-offerings` .Command abgerufen:

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-
cf4c-4343-be8b-d832c45ab51c
```

Ergebnis:

```
{
  "ReservedNode": {
    "OfferingType": "Heavy Utilization",
    "FixedPrice": "",
    "NodeType": "dw.hs1.xlarge",
    "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
    "UsagePrice": "",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": "",
        "RecurringChargeFrequency": "Hourly"
      }
    ],
    "NodeCount": 1,
    "State": "payment-pending",
    "StartTime": "2013-02-13T17:08:39.051Z",
    "Duration": 31536000,
    "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
  },
}
```

```
"ResponseMetadata": {
  "RequestId": "01bda7bf-7600-11e2-b605-2568d7396e7f"
}
```

- API-Einheiten finden Sie [PurchaseReservedNodeOffering](#) in der AWS CLI Befehlsreferenz.

reboot-cluster

Das folgende Codebeispiel zeigt die Verwendung `reboot-cluster`.

AWS CLI

Ein Cluster-Neustart startet einen Cluster neu. Standardmäßig hat die Ausgabe das JSON-Format. Befehl:

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

Ergebnis:

```
{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "Endpoint": {
      "Port": 5439,
      "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
    },
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],
    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      }
    ],
  },
}
```

```

    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": [],
    "AvailabilityZone": "us-east-1a",
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "rebooting",
    "ClusterIdentifier": "mycluster",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {}
  },
  "ResponseMetadata": {
    "RequestId": "61c8b564-64e8-11e2-8f7d-3b939af52818"
  }
}

```

- API-Einheiten finden Sie [RebootCluster](#) in der AWS CLI Befehlsreferenz.

reset-cluster-parameter-group

Das folgende Codebeispiel zeigt die Verwendung `reset-cluster-parameter-group`.

AWS CLI

Das GroupThis Beispiel „Parameter in einem Parameter zurücksetzen“ zeigt, wie alle Parameter in einer Parametergruppe zurückgesetzt werden. Befehl:

```

aws redshift reset-cluster-parameter-group --parameter-group-name
myclusterparametergroup --reset-all-parameters

```

- API-Einheiten finden Sie [ResetClusterParameterGroup](#) in der AWS CLI Befehlsreferenz.

resize-cluster

Das folgende Codebeispiel zeigt die Verwendung `resize-cluster`.

AWS CLI

Um die Größe des Clusters zu ändern

Im folgenden `resize-cluster` Beispiel wird die Größe des angegebenen Clusters geändert.

```
aws redshift resize-cluster \  
  --cluster-identifier mycluster \  
  --cluster-type multi-node \  
  --node-type dc2.large \  
  --number-of-nodes 6 \  
  --classic
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "resizing",  
    "ClusterAvailabilityStatus": "Modifying",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-a1a123ab",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "ClusterSubnetGroupName": "default",  
    "VpcId": "vpc-a1abc1a1",  
    "AvailabilityZone": "us-west-2f",  
    "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",  
    "PendingModifiedValues": {  
      "NodeType": "dc2.large",
```

```
        "NumberOfNodes": 6,
        "ClusterType": "multi-node"
    },
    "ClusterVersion": "1.0",
    "AllowVersionUpgrade": true,
    "NumberOfNodes": 4,
    "PubliclyAccessible": false,
    "Encrypted": false,
    "ClusterSnapshotCopyStatus": {
        "DestinationRegion": "us-west-1",
        "RetentionPeriod": 15,
        "ManualSnapshotRetentionPeriod": -1
    },
    "Tags": [
        {
            "Key": "mytags",
            "Value": "tag1"
        }
    ],
    "EnhancedVpcRouting": false,
    "IamRoles": [],
    "MaintenanceTrackName": "current",
    "DeferredMaintenanceWindows": [
        {
            "DeferMaintenanceIdentifier": "dfm-mUdVCfDcT1B4SGhw6fyF",
            "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
            "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
        }
    ],
    "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z",
    "ResizeInfo": {
        "ResizeType": "ClassicResize",
        "AllowCancelResize": true
    }
}
}
```

Weitere Informationen finden Sie unter [Größenänderung eines Clusters](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie [ResizeCluster](#) in der AWS CLI Befehlsreferenz.

restore-from-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `restore-from-cluster-snapshot`.

AWS CLI

Einen Cluster anhand eines SnapshotThis Beispiels wiederherstellen stellt einen Cluster aus einem Snapshot wiederher.Command:

```
aws redshift restore-from-cluster-snapshot --cluster-identifizier mycluster-clone --
snapshot-identifizier my-snapshot-id
```

Ergebnis:

```
{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],
    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      }
    ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": \[],
    "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "creating",
    "ClusterIdentifier": "mycluster-clone",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {}
  },
  "ResponseMetadata": {
```

```

    "RequestId": "77fd512b-64e3-11e2-8f5b-e90bd6c77476"
  }
}

```

- API-Einheiten finden Sie [RestoreFromClusterSnapshot](#) in AWS CLI der Befehlsreferenz.

restore-table-from-cluster-snapshot

Das folgende Codebeispiel zeigt die Verwendung `restore-table-from-cluster-snapshot`.

AWS CLI

Um eine Tabelle aus einem Cluster-Snapshot wiederherzustellen

Im folgenden `restore-table-from-cluster-snapshot` Beispiel wird aus der angegebenen Tabelle im angegebenen Cluster-Snapshot eine neue Tabelle erstellt.

```

aws redshift restore-table-from-cluster-snapshot /
--cluster-identifier mycluster /
--snapshot-identifier mycluster-2019-11-19-16-17 /
--source-database-name dev /
--source-schema-name public /
--source-table-name mytable /
--target-database-name dev /
--target-schema-name public /
--new-table-name mytable-clone

```

Ausgabe:

```

{
  "TableRestoreStatus": {
    "TableRestoreRequestId": "a123a12b-abc1-1a1a-a123-a1234ab12345",
    "Status": "PENDING",
    "RequestTime": "2019-12-20T00:20:16.402Z",
    "ClusterIdentifier": "mycluster",
    "SnapshotIdentifier": "mycluster-2019-11-19-16-17",
    "SourceDatabaseName": "dev",
    "SourceSchemaName": "public",
    "SourceTableName": "mytable",
    "TargetDatabaseName": "dev",
    "TargetSchemaName": "public",
    "NewTableName": "mytable-clone"
  }
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Wiederherstellen einer Tabelle aus einem Snapshot](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [RestoreTableFromClusterSnapshot AWS CLI](#) Befehlsreferenz.

revoke-cluster-security-group-ingress

Das folgende Codebeispiel zeigt die Verwendung `revoke-cluster-security-group-ingress`.

AWS CLI

Revoke Access from an EC2 Security Group This example widerruft den Zugriff auf eine benannte EC2 Amazon-Sicherheitsgruppe. Befehl:

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name  
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-  
owner-id 123445677890
```

Wenn Sie den Zugriff auf ein CIDR range This Beispiel widerrufen, wird der Zugriff auf einen Bereich widerrufen. Befehl: CIDR

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name  
mysecuritygroup --cidrip 192.168.100.100/32
```

- API-Einheiten finden Sie in der Befehlsreferenz [RevokeClusterSecurityGroupIngress](#). AWS CLI

revoke-snapshot-access

Das folgende Codebeispiel zeigt die Verwendung `revoke-snapshot-access`.

AWS CLI

Die Autorisierung eines AWS Kontos zur Wiederherstellung widerrufen Ein Snapshot This Beispiel widerruft die Autorisierung des AWS Kontos 444455556666 zur Wiederherstellung des Snapshots `my-snapshot-id`. Standardmäßig erfolgt die Ausgabe im JSON Format. COMMAND:


```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-restore-access 444455556666
```

Ergebnis:

```
{
  "Snapshot": {
    "Status": "available",
    "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
    "EstimatedSecondsToCompletion": 0,
    "AvailabilityZone": "us-east-1a",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "Encrypted": false,
    "OwnerAccount": "111122223333",
    "BackupProgressInMegabytes": 11.0,
    "ElapsedTimeInSeconds": 0,
    "DBName": "dev",
    "CurrentBackupRateInMegabytesPerSecond": 0.1534,
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "ActualIncrementalBackupSizeInMegabytes": 11.0,
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "mycluster",
    "TotalBackupSizeInMegabytes": 20.0,
    "Port": 5439,
    "NumberOfNodes": 2,
    "SnapshotIdentifier": "my-snapshot-id"
  }
}
```

- API-Einheiten finden Sie [RevokeSnapshotAccess](#) in der AWS CLI Befehlsreferenz.

rotate-encryption-key

Das folgende Codebeispiel zeigt die Verwendung von `rotate-encryption-key`.

AWS CLI

Um den Verschlüsselungsschlüssel für einen Cluster zu rotieren

Im folgenden `rotate-encryption-key` Beispiel wird der Verschlüsselungsschlüssel für den angegebenen Cluster rotiert.

```
aws redshift rotate-encryption-key \  
  --cluster-identifier mycluster
```

Ausgabe:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "rotating-keys",  
    "ClusterAvailabilityStatus": "Modifying",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-10T19:25:45.886Z",  
    "AutomatedSnapshotRetentionPeriod": 30,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-a1a123ab",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "ClusterSubnetGroupName": "default",  
    "VpcId": "vpc-a1abc1a1",  
    "AvailabilityZone": "us-west-2a",  
    "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",  
    "PendingModifiedValues": {},  
    "ClusterVersion": "1.0",  
    "AllowVersionUpgrade": true,  
  }  
}
```

```
    "NumberOfNodes": 2,
    "PubliclyAccessible": false,
    "Encrypted": true,
    "Tags": [],
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvnvEXAMPLEKEY",
    "EnhancedVpcRouting": false,
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
        "ApplyStatus": "in-sync"
      }
    ],
    "MaintenanceTrackName": "current",
    "DeferredMaintenanceWindows": [],
    "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
  }
}
```

Weitere Informationen finden Sie unter [Amazon Redshift Database Encryption](#) im Amazon Redshift Cluster Management Guide.

- API-Einheiten finden Sie unter [RotateEncryptionKey AWS CLI](#) Befehlsreferenz.

Amazon Rekognition Rekognition-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Rekognition Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

compare-faces

Das folgende Codebeispiel zeigt die Verwendung `compare-faces`.

Weitere Informationen finden Sie unter [Vergleich von Gesichtern in Bildern](#).

AWS CLI

Um Gesichter in zwei Bildern zu vergleichen

Der folgende `compare-faces` Befehl vergleicht Gesichter in zwei Bildern, die in einem Amazon S3 S3-Bucket gespeichert sind.

```
aws rekognition compare-faces \  
  --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \  
  --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

Ausgabe:

```
{  
  "UnmatchedFaces": [],  
  "FaceMatches": [  
    {  
      "Face": {  
        "BoundingBox": {  
          "Width": 0.12368916720151901,  
          "Top": 0.16007372736930847,  
          "Left": 0.5901257991790771,  
          "Height": 0.25140416622161865  
        },  
        "Confidence": 100.0,  
        "Pose": {  
          "Yaw": -3.7351467609405518,  
          "Roll": -0.10309021919965744,  
          "Pitch": 0.8637830018997192  
        },  
        "Quality": {  
          "Sharpness": 95.51618957519531,  
          "Brightness": 65.29893493652344  
        },  
        "Landmarks": [  

```

```
        {
            "Y": 0.26721030473709106,
            "X": 0.6204193830490112,
            "Type": "eyeLeft"
        },
        {
            "Y": 0.26831310987472534,
            "X": 0.6776827573776245,
            "Type": "eyeRight"
        },
        {
            "Y": 0.3514654338359833,
            "X": 0.6241428852081299,
            "Type": "mouthLeft"
        },
        {
            "Y": 0.35258132219314575,
            "X": 0.6713621020317078,
            "Type": "mouthRight"
        },
        {
            "Y": 0.3140771687030792,
            "X": 0.6428444981575012,
            "Type": "nose"
        }
    ]
},
"Similarity": 100.0
}
],
"SourceImageFace": {
    "BoundingBox": {
        "Width": 0.12368916720151901,
        "Top": 0.16007372736930847,
        "Left": 0.5901257991790771,
        "Height": 0.25140416622161865
    },
    "Confidence": 100.0
}
}
```

Weitere Informationen finden Sie unter [Gesichter in Bildern vergleichen im Amazon Rekognition Developer Guide](#).

- API-Einheiten finden Sie [CompareFaces](#) in der AWS CLI Befehlsreferenz.

create-collection

Das folgende Codebeispiel zeigt, wie man es benutzt `create-collection`.

Weitere Informationen finden Sie unter [Erstellen einer Sammlung](#).

AWS CLI

Um eine Sammlung zu erstellen

Der folgende `create-collection` Befehl erstellt eine Sammlung mit dem angegebenen Namen.

```
aws rekognition create-collection \  
  --collection-id "MyCollection"
```

Ausgabe:

```
{  
  "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/  
MyCollection",  
  "FaceModelVersion": "4.0",  
  "StatusCode": 200  
}
```

Weitere Informationen finden Sie unter [Creating a Collection](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [CreateCollection](#) in der AWS CLI Befehlsreferenz.

create-stream-processor

Das folgende Codebeispiel zeigt, wie man es benutzt `create-stream-processor`.

AWS CLI

Um einen neuen Stream-Prozessor zu erstellen

Im folgenden `create-stream-processor` Beispiel wird ein neuer Stream-Prozessor mit der angegebenen Konfiguration erstellt.

```
aws rekognition create-stream-processor --name my-stream-processor \  
  --input '{"KinesisVideoStream":{"Arn":"arn:aws:kinesisvideo:us-  
west-2:123456789012:stream/macwebcam/1530559711205"}}' \  
  --stream-processor-output '{"KinesisDataStream":{"Arn":"arn:aws:kinesis:us-  
west-2:123456789012:stream/AmazonRekognitionRekStream"}}' \  
  --role-arn arn:aws:iam::123456789012:role/AmazonRekognitionDetect \  
  --settings '{"FaceSearch":  
'{"CollectionId":"MyCollection","FaceMatchThreshold":85.5}{'
```

Ausgabe:

```
{  
  "StreamProcessorArn": "arn:aws:rekognition:us-  
west-2:123456789012:streamprocessor/my-stream-processor"  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Streaming-Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [CreateStreamProcessor](#) in der AWS CLI Befehlsreferenz.

delete-collection

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-collection`.

Weitere Informationen finden Sie unter [Löschen einer Sammlung](#).

AWS CLI

Um eine Sammlung zu löschen

Der folgende `delete-collection` Befehl löscht die angegebene Sammlung.

```
aws rekognition delete-collection \  
  --collection-id MyCollection
```

Ausgabe:

```
{  
  "StatusCode": 200  
}
```

Weitere Informationen finden Sie unter [Löschen einer Sammlung](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DeleteCollection](#) in der AWS CLI Befehlsreferenz.

delete-faces

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-faces`.

Weitere Informationen finden Sie unter [Löschen von Gesichtern aus einer Sammlung](#).

AWS CLI

Um Gesichter aus einer Sammlung zu löschen

Der folgende `delete-faces` Befehl löscht die angegebene Fläche aus einer Sammlung.

```
aws rekognition delete-faces \  
  --collection-id MyCollection \  
  --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

Ausgabe:

```
{  
  "DeletedFaces": [  
    "0040279c-0178-436e-b70a-e61b074e96b0"  
  ]  
}
```

Weitere Informationen finden Sie unter [Löschen von Gesichtern aus einer Sammlung](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DeleteFaces](#) in der AWS CLI Befehlsreferenz.

delete-stream-processor

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-stream-processor`.

AWS CLI

Um einen Stream-Prozessor zu löschen

Der folgende `delete-stream-processor` Befehl löscht den angegebenen Stream-Prozessor.

```
aws rekognition delete-stream-processor \  
  --name my-stream-processor
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Streaming-Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DeleteStreamProcessor](#) in der AWS CLI Befehlsreferenz.

describe-collection

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-collection`.

Weitere Informationen finden Sie unter [Beschreiben einer Sammlung](#).

AWS CLI

Um eine Sammlung zu beschreiben

Im folgenden `describe-collection` Beispiel werden die Details zur angegebenen Sammlung angezeigt.

```
aws rekognition describe-collection \  
  --collection-id MyCollection
```

Ausgabe:

```
{  
  "FaceCount": 200,  
  "CreationTimestamp": 1569444828.274,  
  "CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/  
MyCollection",  
  "FaceModelVersion": "4.0"  
}
```

Weitere Informationen finden Sie unter [Beschreibung einer Sammlung](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DescribeCollection](#) in der AWS CLI Befehlsreferenz.

describe-stream-processor

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-stream-processor`.

AWS CLI

Um Informationen über einen Stream-Prozessor zu erhalten

Der folgende `describe-stream-processor` Befehl zeigt Details zum angegebenen Stream-Prozessor an.

```
aws rekognition describe-stream-processor \  
  --name my-stream-processor
```

Ausgabe:

```
{  
  "Status": "STOPPED",  
  "Name": "my-stream-processor",  
  "LastUpdateTimestamp": 1532449292.712,  
  "Settings": {  
    "FaceSearch": {  
      "FaceMatchThreshold": 80.0,  
      "CollectionId": "my-collection"  
    }  
  },  
  "RoleArn": "arn:aws:iam::123456789012:role/AmazonRekognitionDetectStream",  
  "StreamProcessorArn": "arn:aws:rekognition:us-west-2:123456789012:streamprocessor/my-stream-processor",  
  "Output": {  
    "KinesisDataStream": {  
      "Arn": "arn:aws:kinesis:us-west-2:123456789012:stream/AmazonRekognitionRekStream"  
    }  
  },  
  "Input": {  
    "KinesisVideoStream": {  
      "Arn": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/macwebcam/123456789012"  
    }  
  },  
  "CreationTimestamp": 1532449292.712  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Streaming-Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DescribeStreamProcessor](#) in der AWS CLI Befehlsreferenz.

detect-faces

Das folgende Codebeispiel zeigt, wie man es benutzt `detect-faces`.

Weitere Informationen finden Sie unter [Erkennen von Gesichtern in einem Bild](#).

AWS CLI

Um Gesichter in einem Bild zu erkennen

Der folgende `detect-faces` Befehl erkennt Gesichter in dem angegebenen Bild, das in einem Amazon S3 S3-Bucket gespeichert ist.

```
aws rekognition detect-faces \  
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \  
  --attributes "ALL"
```

Ausgabe:

```
{  
  "FaceDetails": [  
    {  
      "Confidence": 100.0,  
      "Eyeglasses": {  
        "Confidence": 98.91107940673828,  
        "Value": false  
      },  
      "Sunglasses": {  
        "Confidence": 99.7966537475586,  
        "Value": false  
      },  
      "Gender": {  
        "Confidence": 99.56611633300781,  
        "Value": "Male"  
      },  
      "Landmarks": [  
        {  
          "Y": 0.26721030473709106,
```

```
    "X": 0.6204193830490112,  
    "Type": "eyeLeft"  
  },  
  {  
    "Y": 0.26831310987472534,  
    "X": 0.6776827573776245,  
    "Type": "eyeRight"  
  },  
  {  
    "Y": 0.3514654338359833,  
    "X": 0.6241428852081299,  
    "Type": "mouthLeft"  
  },  
  {  
    "Y": 0.35258132219314575,  
    "X": 0.6713621020317078,  
    "Type": "mouthRight"  
  },  
  {  
    "Y": 0.3140771687030792,  
    "X": 0.6428444981575012,  
    "Type": "nose"  
  },  
  {  
    "Y": 0.24662546813488007,  
    "X": 0.6001564860343933,  
    "Type": "leftEyeBrowLeft"  
  },  
  {  
    "Y": 0.24326619505882263,  
    "X": 0.6303644776344299,  
    "Type": "leftEyeBrowRight"  
  },  
  {  
    "Y": 0.23818562924861908,  
    "X": 0.6146903038024902,  
    "Type": "leftEyeBrowUp"  
  },  
  {  
    "Y": 0.24373626708984375,  
    "X": 0.6640064716339111,  
    "Type": "rightEyeBrowLeft"  
  },  
  {
```

```
        "Y": 0.24877218902111053,  
        "X": 0.7025929093360901,  
        "Type": "rightEyeBrowRight"  
    },  
    {  
        "Y": 0.23938551545143127,  
        "X": 0.6823262572288513,  
        "Type": "rightEyeBrowUp"  
    },  
    {  
        "Y": 0.265746533870697,  
        "X": 0.6112898588180542,  
        "Type": "leftEyeLeft"  
    },  
    {  
        "Y": 0.2676128149032593,  
        "X": 0.6317071914672852,  
        "Type": "leftEyeRight"  
    },  
    {  
        "Y": 0.262735515832901,  
        "X": 0.6201658248901367,  
        "Type": "leftEyeUp"  
    },  
    {  
        "Y": 0.27025148272514343,  
        "X": 0.6206279993057251,  
        "Type": "leftEyeDown"  
    },  
    {  
        "Y": 0.268223375082016,  
        "X": 0.6658390760421753,  
        "Type": "rightEyeLeft"  
    },  
    {  
        "Y": 0.2672517001628876,  
        "X": 0.687832236289978,  
        "Type": "rightEyeRight"  
    },  
    {  
        "Y": 0.26383838057518005,  
        "X": 0.6769183874130249,  
        "Type": "rightEyeUp"  
    },  
    },
```

```
{
  "Y": 0.27138751745224,
  "X": 0.676596462726593,
  "Type": "rightEyeDown"
},
{
  "Y": 0.32283174991607666,
  "X": 0.6350004076957703,
  "Type": "noseLeft"
},
{
  "Y": 0.3219289481639862,
  "X": 0.6567046642303467,
  "Type": "noseRight"
},
{
  "Y": 0.3420318365097046,
  "X": 0.6450609564781189,
  "Type": "mouthUp"
},
{
  "Y": 0.3664324879646301,
  "X": 0.6455618143081665,
  "Type": "mouthDown"
},
{
  "Y": 0.26721030473709106,
  "X": 0.6204193830490112,
  "Type": "leftPupil"
},
{
  "Y": 0.26831310987472534,
  "X": 0.6776827573776245,
  "Type": "rightPupil"
},
{
  "Y": 0.26343393325805664,
  "X": 0.5946047306060791,
  "Type": "upperJawlineLeft"
},
{
  "Y": 0.3543180525302887,
  "X": 0.6044883728027344,
  "Type": "midJawlineLeft"
}
```

```
    },
    {
      "Y": 0.4084877669811249,
      "X": 0.6477024555206299,
      "Type": "chinBottom"
    },
    {
      "Y": 0.3562754988670349,
      "X": 0.707981526851654,
      "Type": "midJawlineRight"
    },
    {
      "Y": 0.26580461859703064,
      "X": 0.7234612107276917,
      "Type": "upperJawlineRight"
    }
  ],
  "Pose": {
    "Yaw": -3.7351467609405518,
    "Roll": -0.10309021919965744,
    "Pitch": 0.8637830018997192
  },
  "Emotions": [
    {
      "Confidence": 8.74203109741211,
      "Type": "SURPRISED"
    },
    {
      "Confidence": 2.501944065093994,
      "Type": "ANGRY"
    },
    {
      "Confidence": 0.7378743290901184,
      "Type": "DISGUSTED"
    },
    {
      "Confidence": 3.5296201705932617,
      "Type": "HAPPY"
    },
    {
      "Confidence": 1.7162904739379883,
      "Type": "SAD"
    }
  ]
}
```

```
        "Confidence": 9.518536567687988,
        "Type": "CONFUSED"
    },
    {
        "Confidence": 0.45474427938461304,
        "Type": "FEAR"
    },
    {
        "Confidence": 72.79895782470703,
        "Type": "CALM"
    }
],
"AgeRange": {
    "High": 48,
    "Low": 32
},
"EyesOpen": {
    "Confidence": 98.93987274169922,
    "Value": true
},
"BoundingBox": {
    "Width": 0.12368916720151901,
    "Top": 0.16007372736930847,
    "Left": 0.5901257991790771,
    "Height": 0.25140416622161865
},
"Smile": {
    "Confidence": 93.4493179321289,
    "Value": false
},
"MouthOpen": {
    "Confidence": 90.53053283691406,
    "Value": false
},
"Quality": {
    "Sharpness": 95.51618957519531,
    "Brightness": 65.29893493652344
},
"Mustache": {
    "Confidence": 89.85221099853516,
    "Value": false
},
"Beard": {
    "Confidence": 86.1991195678711,
```



```
        "Value": true
      }
    ]
  }
```

Weitere Informationen finden Sie unter [Erkennen von Gesichtern in einem Bild](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DetectFaces](#) in der AWS CLI Befehlsreferenz.

detect-labels

Das folgende Codebeispiel zeigt, wie man es benutzt `detect-labels`.

Weitere Informationen finden Sie unter [Erkennen von Labels in einem Bild](#).

AWS CLI

Um ein Etikett in einem Bild zu erkennen

Das folgende `detect-labels` Beispiel erkennt Szenen und Objekte in einem Bild, das in einem Amazon S3 S3-Bucket gespeichert ist.

```
aws rekognition detect-labels \
  --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

Ausgabe:

```
{
  "Labels": [
    {
      "Instances": [],
      "Confidence": 99.15271759033203,
      "Parents": [
        {
          "Name": "Vehicle"
        },
        {
          "Name": "Transportation"
        }
      ],
      "Name": "Automobile"
    }
  ]
}
```

```
  },
  {
    "Instances": [],
    "Confidence": 99.15271759033203,
    "Parents": [
      {
        "Name": "Transportation"
      }
    ],
    "Name": "Vehicle"
  },
  {
    "Instances": [],
    "Confidence": 99.15271759033203,
    "Parents": [],
    "Name": "Transportation"
  },
  {
    "Instances": [
      {
        "BoundingBox": {
          "Width": 0.10616336017847061,
          "Top": 0.5039216876029968,
          "Left": 0.0037978808395564556,
          "Height": 0.18528179824352264
        },
        "Confidence": 99.15271759033203
      },
      {
        "BoundingBox": {
          "Width": 0.2429988533258438,
          "Top": 0.5251884460449219,
          "Left": 0.7309805154800415,
          "Height": 0.21577216684818268
        },
        "Confidence": 99.1286392211914
      },
      {
        "BoundingBox": {
          "Width": 0.14233611524105072,
          "Top": 0.5333095788955688,
          "Left": 0.6494812965393066,
          "Height": 0.15528248250484467
        }
      }
    ]
  },
```

```
    "Confidence": 98.48368072509766
  },
  {
    "BoundingBox": {
      "Width": 0.11086395382881165,
      "Top": 0.5354844927787781,
      "Left": 0.10355594009160995,
      "Height": 0.10271988064050674
    },
    "Confidence": 96.45606231689453
  },
  {
    "BoundingBox": {
      "Width": 0.06254628300666809,
      "Top": 0.5573825240135193,
      "Left": 0.46083059906959534,
      "Height": 0.053911514580249786
    },
    "Confidence": 93.65448760986328
  },
  {
    "BoundingBox": {
      "Width": 0.10105438530445099,
      "Top": 0.534368634223938,
      "Left": 0.5743985772132874,
      "Height": 0.12226245552301407
    },
    "Confidence": 93.06217193603516
  },
  {
    "BoundingBox": {
      "Width": 0.056389667093753815,
      "Top": 0.5235804319381714,
      "Left": 0.9427769780158997,
      "Height": 0.17163699865341187
    },
    "Confidence": 92.6864013671875
  },
  {
    "BoundingBox": {
      "Width": 0.06003860384225845,
      "Top": 0.5441341400146484,
      "Left": 0.22409997880458832,
      "Height": 0.06737709045410156
    }
  }
}
```

```
    },
    "Confidence": 90.4227066040039
  },
  {
    "BoundingBox": {
      "Width": 0.02848697081208229,
      "Top": 0.5107086896896362,
      "Left": 0,
      "Height": 0.19150497019290924
    },
    "Confidence": 86.65286254882812
  },
  {
    "BoundingBox": {
      "Width": 0.04067881405353546,
      "Top": 0.5566273927688599,
      "Left": 0.316415935754776,
      "Height": 0.03428703173995018
    },
    "Confidence": 85.36471557617188
  },
  {
    "BoundingBox": {
      "Width": 0.043411049991846085,
      "Top": 0.5394920110702515,
      "Left": 0.18293385207653046,
      "Height": 0.0893595889210701
    },
    "Confidence": 82.21705627441406
  },
  {
    "BoundingBox": {
      "Width": 0.031183116137981415,
      "Top": 0.5579366683959961,
      "Left": 0.2853088080883026,
      "Height": 0.03989990055561066
    },
    "Confidence": 81.0157470703125
  },
  {
    "BoundingBox": {
      "Width": 0.031113790348172188,
      "Top": 0.5504819750785828,
      "Left": 0.2580395042896271,
```

```
        "Height": 0.056484755128622055
      },
      "Confidence": 56.13441467285156
    },
    {
      "BoundingBox": {
        "Width": 0.08586374670267105,
        "Top": 0.5438792705535889,
        "Left": 0.5128012895584106,
        "Height": 0.08550430089235306
      },
      "Confidence": 52.37760925292969
    }
  ],
  "Confidence": 99.15271759033203,
  "Parents": [
    {
      "Name": "Vehicle"
    },
    {
      "Name": "Transportation"
    }
  ],
  "Name": "Car"
},
{
  "Instances": [],
  "Confidence": 98.9914321899414,
  "Parents": [],
  "Name": "Human"
},
{
  "Instances": [
    {
      "BoundingBox": {
        "Width": 0.19360728561878204,
        "Top": 0.35072067379951477,
        "Left": 0.43734854459762573,
        "Height": 0.2742200493812561
      },
      "Confidence": 98.9914321899414
    },
    {
      "BoundingBox": {
```

```
        "Width": 0.03801717236638069,
        "Top": 0.5010883808135986,
        "Left": 0.9155802130699158,
        "Height": 0.06597328186035156
    },
    "Confidence": 85.02790832519531
}
],
"Confidence": 98.9914321899414,
"Parents": [],
"Name": "Person"
},
{
    "Instances": [],
    "Confidence": 93.24951934814453,
    "Parents": [],
    "Name": "Machine"
},
{
    "Instances": [
        {
            "BoundingBox": {
                "Width": 0.03561960905790329,
                "Top": 0.6468243598937988,
                "Left": 0.7850857377052307,
                "Height": 0.08878646790981293
            },
            "Confidence": 93.24951934814453
        },
        {
            "BoundingBox": {
                "Width": 0.02217046171426773,
                "Top": 0.6149078607559204,
                "Left": 0.04757237061858177,
                "Height": 0.07136218994855881
            },
            "Confidence": 91.5025863647461
        },
        {
            "BoundingBox": {
                "Width": 0.016197510063648224,
                "Top": 0.6274210214614868,
                "Left": 0.6472989320755005,
                "Height": 0.04955997318029404
            }
        }
    ]
}
```

```
    },
    "Confidence": 85.14686584472656
  },
  {
    "BoundingBox": {
      "Width": 0.020207518711686134,
      "Top": 0.6348286867141724,
      "Left": 0.7295016646385193,
      "Height": 0.07059963047504425
    },
    "Confidence": 83.34547424316406
  },
  {
    "BoundingBox": {
      "Width": 0.020280985161662102,
      "Top": 0.6171894669532776,
      "Left": 0.08744934946298599,
      "Height": 0.05297485366463661
    },
    "Confidence": 79.9981460571289
  },
  {
    "BoundingBox": {
      "Width": 0.018318990245461464,
      "Top": 0.623889148235321,
      "Left": 0.6836880445480347,
      "Height": 0.06730121374130249
    },
    "Confidence": 78.87144470214844
  },
  {
    "BoundingBox": {
      "Width": 0.021310249343514442,
      "Top": 0.6167286038398743,
      "Left": 0.004064912907779217,
      "Height": 0.08317798376083374
    },
    "Confidence": 75.89361572265625
  },
  {
    "BoundingBox": {
      "Width": 0.03604431077837944,
      "Top": 0.7030032277107239,
      "Left": 0.9254803657531738,
```

```
        "Height": 0.04569442570209503
      },
      "Confidence": 64.402587890625
    },
    {
      "BoundingBox": {
        "Width": 0.009834849275648594,
        "Top": 0.5821820497512817,
        "Left": 0.28094568848609924,
        "Height": 0.01964157074689865
      },
      "Confidence": 62.79907989501953
    },
    {
      "BoundingBox": {
        "Width": 0.01475677452981472,
        "Top": 0.6137543320655823,
        "Left": 0.5950819253921509,
        "Height": 0.039063986390829086
      },
      "Confidence": 59.40483474731445
    }
  ],
  "Confidence": 93.24951934814453,
  "Parents": [
    {
      "Name": "Machine"
    }
  ],
  "Name": "Wheel"
},
{
  "Instances": [],
  "Confidence": 92.61514282226562,
  "Parents": [],
  "Name": "Road"
},
{
  "Instances": [],
  "Confidence": 92.37877655029297,
  "Parents": [
    {
      "Name": "Person"
    }
  ]
}
```



```
    ],
    "Name": "Sport"
  },
  {
    "Instances": [],
    "Confidence": 92.37877655029297,
    "Parents": [
      {
        "Name": "Person"
      }
    ],
    "Name": "Sports"
  },
  {
    "Instances": [
      {
        "BoundingBox": {
          "Width": 0.12326609343290329,
          "Top": 0.6332163214683533,
          "Left": 0.44815489649772644,
          "Height": 0.058117982000112534
        },
        "Confidence": 92.37877655029297
      }
    ],
    "Confidence": 92.37877655029297,
    "Parents": [
      {
        "Name": "Person"
      },
      {
        "Name": "Sport"
      }
    ],
    "Name": "Skateboard"
  },
  {
    "Instances": [],
    "Confidence": 90.62931060791016,
    "Parents": [
      {
        "Name": "Person"
      }
    ]
  },
],
```

```
    "Name": "Pedestrian"
  },
  {
    "Instances": [],
    "Confidence": 88.81334686279297,
    "Parents": [],
    "Name": "Asphalt"
  },
  {
    "Instances": [],
    "Confidence": 88.81334686279297,
    "Parents": [],
    "Name": "Tarmac"
  },
  {
    "Instances": [],
    "Confidence": 88.23201751708984,
    "Parents": [],
    "Name": "Path"
  },
  {
    "Instances": [],
    "Confidence": 80.26520538330078,
    "Parents": [],
    "Name": "Urban"
  },
  {
    "Instances": [],
    "Confidence": 80.26520538330078,
    "Parents": [
      {
        "Name": "Building"
      },
      {
        "Name": "Urban"
      }
    ],
    "Name": "Town"
  },
  {
    "Instances": [],
    "Confidence": 80.26520538330078,
    "Parents": [],
    "Name": "Building"
```

```
  },
  {
    "Instances": [],
    "Confidence": 80.26520538330078,
    "Parents": [
      {
        "Name": "Building"
      },
      {
        "Name": "Urban"
      }
    ],
    "Name": "City"
  },
  {
    "Instances": [],
    "Confidence": 78.37934875488281,
    "Parents": [
      {
        "Name": "Car"
      },
      {
        "Name": "Vehicle"
      },
      {
        "Name": "Transportation"
      }
    ],
    "Name": "Parking Lot"
  },
  {
    "Instances": [],
    "Confidence": 78.37934875488281,
    "Parents": [
      {
        "Name": "Car"
      },
      {
        "Name": "Vehicle"
      },
      {
        "Name": "Transportation"
      }
    ],
  },
```

```
    "Name": "Parking"
  },
  {
    "Instances": [],
    "Confidence": 74.37590026855469,
    "Parents": [
      {
        "Name": "Building"
      },
      {
        "Name": "Urban"
      },
      {
        "Name": "City"
      }
    ],
    "Name": "Downtown"
  },
  {
    "Instances": [],
    "Confidence": 69.84622955322266,
    "Parents": [
      {
        "Name": "Road"
      }
    ],
    "Name": "Intersection"
  },
  {
    "Instances": [],
    "Confidence": 57.68518829345703,
    "Parents": [
      {
        "Name": "Sports Car"
      },
      {
        "Name": "Car"
      },
      {
        "Name": "Vehicle"
      },
      {
        "Name": "Transportation"
      }
    ]
  }
```

```
    ],
    "Name": "Coupe"
  },
  {
    "Instances": [],
    "Confidence": 57.68518829345703,
    "Parents": [
      {
        "Name": "Car"
      },
      {
        "Name": "Vehicle"
      },
      {
        "Name": "Transportation"
      }
    ],
    "Name": "Sports Car"
  },
  {
    "Instances": [],
    "Confidence": 56.59492111206055,
    "Parents": [
      {
        "Name": "Path"
      }
    ],
    "Name": "Sidewalk"
  },
  {
    "Instances": [],
    "Confidence": 56.59492111206055,
    "Parents": [
      {
        "Name": "Path"
      }
    ],
    "Name": "Pavement"
  },
  {
    "Instances": [],
    "Confidence": 55.58770751953125,
    "Parents": [
      {
```

```

        "Name": "Building"
      },
      {
        "Name": "Urban"
      }
    ],
    "Name": "Neighborhood"
  }
],
"LabelModelVersion": "2.0"
}

```

Weitere Informationen finden Sie unter [Erkennen von Labels in einem Bild](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie unter [DetectLabels AWS CLI](#) Befehlsreferenz.

detect-moderation-labels

Das folgende Codebeispiel zeigt, wie man es benutzt `detect-moderation-labels`.

Weitere Informationen finden Sie unter [Erkennen von unangemessenen Bildern](#).

AWS CLI

Um unsichere Inhalte in einem Bild zu erkennen

Der folgende `detect-moderation-labels` Befehl erkennt unsichere Inhalte im angegebenen Bild, das in einem Amazon S3 S3-Bucket gespeichert ist.

```

aws rekognition detect-moderation-labels \
  --image "S3Object={Bucket=MyImageS3Bucket,Name=gun.jpg}"

```

Ausgabe:

```

{
  "ModerationModelVersion": "3.0",
  "ModerationLabels": [
    {
      "Confidence": 97.29618072509766,
      "ParentName": "Violence",
      "Name": "Weapon Violence"
    }
  ],
}

```

```

    {
      "Confidence": 97.29618072509766,
      "ParentName": "",
      "Name": "Violence"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Erkennen unsicherer Bilder](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [DetectModerationLabels](#) in AWS CLI der Befehlsreferenz.

detect-text

Das folgende Codebeispiel zeigt, wie man es benutzt `detect-text`.

Weitere Informationen finden Sie unter [Erkennen von Text in einem Bild](#).

AWS CLI

Um Text in einem Bild zu erkennen

Der folgende `detect-text` Befehl erkennt Text im angegebenen Bild.

```

aws rekognition detect-text \
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'

```

Ausgabe:

```

{
  "TextDetections": [
    {
      "Geometry": {
        "BoundingBox": {
          "Width": 0.24624845385551453,
          "Top": 0.28288066387176514,
          "Left": 0.391388863325119,
          "Height": 0.022687450051307678
        },
        "Polygon": [
          {
            "Y": 0.28288066387176514,

```

```
        "X": 0.391388863325119
      },
      {
        "Y": 0.2826388478279114,
        "X": 0.6376373171806335
      },
      {
        "Y": 0.30532628297805786,
        "X": 0.637677013874054
      },
      {
        "Y": 0.305568128824234,
        "X": 0.39142853021621704
      }
    ]
  },
  "Confidence": 94.35709381103516,
  "DetectedText": "ESTD 1882",
  "Type": "LINE",
  "Id": 0
},
{
  "Geometry": {
    "BoundingBox": {
      "Width": 0.33933889865875244,
      "Top": 0.32603850960731506,
      "Left": 0.34534579515457153,
      "Height": 0.07126858830451965
    },
    "Polygon": [
      {
        "Y": 0.32603850960731506,
        "X": 0.34534579515457153
      },
      {
        "Y": 0.32633158564567566,
        "X": 0.684684693813324
      },
      {
        "Y": 0.3976001739501953,
        "X": 0.684575080871582
      },
      {
        "Y": 0.3973070979118347,
```



```
        "X": 0.345236212015152
      }
    ]
  },
  "Confidence": 99.95779418945312,
  "DetectedText": "BRAINS",
  "Type": "LINE",
  "Id": 1
},
{
  "Confidence": 97.22098541259766,
  "Geometry": {
    "BoundingBox": {
      "Width": 0.061079490929841995,
      "Top": 0.2843210697174072,
      "Left": 0.391391396522522,
      "Height": 0.021029088646173477
    },
    "Polygon": [
      {
        "Y": 0.2843210697174072,
        "X": 0.391391396522522
      },
      {
        "Y": 0.2828207015991211,
        "X": 0.4524524509906769
      },
      {
        "Y": 0.3038259446620941,
        "X": 0.4534534513950348
      },
      {
        "Y": 0.30532634258270264,
        "X": 0.3923923969268799
      }
    ]
  },
  "DetectedText": "ESTD",
  "ParentId": 0,
  "Type": "WORD",
  "Id": 2
},
{
  "Confidence": 91.49320983886719,
```

```
    "Geometry": {
      "BoundingBox": {
        "Width": 0.07007007300853729,
        "Top": 0.2828207015991211,
        "Left": 0.5675675868988037,
        "Height": 0.02250562608242035
      },
      "Polygon": [
        {
          "Y": 0.2828207015991211,
          "X": 0.5675675868988037
        },
        {
          "Y": 0.2828207015991211,
          "X": 0.6376376152038574
        },
        {
          "Y": 0.30532634258270264,
          "X": 0.6376376152038574
        },
        {
          "Y": 0.30532634258270264,
          "X": 0.5675675868988037
        }
      ]
    },
    "DetectedText": "1882",
    "ParentId": 0,
    "Type": "WORD",
    "Id": 3
  },
  {
    "Confidence": 99.95779418945312,
    "Geometry": {
      "BoundingBox": {
        "Width": 0.33933934569358826,
        "Top": 0.32633158564567566,
        "Left": 0.3453453481197357,
        "Height": 0.07127484679222107
      },
      "Polygon": [
        {
          "Y": 0.32633158564567566,
          "X": 0.3453453481197357
```

```

    },
    {
      "Y": 0.32633158564567566,
      "X": 0.684684693813324
    },
    {
      "Y": 0.39759939908981323,
      "X": 0.6836836934089661
    },
    {
      "Y": 0.39684921503067017,
      "X": 0.3453453481197357
    }
  ]
},
"DetectedText": "BRAINS",
"ParentId": 1,
"Type": "WORD",
"Id": 4
}
]
}

```

- API-Einheiten finden Sie [DetectText](#) in der AWS CLI Befehlsreferenz.

disassociate-faces

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-faces`.

AWS CLI

```
aws rekognition disassociate-faces --face-ids list-of-face-ids
--user-id user-id --collection-id collection-name --region region-name
```

- API-Einheiten finden Sie [DisassociateFaces](#) in der AWS CLI Befehlsreferenz.

get-celebrity-info

Das folgende Codebeispiel zeigt, wie man es benutzt `get-celebrity-info`.

AWS CLI

Um Informationen über eine Berühmtheit zu erhalten

Der folgende `get-celebrity-info` Befehl zeigt Informationen über den angegebenen Star an. Der `id` Parameter stammt aus einem früheren Aufruf von `recognize-celebrities`.

```
aws rekognition get-celebrity-info --id nnnnnnn
```

Ausgabe:

```
{
  "Name": "Celeb A",
  "Urls": [
    "www.imdb.com/name/aaaaaaaaa"
  ]
}
```

Weitere Informationen finden Sie unter [Informationen über Prominente](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetCelebrityInfo](#) in der AWS CLI Befehlsreferenz.

get-celebrity-recognition

Das folgende Codebeispiel zeigt, wie man es benutzt `get-celebrity-recognition`.

AWS CLI

Um die Ergebnisse einer Operation zur Anerkennung von Prominenten zu erhalten

Mit dem folgenden `get-celebrity-recognition` Befehl werden die Ergebnisse einer Operation zur Erkennung von Prominenten angezeigt, die Sie zuvor durch einen Aufruf `start-celebrity-recognition` gestartet haben.

```
aws rekognition get-celebrity-recognition \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Ausgabe:

```
{
```

```
"NextToken": "3D01Clx1CiT31VsRDkA03IybLb/h5AtDWSGuhYi
+N1FIJwwPtAkuKzDhL2rV3GcwmNt77+12",
  "Celebrities": [
    {
      "Timestamp": 0,
      "Celebrity": {
        "Confidence": 96.0,
        "Face": {
          "BoundingBox": {
            "Width": 0.70333331823349,
            "Top": 0.16750000417232513,
            "Left": 0.19555555284023285,
            "Height": 0.3956249952316284
          },
          "Landmarks": [
            {
              "Y": 0.31031012535095215,
              "X": 0.441436767578125,
              "Type": "eyeLeft"
            },
            {
              "Y": 0.3081788718700409,
              "X": 0.6437258720397949,
              "Type": "eyeRight"
            },
            {
              "Y": 0.39542075991630554,
              "X": 0.5572493076324463,
              "Type": "nose"
            },
            {
              "Y": 0.4597957134246826,
              "X": 0.4579732120037079,
              "Type": "mouthLeft"
            },
            {
              "Y": 0.45688048005104065,
              "X": 0.6349081993103027,
              "Type": "mouthRight"
            }
          ],
          "Pose": {
            "Yaw": 8.943398475646973,
            "Roll": -2.0309247970581055,
```

```
        "Pitch": -0.5674862861633301
      },
      "Quality": {
        "Sharpness": 99.40211486816406,
        "Brightness": 89.47132110595703
      },
      "Confidence": 99.99861145019531
    },
    "Name": "CelebrityA",
    "Urls": [
      "www.imdb.com/name/111111111"
    ],
    "Id": "nnnnnn"
  }
},
{
  "Timestamp": 467,
  "Celebrity": {
    "Confidence": 99.0,
    "Face": {
      "BoundingBox": {
        "Width": 0.6877777576446533,
        "Top": 0.18437500298023224,
        "Left": 0.20555555820465088,
        "Height": 0.3868750035762787
      },
      "Landmarks": [
        {
          "Y": 0.31895750761032104,
          "X": 0.4411413371562958,
          "Type": "eyeLeft"
        },
        {
          "Y": 0.3140959143638611,
          "X": 0.6523157954216003,
          "Type": "eyeRight"
        },
        {
          "Y": 0.4016456604003906,
          "X": 0.5682755708694458,
          "Type": "nose"
        },
        {
          "Y": 0.46894142031669617,
```

```

        "X": 0.4597797095775604,
        "Type": "mouthLeft"
    },
    {
        "Y": 0.46971091628074646,
        "X": 0.6286435127258301,
        "Type": "mouthRight"
    }
],
"Pose": {
    "Yaw": 10.433465957641602,
    "Roll": -3.347442388534546,
    "Pitch": 1.3709543943405151
},
"Quality": {
    "Sharpness": 99.5531005859375,
    "Brightness": 88.5764389038086
},
"Confidence": 99.99148559570312
},
"Name": "Jane Celebrity",
"Urls": [
    "www.imdb.com/name/1111111111"
],
"Id": "nnnnnn"
}
}
],
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.978118896484375,
    "Codec": "h264",
    "DurationMillis": 4570,
    "FrameHeight": 1920,
    "FrameWidth": 1080
}
}
}

```

Weitere Informationen finden Sie unter [Erkennen von Prominenten in einem gespeicherten Video](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetCelebrityRecognition](#) in der AWS CLI Befehlsreferenz.

get-content-moderation

Das folgende Codebeispiel zeigt, wie man es benutzt `get-content-moderation`.

AWS CLI

Um die Ergebnisse eines unsicheren Inhaltsvorgangs abzurufen

Der folgende `get-content-moderation` Befehl zeigt die Ergebnisse eines unsicheren Inhaltsvorgangs an, den Sie zuvor durch einen Aufruf gestartet haben. `start-content-moderation`

```
aws rekognition get-content-moderation \  
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Ausgabe:

```
{  
  "NextToken": "dlhcKMHMzpCBGFukz6I03JMcWiJAamCVhXHT3r6b4b5Tfbyw3q7o+Jeezt  
+ZpgfOnW9FCCgQ",  
  "ModerationLabels": [  
    {  
      "Timestamp": 0,  
      "ModerationLabel": {  
        "Confidence": 97.39583587646484,  
        "ParentName": "",  
        "Name": "Violence"  
      }  
    },  
    {  
      "Timestamp": 0,  
      "ModerationLabel": {  
        "Confidence": 97.39583587646484,  
        "ParentName": "Violence",  
        "Name": "Weapon Violence"  
      }  
    }  
  ],  
  "JobStatus": "SUCCEEDED",  
  "VideoMetadata": {  
    "Format": "QuickTime / MOV",  
    "FrameRate": 29.97515869140625,  
    "Codec": "h264",
```



```
    "DurationMillis": 6039,  
    "FrameHeight": 1920,  
    "FrameWidth": 1080  
  }  
}
```

Weitere Informationen finden Sie unter [Erkennen unsicherer gespeicherter Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetContentModeration](#) in AWS CLI der Befehlsreferenz.

get-face-detection

Das folgende Codebeispiel zeigt, wie man es benutzt `get-face-detection`.

AWS CLI

Um die Ergebnisse einer Gesichtserkennungsoperation zu erhalten

Der folgende `get-face-detection` Befehl zeigt die Ergebnisse einer Gesichtserkennung an, die Sie zuvor durch einen Aufruf gestartet haben `start-face-detection`.

```
aws rekognition get-face-detection \  
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Ausgabe:

```
{  
  "Faces": [  
    {  
      "Timestamp": 467,  
      "Face": {  
        "BoundingBox": {  
          "Width": 0.1560753583908081,  
          "Top": 0.13555361330509186,  
          "Left": -0.0952017530798912,  
          "Height": 0.6934483051300049  
        },  
        "Landmarks": [  
          {  
            "Y": 0.4013825058937073,  
            "X": -0.041750285774469376,  
            "Type": "eyeLeft"  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
    },
    {
      "Y": 0.41695496439933777,
      "X": 0.027979329228401184,
      "Type": "eyeRight"
    },
    {
      "Y": 0.6375303268432617,
      "X": -0.04034662991762161,
      "Type": "mouthLeft"
    },
    {
      "Y": 0.6497718691825867,
      "X": 0.013960429467260838,
      "Type": "mouthRight"
    },
    {
      "Y": 0.5238034129142761,
      "X": 0.008022055961191654,
      "Type": "nose"
    }
  ],
  "Pose": {
    "Yaw": -58.07863998413086,
    "Roll": 1.9384294748306274,
    "Pitch": -24.66305160522461
  },
  "Quality": {
    "Sharpness": 83.14741516113281,
    "Brightness": 25.75942611694336
  },
  "Confidence": 87.7622299194336
}
},
{
  "Timestamp": 967,
  "Face": {
    "BoundingBox": {
      "Width": 0.28559377789497375,
      "Top": 0.19436298310756683,
      "Left": 0.024553587660193443,
      "Height": 0.7216082215309143
    },
    "Landmarks": [
```

```
        {
            "Y": 0.4650231599807739,
            "X": 0.16269078850746155,
            "Type": "eyeLeft"
        },
        {
            "Y": 0.4843238294124603,
            "X": 0.2782580852508545,
            "Type": "eyeRight"
        },
        {
            "Y": 0.71530681848526,
            "X": 0.1741468608379364,
            "Type": "mouthLeft"
        },
        {
            "Y": 0.7310671210289001,
            "X": 0.26857468485832214,
            "Type": "mouthRight"
        },
        {
            "Y": 0.582602322101593,
            "X": 0.2566150426864624,
            "Type": "nose"
        }
    ],
    "Pose": {
        "Yaw": 11.487052917480469,
        "Roll": 5.074230670928955,
        "Pitch": 15.396159172058105
    },
    "Quality": {
        "Sharpness": 73.32209777832031,
        "Brightness": 54.96497344970703
    },
    "Confidence": 99.99998474121094
}
}
],
"NextToken":
"OzL223pDKy91160/02KXRqFIEAwxy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
```

```
    "FrameRate": 29.970617294311523,  
    "Codec": "h264",  
    "DurationMillis": 6806,  
    "FrameHeight": 1080,  
    "FrameWidth": 1920  
  }  
}
```

Weitere Informationen finden Sie unter [Erkennen von Gesichtern in einem gespeicherten Video](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetFaceDetection](#) in der AWS CLI Befehlsreferenz.

get-face-search

Das folgende Codebeispiel zeigt, wie man es benutzt `get-face-search`.

AWS CLI

Um die Ergebnisse einer Gesichtssuche abzurufen

Der folgende `get-face-search` Befehl zeigt die Ergebnisse einer Gesichtssuche an, die Sie zuvor mit einem Aufruf gestartet haben `start-face-search`.

```
aws rekognition get-face-search \  
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Ausgabe:

```
{  
  "Persons": [  
    {  
      "Timestamp": 467,  
      "FaceMatches": [],  
      "Person": {  
        "Index": 0,  
        "Face": {  
          "BoundingBox": {  
            "Width": 0.1560753583908081,  
            "Top": 0.13555361330509186,  
            "Left": -0.0952017530798912,  
            "Height": 0.6934483051300049  
          },  
          "Confidence": 0.9999999999999999  
        }  
      }  
    }  
  ]  
}
```

```
    "Landmarks": [
      {
        "Y": 0.4013825058937073,
        "X": -0.041750285774469376,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.41695496439933777,
        "X": 0.027979329228401184,
        "Type": "eyeRight"
      },
      {
        "Y": 0.6375303268432617,
        "X": -0.04034662991762161,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.6497718691825867,
        "X": 0.013960429467260838,
        "Type": "mouthRight"
      },
      {
        "Y": 0.5238034129142761,
        "X": 0.008022055961191654,
        "Type": "nose"
      }
    ],
    "Pose": {
      "Yaw": -58.07863998413086,
      "Roll": 1.9384294748306274,
      "Pitch": -24.66305160522461
    },
    "Quality": {
      "Sharpness": 83.14741516113281,
      "Brightness": 25.75942611694336
    },
    "Confidence": 87.7622299194336
  }
},
{
  "Timestamp": 967,
  "FaceMatches": [
    {
```

```
    "Face": {
      "BoundingBox": {
        "Width": 0.12368900328874588,
        "Top": 0.16007399559020996,
        "Left": 0.5901259779930115,
        "Height": 0.2514039874076843
      },
      "FaceId": "056a95fa-2060-4159-9cab-7ed4daa030fa",
      "ExternalImageId": "image3.jpg",
      "Confidence": 100.0,
      "ImageId": "08f8a078-8929-37fd-8e8f-aadf690e8232"
    },
    "Similarity": 98.44476318359375
  }
],
"Person": {
  "Index": 1,
  "Face": {
    "BoundingBox": {
      "Width": 0.28559377789497375,
      "Top": 0.19436298310756683,
      "Left": 0.024553587660193443,
      "Height": 0.7216082215309143
    },
    "Landmarks": [
      {
        "Y": 0.4650231599807739,
        "X": 0.16269078850746155,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.4843238294124603,
        "X": 0.2782580852508545,
        "Type": "eyeRight"
      },
      {
        "Y": 0.71530681848526,
        "X": 0.1741468608379364,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.7310671210289001,
        "X": 0.26857468485832214,
        "Type": "mouthRight"
      }
    ]
  }
}
```

```

        },
        {
            "Y": 0.582602322101593,
            "X": 0.2566150426864624,
            "Type": "nose"
        }
    ],
    "Pose": {
        "Yaw": 11.487052917480469,
        "Roll": 5.074230670928955,
        "Pitch": 15.396159172058105
    },
    "Quality": {
        "Sharpness": 73.32209777832031,
        "Brightness": 54.96497344970703
    },
    "Confidence": 99.99998474121094
}
}
}
],
"NextToken": "5bkgcezyuaqhtWk3C80TW6cjRghrwV9XDMivm5B3MXm+Lv6G+L+GejyFHPhoNa/
ldXIC4c/d",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
}
}
}

```

Weitere Informationen finden Sie unter [Suchen in gespeicherten Videos nach Gesichtern](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetFaceSearch](#) in der AWS CLI Befehlsreferenz.

get-label-detection

Das folgende Codebeispiel zeigt, wie man es benutzt `get-label-detection`.

AWS CLI

Um die Ergebnisse einer Objekt- und Szenenerkennung abzurufen

Mit dem folgenden `get-label-detection` Befehl werden die Ergebnisse einer Objekt- und Szenenerkennung angezeigt, die Sie zuvor mit einem Aufruf gestartet haben `start-label-detection`.

```
aws rekognition get-label-detection \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Ausgabe:

```
{
  "Labels": [
    {
      "Timestamp": 0,
      "Label": {
        "Instances": [],
        "Confidence": 50.19071578979492,
        "Parents": [
          {
            "Name": "Person"
          },
          {
            "Name": "Crowd"
          }
        ],
        "Name": "Audience"
      }
    },
    {
      "Timestamp": 0,
      "Label": {
        "Instances": [],
        "Confidence": 55.74115753173828,
        "Parents": [
          {
            "Name": "Room"
          },
          {
            "Name": "Indoors"
          }
        ],

```



```

        {
            "Name": "School"
        }
    ],
    "Name": "Classroom"
}
}
],
"JobStatus": "SUCCEEDED",
"LabelModelVersion": "2.0",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
},
"NextToken": "BMugzAi4L72IERzQdbpyMQuEFBsjl05W0Yx3mfG+sR9mm98E1/
Cp0benspRfs/5FBQFs4X7G"
}

```

Weitere Informationen finden Sie unter [Erkennen von Labels in einem Video](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetLabelDetection](#) in der AWS CLI Befehlsreferenz.

get-person-tracking

Das folgende Codebeispiel zeigt, wie man es benutzt `get-person-tracking`.

AWS CLI

Um die Ergebnisse einer Personenpfadsuche zu ermitteln

Mit dem folgenden `get-person-tracking` Befehl werden die Ergebnisse eines Personenpfadvorgangs angezeigt, den Sie zuvor durch einen Aufruf gestartet haben. `start-person-tracking`

```
aws rekognition get-person-tracking \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Ausgabe:

```
{
  "Persons": [
    {
      "Timestamp": 500,
      "Person": {
        "BoundingBox": {
          "Width": 0.4151041805744171,
          "Top": 0.07870370149612427,
          "Left": 0.0,
          "Height": 0.9212962985038757
        },
        "Index": 0
      }
    },
    {
      "Timestamp": 567,
      "Person": {
        "BoundingBox": {
          "Width": 0.4755208194255829,
          "Top": 0.07777778059244156,
          "Left": 0.0,
          "Height": 0.9194444417953491
        },
        "Index": 0
      }
    }
  ],
  "NextToken": "D/vRIYnyhG79ugdta3f+8cRg9oSRO
+HigG0uxRiYpTn0ExnqTi1CJektVAc4HrAXDv25eHYk",
  "JobStatus": "SUCCEEDED",
  "VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
  }
}
```

Weitere Informationen finden Sie unter [People Pathing](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [GetPersonTracking](#) in AWS CLI der Befehlsreferenz.

index-faces

Das folgende Codebeispiel zeigt, wie man es benutzt `index-faces`.

Weitere Informationen finden Sie unter [Hinzufügen von Gesichtern zu einer Sammlung](#).

AWS CLI

Um Gesichter zu einer Sammlung hinzuzufügen

Mit dem folgenden `index-faces` Befehl werden die in einem Bild gefundenen Gesichter zur angegebenen Sammlung hinzugefügt.

```
aws rekognition index-faces \  
  --image '{"S3Object":{"Bucket":"MyVideoS3Bucket","Name":"MyPicture.jpg"}}' \  
  --collection-id MyCollection \  
  --max-faces 1 \  
  --quality-filter "AUTO" \  
  --detection-attributes "ALL" \  
  --external-image-id "MyPicture.jpg"
```

Ausgabe:

```
{  
  "FaceRecords": [  
    {  
      "FaceDetail": {  
        "Confidence": 99.993408203125,  
        "Eyeglasses": {  
          "Confidence": 99.11750030517578,  
          "Value": false  
        },  
        "Sunglasses": {  
          "Confidence": 99.98249053955078,  
          "Value": false  
        },  
        "Gender": {  
          "Confidence": 99.92769622802734,  
          "Value": "Male"  
        },  
        "Landmarks": [  
          {  
            "Y": 0.26750367879867554,  
            "X": 0.6202793717384338,  

```

```
    "Type": "eyeLeft"
  },
  {
    "Y": 0.26642778515815735,
    "X": 0.6787431836128235,
    "Type": "eyeRight"
  },
  {
    "Y": 0.31361380219459534,
    "X": 0.6421601176261902,
    "Type": "nose"
  },
  {
    "Y": 0.3495299220085144,
    "X": 0.6216195225715637,
    "Type": "mouthLeft"
  },
  {
    "Y": 0.35194727778434753,
    "X": 0.669899046421051,
    "Type": "mouthRight"
  },
  {
    "Y": 0.26844894886016846,
    "X": 0.6210268139839172,
    "Type": "leftPupil"
  },
  {
    "Y": 0.26707562804222107,
    "X": 0.6817160844802856,
    "Type": "rightPupil"
  },
  {
    "Y": 0.24834522604942322,
    "X": 0.6018546223640442,
    "Type": "leftEyeBrowLeft"
  },
  {
    "Y": 0.24397172033786774,
    "X": 0.6172008514404297,
    "Type": "leftEyeBrowUp"
  },
  {
    "Y": 0.24677404761314392,
```

```
        "X": 0.6339119076728821,  
        "Type": "leftEyeBrowRight"  
    },  
    {  
        "Y": 0.24582654237747192,  
        "X": 0.6619398593902588,  
        "Type": "rightEyeBrowLeft"  
    },  
    {  
        "Y": 0.23973053693771362,  
        "X": 0.6804757118225098,  
        "Type": "rightEyeBrowUp"  
    },  
    {  
        "Y": 0.24441994726657867,  
        "X": 0.6978968977928162,  
        "Type": "rightEyeBrowRight"  
    },  
    {  
        "Y": 0.2695908546447754,  
        "X": 0.6085202693939209,  
        "Type": "leftEyeLeft"  
    },  
    {  
        "Y": 0.26716896891593933,  
        "X": 0.6315826177597046,  
        "Type": "leftEyeRight"  
    },  
    {  
        "Y": 0.26289820671081543,  
        "X": 0.6202316880226135,  
        "Type": "leftEyeUp"  
    },  
    {  
        "Y": 0.27123287320137024,  
        "X": 0.6205548048019409,  
        "Type": "leftEyeDown"  
    },  
    {  
        "Y": 0.2668408751487732,  
        "X": 0.6663622260093689,  
        "Type": "rightEyeLeft"  
    },  
    {
```

```
        "Y": 0.26741549372673035,
        "X": 0.6910083889961243,
        "Type": "rightEyeRight"
    },
    {
        "Y": 0.2614026665687561,
        "X": 0.6785826086997986,
        "Type": "rightEyeUp"
    },
    {
        "Y": 0.27075251936912537,
        "X": 0.6789616942405701,
        "Type": "rightEyeDown"
    },
    {
        "Y": 0.3211299479007721,
        "X": 0.6324167847633362,
        "Type": "noseLeft"
    },
    {
        "Y": 0.32276326417922974,
        "X": 0.6558475494384766,
        "Type": "noseRight"
    },
    {
        "Y": 0.34385165572166443,
        "X": 0.6444970965385437,
        "Type": "mouthUp"
    },
    {
        "Y": 0.3671635091304779,
        "X": 0.6459195017814636,
        "Type": "mouthDown"
    }
],
"Pose": {
    "Yaw": -9.54541015625,
    "Roll": -0.5709401965141296,
    "Pitch": 0.6045494675636292
},
"Emotions": [
    {
        "Confidence": 39.90074157714844,
        "Type": "HAPPY"
    }
]
```

```
    },
    {
      "Confidence": 23.38753890991211,
      "Type": "CALM"
    },
    {
      "Confidence": 5.840933322906494,
      "Type": "CONFUSED"
    }
  ],
  "AgeRange": {
    "High": 63,
    "Low": 45
  },
  "EyesOpen": {
    "Confidence": 99.80887603759766,
    "Value": true
  },
  "BoundingBox": {
    "Width": 0.18562500178813934,
    "Top": 0.1618015021085739,
    "Left": 0.5575000047683716,
    "Height": 0.24770642817020416
  },
  "Smile": {
    "Confidence": 99.69740295410156,
    "Value": false
  },
  "MouthOpen": {
    "Confidence": 99.97393798828125,
    "Value": false
  },
  "Quality": {
    "Sharpness": 95.54405975341797,
    "Brightness": 63.867706298828125
  },
  "Mustache": {
    "Confidence": 97.05007934570312,
    "Value": false
  },
  "Beard": {
    "Confidence": 87.34505462646484,
    "Value": false
  }
}
```

```
    },
    "Face": {
      "BoundingBox": {
        "Width": 0.18562500178813934,
        "Top": 0.1618015021085739,
        "Left": 0.5575000047683716,
        "Height": 0.24770642817020416
      },
      "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
      "ExternalImageId": "example-image.jpg",
      "Confidence": 99.993408203125,
      "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
    }
  ],
  "UnindexedFaces": [],
  "FaceModelVersion": "3.0",
  "OrientationCorrection": "ROTATE_0"
}
```

Weitere Informationen finden Sie unter [Gesichter zu einer Sammlung hinzufügen](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [IndexFaces](#) in der AWS CLI Befehlsreferenz.

list-collections

Das folgende Codebeispiel zeigt, wie man es benutzt `list-collections`.

Weitere Informationen finden Sie unter [Sammlungen auflisten](#).

AWS CLI

Um die verfügbaren Sammlungen aufzulisten

Der folgende `list-collections` Befehl listet die verfügbaren Sammlungen im AWS Konto auf.

```
aws rekognition list-collections
```

Ausgabe:

```
{
  "FaceModelVersions": [
```



```
    "2.0",
    "3.0",
    "3.0",
    "3.0",
    "4.0",
    "1.0",
    "3.0",
    "4.0",
    "4.0",
    "4.0"
  ],
  "CollectionIds": [
    "MyCollection1",
    "MyCollection2",
    "MyCollection3",
    "MyCollection4",
    "MyCollection5",
    "MyCollection6",
    "MyCollection7",
    "MyCollection8",
    "MyCollection9",
    "MyCollection10"
  ]
}
```

Weitere Informationen finden Sie unter [Sammlungen auflisten](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [ListCollections](#) in der AWS CLI Befehlsreferenz.

list-faces

Das folgende Codebeispiel zeigt, wie man es benutzt `list-faces`.

Weitere Informationen finden Sie unter [Gesichter in einer Sammlung auflisten](#).

AWS CLI

Um die Gesichter in einer Sammlung aufzulisten

Der folgende `list-faces` Befehl listet die Gesichter in der angegebenen Sammlung auf.

```
aws rekognition list-faces \
```

```
--collection-id MyCollection
```

Ausgabe:

```
{
  "FaceModelVersion": "3.0",
  "Faces": [
    {
      "BoundingBox": {
        "Width": 0.5216310024261475,
        "Top": 0.3256250023841858,
        "Left": 0.13394300639629364,
        "Height": 0.3918749988079071
      },
      "FaceId": "0040279c-0178-436e-b70a-e61b074e96b0",
      "ExternalImageId": "image1.jpg",
      "Confidence": 100.0,
      "ImageId": "f976e487-3719-5e2d-be8b-ea2724c26991"
    },
    {
      "BoundingBox": {
        "Width": 0.5074880123138428,
        "Top": 0.3774999976158142,
        "Left": 0.18302799761295319,
        "Height": 0.3812499940395355
      },
      "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
      "ExternalImageId": "image2.jpg",
      "Confidence": 99.99930572509766,
      "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
    },
    {
      "BoundingBox": {
        "Width": 0.5574039816856384,
        "Top": 0.37187498807907104,
        "Left": 0.14559100568294525,
        "Height": 0.4181250035762787
      },
      "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
      "ExternalImageId": "image3.jpg",
      "Confidence": 99.99960327148438,
      "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
    }
  ],
}
```

```
{
  "BoundingBox": {
    "Width": 0.18562500178813934,
    "Top": 0.1618019938468933,
    "Left": 0.5575000047683716,
    "Height": 0.24770599603652954
  },
  "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
  "ExternalImageId": "image4.jpg",
  "Confidence": 99.99340057373047,
  "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
},
{
  "BoundingBox": {
    "Width": 0.5307819843292236,
    "Top": 0.2862499952316284,
    "Left": 0.1564060002565384,
    "Height": 0.3987500071525574
  },
  "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
  "ExternalImageId": "image5.jpg",
  "Confidence": 99.99970245361328,
  "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
},
{
  "BoundingBox": {
    "Width": 0.5773710012435913,
    "Top": 0.34437501430511475,
    "Left": 0.12396000325679779,
    "Height": 0.4337500035762787
  },
  "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
  "ExternalImageId": "image6.jpg",
  "Confidence": 100.0,
  "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
},
{
  "BoundingBox": {
    "Width": 0.5349419713020325,
    "Top": 0.29124999046325684,
    "Left": 0.16389399766921997,
    "Height": 0.40187498927116394
  },
  "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
```

```
    "ExternalImageId": "image7.jpg",
    "Confidence": 99.99979400634766,
    "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
  },
  {
    "BoundingBox": {
      "Width": 0.41499999165534973,
      "Top": 0.09187500178813934,
      "Left": 0.28083300590515137,
      "Height": 0.3112500011920929
    },
    "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
    "ExternalImageId": "image8.jpg",
    "Confidence": 99.99769592285156,
    "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
  },
  {
    "BoundingBox": {
      "Width": 0.48166701197624207,
      "Top": 0.20999999344348907,
      "Left": 0.21250000596046448,
      "Height": 0.36125001311302185
    },
    "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
    "ExternalImageId": "image9.jpg",
    "Confidence": 99.99949645996094,
    "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
  },
  {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
    "ExternalImageId": "image10.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
  }
]
}
```

Weitere Informationen finden Sie unter [Gesichter in einer Sammlung auflisten](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [ListFaces](#) in der AWS CLI Befehlsreferenz.

list-stream-processors

Das folgende Codebeispiel zeigt, wie man es benutzt `list-stream-processors`.

AWS CLI

Um die Stream-Prozessoren in Ihrem Konto aufzulisten

Der folgende `list-stream-processors` Befehl listet die Stream-Prozessoren in Ihrem Konto und deren Status auf.

```
aws rekognition list-stream-processors
```

Ausgabe:

```
{
  "StreamProcessors": [
    {
      "Status": "STOPPED",
      "Name": "my-stream-processor"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Streaming-Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [ListStreamProcessors](#) in der AWS CLI Befehlsreferenz.

recognize-celebrities

Das folgende Codebeispiel zeigt, wie man es benutzt `recognize-celebrities`.

Weitere Informationen finden Sie unter [Erkennen von Prominenten in einem Bild](#).

AWS CLI

Um Prominente in einem Bild zu erkennen

Der folgende `recognize-celebrities` Befehl erkennt Prominente in dem angegebenen Bild, das in einem Amazon S3 S3-Bucket gespeichert ist. :

```
aws rekognition recognize-celebrities \  
  --image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

Ausgabe:

```
{  
  "UnrecognizedFaces": [  
    {  
      "BoundingBox": {  
        "Width": 0.14416666328907013,  
        "Top": 0.077777778059244156,  
        "Left": 0.625,  
        "Height": 0.2746031880378723  
      },  
      "Confidence": 99.9990234375,  
      "Pose": {  
        "Yaw": 10.80408763885498,  
        "Roll": -12.761146545410156,  
        "Pitch": 10.96889877319336  
      },  
      "Quality": {  
        "Sharpness": 94.1185531616211,  
        "Brightness": 79.18367004394531  
      },  
      "Landmarks": [  
        {  
          "Y": 0.18220913410186768,  
          "X": 0.6702951788902283,  
          "Type": "eyeLeft"  
        },  
        {  
          "Y": 0.16337193548679352,  
          "X": 0.7188183665275574,  
          "Type": "eyeRight"  
        },  
        {  
          "Y": 0.20739148557186127,  
          "X": 0.7055801749229431,  
          "Type": "nose"  
        }  
      ]  
    }  
  ]  
}
```

```
        {
          "Y": 0.2889308035373688,
          "X": 0.687512218952179,
          "Type": "mouthLeft"
        },
        {
          "Y": 0.2706988751888275,
          "X": 0.7250053286552429,
          "Type": "mouthRight"
        }
      ]
    },
  ],
  "CelebrityFaces": [
    {
      "MatchConfidence": 100.0,
      "Face": {
        "BoundingBox": {
          "Width": 0.14000000059604645,
          "Top": 0.1190476194024086,
          "Left": 0.82833331823349,
          "Height": 0.2666666805744171
        },
        "Confidence": 99.99359130859375,
        "Pose": {
          "Yaw": -10.509642601013184,
          "Roll": -14.51749324798584,
          "Pitch": 13.799399375915527
        },
        "Quality": {
          "Sharpness": 78.74752044677734,
          "Brightness": 42.201324462890625
        },
        "Landmarks": [
          {
            "Y": 0.2290833294391632,
            "X": 0.8709492087364197,
            "Type": "eyeLeft"
          },
          {
            "Y": 0.20639978349208832,
            "X": 0.9153988361358643,
            "Type": "eyeRight"
          }
        ]
      }
    }
  ]
}
```

```
        {
            "Y": 0.25417643785476685,
            "X": 0.8907724022865295,
            "Type": "nose"
        },
        {
            "Y": 0.32729196548461914,
            "X": 0.8876466155052185,
            "Type": "mouthLeft"
        },
        {
            "Y": 0.3115464746952057,
            "X": 0.9238573312759399,
            "Type": "mouthRight"
        }
    ]
},
>Name": "Celeb A",
>Urls": [
    "www.imdb.com/name/aaaaaaaaa"
],
>Id": "1111111"
},
{
    "MatchConfidence": 97.0,
    "Face": {
        "BoundingBox": {
            "Width": 0.13333334028720856,
            "Top": 0.24920634925365448,
            "Left": 0.4449999928474426,
            "Height": 0.2539682686328888
        },
        "Confidence": 99.99979400634766,
        "Pose": {
            "Yaw": 6.557040691375732,
            "Roll": -7.316643714904785,
            "Pitch": 9.272967338562012
        },
        "Quality": {
            "Sharpness": 83.23492431640625,
            "Brightness": 78.83267974853516
        },
        "Landmarks": [
            {
```



```
        "Y": 0.3625510632991791,
        "X": 0.48898839950561523,
        "Type": "eyeLeft"
    },
    {
        "Y": 0.35366007685661316,
        "X": 0.5313721299171448,
        "Type": "eyeRight"
    },
    {
        "Y": 0.3894785940647125,
        "X": 0.5173314809799194,
        "Type": "nose"
    },
    {
        "Y": 0.44889405369758606,
        "X": 0.5020005702972412,
        "Type": "mouthLeft"
    },
    {
        "Y": 0.4408611059188843,
        "X": 0.5351271629333496,
        "Type": "mouthRight"
    }
]
},
"Name": "Celeb B",
"Urls": [
    "www.imdb.com/name/bbbbbbbbbb"
],
"Id": "2222222"
},
{
    "MatchConfidence": 100.0,
    "Face": {
        "BoundingBox": {
            "Width": 0.12416666746139526,
            "Top": 0.2968254089355469,
            "Left": 0.2150000035762787,
            "Height": 0.23650793731212616
        },
        "Confidence": 99.99958801269531,
        "Pose": {
            "Yaw": 7.801797866821289,
```

```
        "Roll": -8.326810836791992,  
        "Pitch": 7.844768047332764  
    },  
    "Quality": {  
        "Sharpness": 86.93206024169922,  
        "Brightness": 79.81291198730469  
    },  
    "Landmarks": [  
        {  
            "Y": 0.4027804136276245,  
            "X": 0.2575301229953766,  
            "Type": "eyeLeft"  
        },  
        {  
            "Y": 0.3934555947780609,  
            "X": 0.2956969439983368,  
            "Type": "eyeRight"  
        },  
        {  
            "Y": 0.4309830069541931,  
            "X": 0.2837020754814148,  
            "Type": "nose"  
        },  
        {  
            "Y": 0.48186683654785156,  
            "X": 0.26812544465065,  
            "Type": "mouthLeft"  
        },  
        {  
            "Y": 0.47338807582855225,  
            "X": 0.29905644059181213,  
            "Type": "mouthRight"  
        }  
    ]  
},  
"Name": "Celeb C",  
"Urls": [  
    "www.imdb.com/name/ccccccccc"  
],  
"Id": "3333333"  
},  
{  
    "MatchConfidence": 97.0,  
    "Face": {
```

```
"BoundingBox": {
  "Width": 0.11916666477918625,
  "Top": 0.3698412775993347,
  "Left": 0.008333333767950535,
  "Height": 0.22698412835597992
},
"Confidence": 99.99999237060547,
"Pose": {
  "Yaw": 16.38478660583496,
  "Roll": -1.0260354280471802,
  "Pitch": 5.975185394287109
},
"Quality": {
  "Sharpness": 83.23492431640625,
  "Brightness": 61.408443450927734
},
"Landmarks": [
  {
    "Y": 0.4632347822189331,
    "X": 0.049406956881284714,
    "Type": "eyeLeft"
  },
  {
    "Y": 0.46388113498687744,
    "X": 0.08722897619009018,
    "Type": "eyeRight"
  },
  {
    "Y": 0.5020678639411926,
    "X": 0.0758260041475296,
    "Type": "nose"
  },
  {
    "Y": 0.544157862663269,
    "X": 0.054029736667871475,
    "Type": "mouthLeft"
  },
  {
    "Y": 0.5463630557060242,
    "X": 0.08464983850717545,
    "Type": "mouthRight"
  }
]
},
```

```

        "Name": "Celeb D",
        "Urls": [
            "www.imdb.com/name/ddddddddd"
        ],
        "Id": "44444444"
    }
]
}

```

Weitere Informationen finden Sie unter [Erkennen von Prominenten in einem Bild](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [RecognizeCelebrities](#) in der AWS CLI Befehlsreferenz.

search-faces-by-image

Das folgende Codebeispiel zeigt, wie man es benutzt `search-faces-by-image`.

Weitere Informationen finden Sie unter [Nach einem Gesicht suchen \(Bild\)](#).

AWS CLI

Um in einer Sammlung nach Gesichtern zu suchen, die dem größten Gesicht in einem Bild entsprechen.

Mit dem folgenden `search-faces-by-image` Befehl wird in einer Sammlung nach Gesichtern gesucht, die dem größten Gesicht im angegebenen Bild entsprechen. :

```

aws rekognition search-faces-by-image \
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePerson.jpg"}}' \
  --collection-id MyFaceImageCollection

{
  "SearchedFaceBoundingBox": {
    "Width": 0.18562500178813934,
    "Top": 0.1618015021085739,
    "Left": 0.5575000047683716,
    "Height": 0.24770642817020416
  },
  "SearchedFaceConfidence": 99.993408203125,
  "FaceMatches": [
    {
      "Face": {

```

```
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
    "ExternalImageId": "example-image.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
  },
  "Similarity": 99.97913360595703
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
    "ExternalImageId": "image3.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
  },
  "Similarity": 99.97913360595703
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.41499999165534973,
      "Top": 0.09187500178813934,
      "Left": 0.28083300590515137,
      "Height": 0.3112500011920929
    },
    "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
    "ExternalImageId": "image2.jpg",
    "Confidence": 99.99769592285156,
    "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
  },
  "Similarity": 99.18069458007812
},
{
```

```
"Face": {
  "BoundingBox": {
    "Width": 0.48166701197624207,
    "Top": 0.20999999344348907,
    "Left": 0.21250000596046448,
    "Height": 0.36125001311302185
  },
  "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
  "ExternalImageId": "image1.jpg",
  "Confidence": 99.99949645996094,
  "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
},
"Similarity": 98.66607666015625
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5349419713020325,
      "Top": 0.29124999046325684,
      "Left": 0.16389399766921997,
      "Height": 0.40187498927116394
    },
    "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
    "ExternalImageId": "image9.jpg",
    "Confidence": 99.99979400634766,
    "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
  },
  "Similarity": 98.24278259277344
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5307819843292236,
      "Top": 0.2862499952316284,
      "Left": 0.1564060002565384,
      "Height": 0.3987500071525574
    },
    "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
    "ExternalImageId": "image10.jpg",
    "Confidence": 99.99970245361328,
    "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
  },
  "Similarity": 98.10665893554688
},
}
```

```
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5074880123138428,
      "Top": 0.3774999976158142,
      "Left": 0.18302799761295319,
      "Height": 0.3812499940395355
    },
    "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
    "ExternalImageId": "image6.jpg",
    "Confidence": 99.99930572509766,
    "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
  },
  "Similarity": 98.10526275634766
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5574039816856384,
      "Top": 0.37187498807907104,
      "Left": 0.14559100568294525,
      "Height": 0.4181250035762787
    },
    "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
    "ExternalImageId": "image5.jpg",
    "Confidence": 99.99960327148438,
    "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
  },
  "Similarity": 97.94659423828125
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5773710012435913,
      "Top": 0.34437501430511475,
      "Left": 0.12396000325679779,
      "Height": 0.4337500035762787
    },
    "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
    "ExternalImageId": "image8.jpg",
    "Confidence": 100.0,
    "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
  },
  "Similarity": 97.93476867675781
}
```

```

    ],
    "FaceModelVersion": "3.0"
}

```

Weitere Informationen finden Sie unter [Mit einem Bild nach einem Gesicht suchen](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [SearchFacesByImage](#) in der AWS CLI Befehlsreferenz.

search-faces

Das folgende Codebeispiel zeigt, wie man es benutzt `search-faces`.

Weitere Informationen finden Sie unter [Nach einem Gesicht suchen \(Gesichts-ID\)](#).

AWS CLI

Um in einer Sammlung nach Gesichtern zu suchen, die einer Gesichts-ID entsprechen.

Mit dem folgenden `search-faces` Befehl wird in einer Sammlung nach Gesichtern gesucht, die der angegebenen Gesichts-ID entsprechen.

```

aws rekognition search-faces \
  --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \
  --collection-id MyCollection

```

Ausgabe:

```

{
  "SearchedFaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
  "FaceModelVersion": "3.0",
  "FaceMatches": [
    {
      "Face": {
        "BoundingBox": {
          "Width": 0.48166701197624207,
          "Top": 0.20999999344348907,
          "Left": 0.21250000596046448,
          "Height": 0.36125001311302185
        },
        "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",

```



```
        "ExternalImageId": "image1.jpg",
        "Confidence": 99.99949645996094,
        "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
    },
    "Similarity": 99.30997467041016
},
{
    "Face": {
        "BoundingBox": {
            "Width": 0.18562500178813934,
            "Top": 0.1618019938468933,
            "Left": 0.5575000047683716,
            "Height": 0.24770599603652954
        },
        "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
        "ExternalImageId": "example-image.jpg",
        "Confidence": 99.99340057373047,
        "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
    },
    "Similarity": 99.24862670898438
},
{
    "Face": {
        "BoundingBox": {
            "Width": 0.18562500178813934,
            "Top": 0.1618019938468933,
            "Left": 0.5575000047683716,
            "Height": 0.24770599603652954
        },
        "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
        "ExternalImageId": "image3.jpg",
        "Confidence": 99.99340057373047,
        "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
    },
    "Similarity": 99.24862670898438
},
{
    "Face": {
        "BoundingBox": {
            "Width": 0.5349419713020325,
            "Top": 0.29124999046325684,
            "Left": 0.16389399766921997,
            "Height": 0.40187498927116394
        },
```

```
    "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
    "ExternalImageId": "image9.jpg",
    "Confidence": 99.99979400634766,
    "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
  },
  "Similarity": 96.73158264160156
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5307819843292236,
      "Top": 0.2862499952316284,
      "Left": 0.1564060002565384,
      "Height": 0.3987500071525574
    },
    "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
    "ExternalImageId": "image10.jpg",
    "Confidence": 99.99970245361328,
    "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
  },
  "Similarity": 96.48291015625
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5074880123138428,
      "Top": 0.3774999976158142,
      "Left": 0.18302799761295319,
      "Height": 0.3812499940395355
    },
    "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
    "ExternalImageId": "image6.jpg",
    "Confidence": 99.99930572509766,
    "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
  },
  "Similarity": 96.43287658691406
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5574039816856384,
      "Top": 0.37187498807907104,
      "Left": 0.14559100568294525,
      "Height": 0.4181250035762787
```

```

    },
    "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
    "ExternalImageId": "image5.jpg",
    "Confidence": 99.99960327148438,
    "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
  },
  "Similarity": 95.25305938720703
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5773710012435913,
      "Top": 0.34437501430511475,
      "Left": 0.12396000325679779,
      "Height": 0.4337500035762787
    },
    "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
    "ExternalImageId": "image8.jpg",
    "Confidence": 100.0,
    "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
  },
  "Similarity": 95.22837829589844
}
]
}

```

Weitere Informationen finden Sie unter [Suchen nach einem Gesicht anhand seiner Gesichts-ID](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [SearchFaces](#) in der AWS CLI Befehlsreferenz.

start-celebrity-recognition

Das folgende Codebeispiel zeigt, wie man es benutzt `start-celebrity-recognition`.

AWS CLI

Um die Anerkennung von Prominenten in einem gespeicherten Video zu starten

Der folgende `start-celebrity-recognition` Befehl startet einen Job zur Suche nach Prominenten in der angegebenen Videodatei, die in einem Amazon S3 S3-Bucket gespeichert ist.

```
aws rekognition start-celebrity-recognition \
```

```
--video "S3object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Ausgabe:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Weitere Informationen finden Sie unter [Erkennen von Prominenten in einem gespeicherten Video](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartCelebrityRecognition](#) in der AWS CLI Befehlsreferenz.

start-content-moderation

Das folgende Codebeispiel zeigt, wie man es benutzt `start-content-moderation`.

AWS CLI

Um die Erkennung unsicherer Inhalte in einem gespeicherten Video zu starten

Der folgende `start-content-moderation` Befehl startet einen Job zur Erkennung unsicherer Inhalte in der angegebenen Videodatei, die in einem Amazon S3 S3-Bucket gespeichert ist.

```
aws rekognition start-content-moderation \  
  --video "S3object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Ausgabe:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Weitere Informationen finden Sie unter [Erkennen unsicherer gespeicherter Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartContentModeration](#) in AWS CLI der Befehlsreferenz.

start-face-detection

Das folgende Codebeispiel zeigt die Verwendung `start-face-detection`.

AWS CLI

Um Gesichter in einem Video zu erkennen

Der folgende `start-face-detection` Befehl startet einen Job zur Erkennung von Gesichtern in der angegebenen Videodatei, die in einem Amazon S3 S3-Bucket gespeichert ist.

```
aws rekognition start-face-detection
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Ausgabe:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Weitere Informationen finden Sie unter [Erkennen von Gesichtern in einem gespeicherten Video](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartFaceDetection](#) in der AWS CLI Befehlsreferenz.

start-face-search

Das folgende Codebeispiel zeigt die Verwendung `start-face-search`.

AWS CLI

Um in einer Sammlung nach Gesichtern zu suchen, die mit Gesichtern übereinstimmen, die in einem Video erkannt wurden

Der folgende `start-face-search` Befehl startet einen Job zur Suche nach Gesichtern in einer Sammlung, die mit Gesichtern übereinstimmen, die in der angegebenen Videodatei in einem Amazon S3 S3-Bucket erkannt wurden.

```
aws rekognition start-face-search \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
  --collection-id collection
```

Ausgabe:

```
{
```

```
"JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Weitere Informationen finden Sie unter [Suchen in gespeicherten Videos nach Gesichtern](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartFaceSearch](#) in der AWS CLI Befehlsreferenz.

start-label-detection

Das folgende Codebeispiel zeigt die Verwendung `start-label-detection`.

AWS CLI

Um Objekte und Szenen in einem Video zu erkennen

Der folgende `start-label-detection` Befehl startet einen Job zur Erkennung von Objekten und Szenen in der angegebenen Videodatei, die in einem Amazon S3 S3-Bucket gespeichert ist.

```
aws rekognition start-label-detection \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Ausgabe:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Weitere Informationen finden Sie unter [Erkennen von Labels in einem Video](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartLabelDetection](#) in der AWS CLI Befehlsreferenz.

start-person-tracking

Das folgende Codebeispiel zeigt die Verwendung `start-person-tracking`.

AWS CLI

Um die Pfadsuche von Personen in einem gespeicherten Video zu starten

Mit dem folgenden `start-person-tracking` Befehl wird ein Job gestartet, um die Pfade zu verfolgen, die Benutzer in der angegebenen Videodatei, die in einem Amazon S3 S3-Bucket gespeichert ist, zurücklegen. :

```
aws rekognition start-person-tracking \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Ausgabe:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Weitere Informationen finden Sie unter [People Pathing](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartPersonTracking](#) in AWS CLI der Befehlsreferenz.

start-stream-processor

Das folgende Codebeispiel zeigt die Verwendung `start-stream-processor`.

AWS CLI

Um einen Stream-Prozessor zu starten

Mit dem folgenden `start-stream-processor` Befehl wird der angegebene Videostream-Prozessor gestartet.

```
aws rekognition start-stream-processor \  
  --name my-stream-processor
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Streaming-Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StartStreamProcessor](#) in der AWS CLI Befehlsreferenz.

stop-stream-processor

Das folgende Codebeispiel zeigt die Verwendung `stop-stream-processor`.

AWS CLI

Um einen laufenden Stream-Prozessor zu stoppen

Der folgende `stop-stream-processor` Befehl stoppt den angegebenen laufenden Stream-Prozessor.

```
aws rekognition stop-stream-processor \  
  --name my-stream-processor
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Streaming-Videos](#) im Amazon Rekognition Developer Guide.

- API-Einheiten finden Sie [StopStreamProcessor](#) in der AWS CLI Befehlsreferenz.

AWS RAM Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS RAM.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-resource-share-invitation

Das folgende Codebeispiel zeigt die Verwendung `accept-resource-share-invitation`.

AWS CLI

Um eine Einladung zur gemeinsamen Nutzung von Ressourcen anzunehmen

Im folgenden `accept-resource-share-invitation` Beispiel wird die angegebene Einladung zur gemeinsamen Nutzung einer Ressource akzeptiert. Principals im eingeladenen Konto können sofort damit beginnen, die Ressourcen in der Freigabe zu nutzen.

```
aws ram accept-resource-share-invitation \
  --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-
  share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

Ausgabe:

```
{
  "resourceShareInvitation": {
    "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111111111111:resource-
  share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE",
    "resourceShareName": "MyLicenseShare",
    "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-
  share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
    "senderAccountId": "111111111111",
    "receiverAccountId": "222222222222",
    "invitationTimestamp": "2021-09-22T15:07:35.620000-07:00",
    "status": "ACCEPTED"
  }
}
```

- API-Einheiten finden Sie [AcceptResourceShareInvitation](#) in der AWS CLI Befehlsreferenz.

associate-resource-share-permission

Das folgende Codebeispiel zeigt die Verwendung `associate-resource-share-permission`.

AWS CLI

Um eine RAM verwaltete Berechtigung einer Ressourcenfreigabe zuzuordnen

Im folgenden `associate-resource-share-permission` Beispiel wird die vorhandene verwaltete Berechtigung für den entsprechenden Ressourcentyp durch die angegebene verwaltete Berechtigung ersetzt. Der Zugriff auf alle Ressourcen des entsprechenden Ressourcentyps wird durch die neue Berechtigung geregelt.

```
aws ram associate-resource-share-permission \
```

```

--permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite \
--replace \
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE

```

Ausgabe:

```

{
  "returnValue": true
}

```

- API-Einheiten finden Sie [AssociateResourceSharePermission](#) in der AWS CLI Befehlsreferenz.

associate-resource-share

Das folgende Codebeispiel zeigt die Verwendung `associate-resource-share`.

AWS CLI

Beispiel 1: Um eine Ressource einer Ressourcenfreigabe zuzuordnen

Das folgende `associate-resource-share` Beispiel fügt der angegebenen Ressourcenfreigabe eine Lizenzkonfiguration hinzu.

```

aws ram associate-resource-share \
--resource-share arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
--resource-arns arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE

```

Ausgabe:

```

{
  "resourceShareAssociations": [
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
      "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
      "associationType": "RESOURCE",
    }
  ]
}

```

```

        "status": "ASSOCIATING",
        "external": false
    }
]
}

```

Beispiel 2: Um einen Prinzipal einer Ressourcenfreigabe zuzuordnen

Das folgende `associate-resource-share` Beispiel gewährt allen Konten in der angegebenen Organisationseinheit Zugriff auf die angegebene Ressourcenfreigabe.

```

aws ram associate-resource-share \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
  --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE

```

Ausgabe:

```

{
  "resourceShareAssociations": [
    {
      "status": "ASSOCIATING",
      "associationType": "PRINCIPAL",
      "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
      "external": false,
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE"
    }
  ]
}

```

- API-Einheiten finden Sie [AssociateResourceShare](#) unter AWS CLI Befehlsreferenz.

create-resource-share

Das folgende Codebeispiel zeigt die Verwendung `create-resource-share`.

AWS CLI

Beispiel 1: Um eine Ressourcenfreigabe zu erstellen

Im folgenden `create-resource-share` Beispiel wird eine leere Ressourcenfreigabe mit dem angegebenen Namen erstellt. Sie müssen der Freigabe Ressourcen, Prinzipale und Berechtigungen separat hinzufügen.

```
aws ram create-resource-share \  
  --name MyNewResourceShare
```

Ausgabe:

```
{  
  "resourceShare": {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/4476c27d-8feb-4b21-afe9-7de23EXAMPLE",  
    "name": "MyNewResourceShare",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": true,  
    "status": "ACTIVE",  
    "creationTime": 1634586271.302,  
    "lastUpdatedTime": 1634586271.302  
  }  
}
```

Beispiel 2: So erstellen Sie eine Ressourcenfreigabe mit AWS Konten als Principals

Das folgende `create-resource-share` Beispiel erstellt eine Ressourcenfreigabe und gewährt Zugriff auf das angegebene AWS Konto (222222222222). Wenn die angegebenen Principals nicht Teil derselben AWS Organisation sind, werden Einladungen gesendet und müssen akzeptiert werden, bevor der Zugriff gewährt wird.

```
aws ram create-resource-share \  
  --name MyNewResourceShare \  
  --principals 222222222222
```

Beispiel 3: So erstellen Sie eine Ressourcenfreigabe, die auf Ihre AWS Organisation beschränkt ist

Das folgende `create-resource-share` Beispiel erstellt eine Ressourcenfreigabe, die auf Konten in der AWS Organisation beschränkt ist, der Ihr Konto angehört, und fügt die angegebene Organisationseinheit als Hauptbenutzer hinzu. Alle Konten in dieser Organisationseinheit können die Ressourcen in der Ressourcenfreigabe verwenden.

```
aws ram create-resource-share \  
  --name MyNewResourceShare \  
  --no-allow-external-principals \  
  --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-  
rEXAMPLE
```

Ausgabe:

```
{  
  "resourceShare": {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",  
    "name": "MyNewResourceShare",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": false,  
    "status": "ACTIVE",  
    "creationTime": 1634587042.49,  
    "lastUpdatedTime": 1634587042.49  
  }  
}
```

- API-Einheiten finden Sie [CreateResourceShare](#) in der AWS CLI Befehlsreferenz.

delete-resource-share

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-share`.

AWS CLI

Um eine Ressourcenfreigabe zu löschen

Im folgenden `delete-resource-share` Beispiel wird die angegebene Ressourcenfreigabe gelöscht.

```
aws ram delete-resource-share \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Die folgende Ausgabe weist auf Erfolg hin:

```
{
```

```
"returnValue": true
}
```

- API-Einheiten finden Sie [DeleteResourceShare](#) in der AWS CLI Befehlsreferenz.

disassociate-resource-share-permission

Das folgende Codebeispiel zeigt die Verwendung `disassociate-resource-share-permission`.

AWS CLI

Um eine RAM verwaltete Berechtigung für einen Ressourcentyp aus einer Ressourcenfreigabe zu entfernen

Im folgenden `disassociate-resource-share-permission` Beispiel wird die RAM verwaltete Berechtigung für Glue-Datenbanken aus der angegebenen Ressourcenfreigabe entfernt.

```
aws ram disassociate-resource-share-permission \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
  --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite
```

Ausgabe:

```
{
  "returnValue": true
}
```

- API-Einheiten finden Sie [DisassociateResourceSharePermission](#) in der AWS CLI Befehlsreferenz.

disassociate-resource-share

Das folgende Codebeispiel zeigt die Verwendung `disassociate-resource-share`.

AWS CLI

Um eine Ressource aus einer Ressourcenfreigabe zu entfernen

Im folgenden `disassociate-resource-share` Beispiel wird die angegebene Ressource, in diesem Fall ein VPC Subnetz, aus der angegebenen Ressourcenfreigabe entfernt. Alle Principals mit Zugriff auf die Ressourcenfreigabe können keine Operationen mehr mit dieser Ressource ausführen.

```
aws ram disassociate-resource-share \  
  --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0250c25a1fEXAMPLE \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Ausgabe:

```
{  
  "resourceShareAssociations": [  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",  
    "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0250c25a1fEXAMPLE",  
    "associationType": "RESOURCE",  
    "status": "DISASSOCIATING",  
    "external": false  
  ]  
}
```

- API-Einheiten finden Sie [DisassociateResourceShare](#) in der AWS CLI Befehlsreferenz.

enable-sharing-with-aws-organization

Das folgende Codebeispiel zeigt die Verwendung `enable-sharing-with-aws-organization`.

AWS CLI

Um die gemeinsame Nutzung von Ressourcen zwischen AWS Organizations zu ermöglichen

Das folgende `enable-sharing-with-aws-organization` Beispiel ermöglicht die gemeinsame Nutzung von Ressourcen in Ihrer Organisation und Ihren Organisationseinheiten.

```
aws ram enable-sharing-with-aws-organization
```

Die folgende Ausgabe zeigt den Erfolg an.

```
{
  "returnValue": true
}
```

- API-Einheiten finden Sie [EnableSharingWithAwsOrganization](#) in der AWS CLI Befehlsreferenz.

get-permission

Das folgende Codebeispiel zeigt die Verwendung `get-permission`.

AWS CLI

Um die Details für eine RAM verwaltete Berechtigung abzurufen

Im folgenden `get-permission` Beispiel werden die Details für die Standardversion der angegebenen RAM verwalteten Berechtigung angezeigt.

```
aws ram get-permission \
  --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase
```

Ausgabe:

```
{
  "permission": {
    "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
    "version": "2",
    "defaultVersion": true,
    "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
    "resourceType": "glue:Database",
    "permission": [{"Effect": "Allow", "Action": ["glue:GetTable",
"glue:UpdateTable", "glue>DeleteTable", "glue:BatchDeleteTable",
"glue:BatchDeleteTableVersion", "glue:GetTableVersion", "glue:GetTableVersions",
"glue:GetPartition", "glue:GetPartitions", "glue:BatchGetPartition",
"glue:BatchCreatePartition", "glue>CreatePartition", "glue:UpdatePartition",
"glue:BatchDeletePartition", "glue>DeletePartition", "glue:GetTables",
"glue:SearchTables"]}],
    "creationTime": 1624912434.431,
    "lastUpdatedTime": 1624912434.431,
    "isResourceTypeDefault": false
  }
}
```



```
}
}
```

- API-Einheiten finden Sie [GetPermission](#) unter AWS CLI Befehlsreferenz.

get-resource-policies

Das folgende Codebeispiel zeigt die Verwendung `get-resource-policies`.

AWS CLI

Um die Richtlinien für eine Ressource abzurufen

Im folgenden `get-resource-policies` Beispiel werden die ressourcenbasierten Berechtigungsrichtlinien für die angegebene Ressource angezeigt, die einer Ressourcenfreigabe zugeordnet ist.

```
aws ram get-resource-policies \
  --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE
```

Ausgabe:

```
{
  "policies": [
    [{"Version": "2008-10-17", "Statement": [{"Sid": "RamStatement1",
      "Effect": "Allow", "Principal": {"AWS": []}, "Action": ["ec2:RunInstances",
      "ec2:CreateNetworkInterface", "ec2:DescribeSubnets"], "Resource":
      "arn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1fEXAMPLE"}]}]
}
```

- API-Einheiten finden Sie unter [GetResourcePolicies AWS CLI](#) Befehlsreferenz.

get-resource-share-associations

Das folgende Codebeispiel zeigt die Verwendung `get-resource-share-associations`.

AWS CLI

Beispiel 1: Um alle Ressourcenzuordnungen für alle Ressourcentypen aufzulisten

Im folgenden `get-resource-share-associations` Beispiel werden die Ressourcenzuordnungen für alle Ressourcentypen in all Ihren Ressourcenfreigaben aufgeführt.

```
aws ram get-resource-share-associations \
  --association-type RESOURCE
```

Ausgabe:

```
{
  "resourceShareAssociations": [
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE",
      "resourceShareName": "MySubnetShare",
      "associationType": "RESOURCE",
      "status": "ASSOCIATED",
      "creationTime": 1565303590.973,
      "lastUpdatedTime": 1565303591.695,
      "external": false
    },
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/8167bdfe-4480-4a01-8632-315e0EXAMPLE",
      "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
      "resourceShareName": "MyLicenseShare",
      "associationType": "RESOURCE",
      "status": "ASSOCIATED",
      "creationTime": 1632342958.457,
      "lastUpdatedTime": 1632342958.907,
      "external": false
    }
  ]
}
```

Beispiel 2: Um die Hauptzuordnungen für eine Ressourcenfreigabe aufzulisten

Im folgenden `get-resource-share-associations` Beispiel werden nur die Hauptzuordnungen für die angegebene Ressourcenfreigabe aufgeführt.

```
aws ram get-resource-share-associations \
  --resource-share-arns arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE \
  --association-type PRINCIPAL
```

Ausgabe:

```
{
  "resourceShareAssociations": [
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
      "resourceShareName": "MyNewResourceShare",
      "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
      "associationType": "PRINCIPAL",
      "status": "ASSOCIATED",
      "creationTime": 1634587042.49,
      "lastUpdatedTime": 1634587044.291,
      "external": false
    }
  ]
}
```

- API-Einheiten finden Sie [GetResourceShareAssociations](#) unter AWS CLI Befehlsreferenz.

get-resource-share-invitations

Das folgende Codebeispiel zeigt die Verwendung `get-resource-share-invitations`.

AWS CLI

Um Ihre Resource Share-Einladungen aufzulisten

Im folgenden `get-resource-share-invitations` Beispiel werden Ihre aktuellen Einladungen zur gemeinsamen Nutzung von Ressourcen aufgeführt.

```
aws ram get-resource-share-invitations
```

Ausgabe:

```
{
  "resourceShareInvitations": [
    {
      "resourceShareInvitationArn": "arn:aws:ram:us-
west2-1:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-
e6117EXAMPLE",
      "resourceShareName": "project-resource-share",
      "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
fcb639f0-1449-4744-35bc-a983fEXAMPLE",
      "senderAccountId": "111111111111",
      "receiverAccountId": "222222222222",
      "invitationTimestamp": 1565312166.258,
      "status": "PENDING"
    }
  ]
}
```

- API-Einheiten finden Sie [GetResourceShareInvitations](#) in der AWS CLI Befehlsreferenz.

get-resource-shares

Das folgende Codebeispiel zeigt die Verwendung `get-resource-shares`.

AWS CLI

Beispiel 1: Um Ressourcenfreigaben aufzulisten, die Ihnen gehören und die Sie mit anderen teilen

Im folgenden `get-resource-shares` Beispiel werden die Ressourcenfreigaben aufgeführt, die Sie erstellt haben und die Sie mit anderen teilen.

```
aws ram get-resource-shares \
  --resource-owner SELF
```

Ausgabe:

```
{
  "resourceShares": [
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/3ab63985-99d9-1cd2-7d24-75e93EXAMPLE",
      "name": "my-resource-share",

```

```

    "owningAccountId": "123456789012",
    "allowExternalPrincipals": false,
    "status": "ACTIVE",
    "tags": [
      {
        "key": "project",
        "value": "lima"
      }
    ]
    "creationTime": 1565295733.282,
    "lastUpdatedTime": 1565295733.282
  },
  {
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
    "name": "my-resource-share",
    "owningAccountId": "123456789012",
    "allowExternalPrincipals": true,
    "status": "ACTIVE",
    "creationTime": 1565295733.282,
    "lastUpdatedTime": 1565295733.282
  }
]
}

```

Beispiel 2: Um Ressourcenfreigaben aufzulisten, die anderen gehören und mit Ihnen geteilt wurden

Im folgenden `get-resource-shares` Beispiel werden die Ressourcenfreigaben aufgeführt, die andere erstellt und mit Ihnen geteilt haben. In diesem Beispiel gibt es keine.

```

aws ram get-resource-shares \
  --resource-owner OTHER-ACCOUNTS

```

Ausgabe:

```

{
  "resourceShares": []
}

```

- API-Einheiten finden Sie [GetResourceShares](#) in der AWS CLI Befehlsreferenz.

list-pending-invitation-resources

Das folgende Codebeispiel zeigt die Verwendung `list-pending-invitation-resources`.

AWS CLI

Um die Ressourcen aufzulisten, die in einer ausstehenden Ressourcenfreigabe verfügbar sind

Im folgenden `list-pending-invitation-resources` Beispiel werden alle Ressourcen aufgeführt, die sich in der Ressourcenfreigabe befinden, die der angegebenen Einladung zugeordnet ist.

```
aws ram list-pending-invitation-resources \
  --resource-share-invitation-arn arn:aws:ram:us-west-2:123456789012:resource-
share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

Ausgabe:

```
{
  "resources": [
    {
      "arn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-04a555b0e6EXAMPLE",
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
      "creationTime": 1634676051.269,
      "lastUpdatedTime": 1634676052.07,
      "status": "AVAILABLE",
      "type": "ec2:Subnet"
    },
    {
      "arn": "arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "creationTime": 1624912434.431,
      "lastUpdatedTime": 1624912434.431,
      "status": "AVAILABLE",
      "type": "license-manager:LicenseConfiguration"
    }
  ]
}
```

- API-Einheiten finden Sie [ListPendingInvitationResources](#) unter AWS CLI Befehlsreferenz.

list-permissions

Das folgende Codebeispiel zeigt die Verwendung `list-permissions`.

AWS CLI

Um die verfügbaren RAM verwalteten Berechtigungen aufzulisten

Das folgende `list-permissions` Beispiel listet alle RAM verwalteten Berechtigungen auf, die nur für den Datenbankressourcentyp AWS Glue verfügbar sind.

```
aws ram list-permissions \  
  --resource-type glue:Database
```

Ausgabe:

```
{  
  "permissions": [  
    {  
      "arn": "arn:aws:ram::aws:permission/  
AWSRAMDefaultPermissionGlueDatabase",  
      "version": "1",  
      "defaultVersion": true,  
      "name": "AWSRAMDefaultPermissionGlueDatabase",  
      "resourceType": "glue:Database",  
      "creationTime": 1592007820.935,  
      "lastUpdatedTime": 1592007820.935,  
      "isResourceTypeDefault": true  
    },  
    {  
      "arn": "arn:aws:ram::aws:permission/  
AWSRAMPermissionGlueAllTablesReadWriteForDatabase",  
      "version": "2",  
      "defaultVersion": true,  
      "name": "AWSRAMPermissionGlueAllTablesReadWriteForDatabase",  
      "resourceType": "glue:Database",  
      "creationTime": 1624912413.323,  
      "lastUpdatedTime": 1624912413.323,  
      "isResourceTypeDefault": false  
    },  
  ],  
}
```

```

    {
      "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
      "version": "2",
      "defaultVersion": true,
      "name": "AWSRAMPermissionGlueDatabaseReadWrite",
      "resourceType": "glue:Database",
      "creationTime": 1624912417.4,
      "lastUpdatedTime": 1624912417.4,
      "isResourceTypeDefault": false
    },
    {
      "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
      "version": "2",
      "defaultVersion": true,
      "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
      "resourceType": "glue:Database",
      "creationTime": 1624912434.431,
      "lastUpdatedTime": 1624912434.431,
      "isResourceTypeDefault": false
    }
  ]
}

```

Im folgenden `list-permissions` Beispiel werden die verfügbaren RAM verwalteten Berechtigungen für alle Ressourcentypen angezeigt.

```
aws ram list-permissions
```

Ausgabe:

```

{
  "permissions": [
    {
      "arn": "arn:aws:ram::aws:permission/
AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
      "version": "1",
      "defaultVersion": true,
      "name":
"AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
      "resourceType": "acm-pca:CertificateAuthority",
      "creationTime": 1623264861.085,

```



```

        "lastUpdatedTime": 1623264861.085,
        "isResourceTypeDefault": false
    },
    {
        "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionAppMesh",
        "version": "1",
        "defaultVersion": true,
        "name": "AWSRAMDefaultPermissionAppMesh",
        "resourceType": "appmesh:Mesh",
        "creationTime": 1589307188.584,
        "lastUpdatedTime": 1589307188.584,
        "isResourceTypeDefault": true
    },
    ...TRUNCATED FOR BREVITY...
    {
        "arn": "arn:aws:ram::aws:permission/
AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
        "version": "1",
        "defaultVersion": true,
        "name":
"AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
        "resourceType": "acm-pca:CertificateAuthority",
        "creationTime": 1623264876.75,
        "lastUpdatedTime": 1623264876.75,
        "isResourceTypeDefault": false
    }
]
}

```

- API-Einheiten finden Sie [ListPermissions](#) in der AWS CLI Befehlsreferenz.

list-principals

Das folgende Codebeispiel zeigt die Verwendung `list-principals`.

AWS CLI

Um Principals mit Zugriff auf eine Ressource aufzulisten

Im folgenden `list-principals` Beispiel wird eine Liste der Prinzipale angezeigt, die über beliebige Ressourcenfreigaben auf Ressourcen des angegebenen Typs zugreifen können.

```
aws ram list-principals \
```

```
--resource-type ec2:Subnet
```

Ausgabe:

```
{
  "principals": [
    {
      "id": "arn:aws:organizations::123456789012:ou/o-gx7EXAMPLE/ou-29c5-
zEXAMPLE",
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "creationTime": 1565298209.737,
      "lastUpdatedTime": 1565298211.019,
      "external": false
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListPrincipals AWS CLI](#) Befehlsreferenz.

list-resource-share-permissions

Das folgende Codebeispiel zeigt die Verwendung `list-resource-share-permissions`.

AWS CLI

Um alle RAM verwalteten Berechtigungen aufzulisten, die derzeit mit einer Ressourcenfreigabe verknüpft sind

Im folgenden `list-resource-share-permissions` Beispiel werden alle RAM verwalteten Berechtigungen aufgeführt, die mit der angegebenen Ressourcenfreigabe verknüpft sind.

```
aws ram list-resource-share-permissions \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

Ausgabe:

```
{
  "permissions": [
    {
```

```

        "arn": "arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionLicenseConfiguration",
        "version": "1",
        "resourceType": "license-manager:LicenseConfiguration",
        "status": "ASSOCIATED",
        "lastUpdatedTime": 1632342984.234
    },
    {
        "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
        "version": "2",
        "resourceType": "glue:Database",
        "status": "ASSOCIATED",
        "lastUpdatedTime": 1632512462.297
    }
]
}

```

- API-Einheiten finden Sie [ListResourceSharePermissions](#) unter AWS CLI Befehlsreferenz.

list-resource-types

Das folgende Codebeispiel zeigt die Verwendung `list-resource-types`.

AWS CLI

Um die Ressourcentypen aufzulisten, die unterstützt werden von AWS RAM

Das folgende `list-resource-types` Beispiel listet alle Ressourcentypen auf, die derzeit von unterstützt werden AWS RAM.

```
aws ram list-resource-types
```

Ausgabe:

```

{
  "resourceTypes": [
    {
      "resourceType": "route53resolver:FirewallRuleGroup",
      "serviceName": "route53resolver"
    },
    {

```

```

        "resourceType": "ec2:LocalGatewayRouteTable",
        "serviceName": "ec2"
    },
    ...OUTPUT TRUNCATED FOR BREVITY...
    {
        "resourceType": "ec2:Subnet",
        "serviceName": "ec2"
    },
    {
        "resourceType": "ec2:TransitGatewayMulticastDomain",
        "serviceName": "ec2"
    }
]
}

```

- API-Einheiten finden Sie [ListResourceTypes](#) in der AWS CLI Befehlsreferenz.

list-resources

Das folgende Codebeispiel zeigt die Verwendung `list-resources`.

AWS CLI

Um die Ressourcen aufzulisten, die einer Ressourcenfreigabe zugeordnet sind

Im folgenden `list-resources` Beispiel werden alle Ressourcen in der angegebenen Ressourcenfreigabe aufgeführt, die dem angegebenen Ressourcentyp entsprechen.

```

aws ram list-resources \
  --resource-type ec2:Subnet \
  --resource-owner SELF \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE

```

Ausgabe:

```

{
  "resources": [
    {
      "arn": "aarn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1f4e15235",
      "type": "ec2:Subnet",
    }
  ]
}

```

```

        "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
        "creationTime": 1565301545.023,
        "lastUpdatedTime": 1565301545.947
    }
]
}

```

- API-Einheiten finden Sie [ListResources](#) unter AWS CLI Befehlsreferenz.

promote-resource-share-created-from-policy

Das folgende Codebeispiel zeigt die Verwendung `promote-resource-share-created-from-policy`.

AWS CLI

Um eine auf Ressourcenrichtlinien basierende gemeinsame Nutzung von Ressourcen auf den vollen Funktionsumfang hochzustufen AWS RAM

Im folgenden `promote-resource-share-created-from-policy` Beispiel wird eine Ressourcenfreigabe, die Sie implizit durch Anhängen einer ressourcenbasierten Richtlinie erstellt haben, so konvertiert, dass sie mit der Konsole und ihren Vorgängen voll funktionsfähig ist. AWS RAM CLI API

```

aws ram promote-resource-share-created-from-policy \
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-
share/91fa8429-2d06-4032-909a-90909EXAMPLE

```

Ausgabe:

```

{
  "returnValue": true
}

```

- API-Einheiten finden Sie in der Befehlsreferenz [PromoteResourceShareCreatedFromPolicy](#).AWS CLI

reject-resource-share-invitation

Das folgende Codebeispiel zeigt die Verwendung `reject-resource-share-invitation`.

AWS CLI

Um eine Einladung zur gemeinsamen Nutzung von Ressourcen abzulehnen

Im folgenden `reject-resource-share-invitation` Beispiel wird die angegebene Einladung zur gemeinsamen Nutzung einer Ressource abgelehnt.

```
aws ram reject-resource-share-invitation \  
  --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-  
share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE
```

Ausgabe:

```
"resourceShareInvitations": [  
  {  
    "resourceShareInvitationArn": "arn:aws:ram:us-west2-1:111111111111:resource-  
share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE",  
    "resourceShareName": "project-resource-share",  
    "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/  
fcb639f0-1449-4744-35bc-a983fEXAMPLE",  
    "senderAccountId": "111111111111",  
    "receiverAccountId": "222222222222",  
    "invitationTimestamp": 1565319592.463,  
    "status": "REJECTED"  
  }  
]
```

- API-Einheiten finden Sie [RejectResourceShareInvitation](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags zu einer Ressourcenfreigabe hinzuzufügen

Im folgenden `tag-resource` Beispiel werden der angegebenen Ressourcenfreigabe ein Tag-Schlüssel `project` und der zugehörige Wert `lima` hinzugefügt.

```
aws ram tag-resource \  
  --resource-arn arn:aws:ram:us-west-2:111111111111:resource-  
share/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE
```

```
--tags key=project,value=Lima \  
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Ressourcenfreigabe zu entfernen

Im folgenden `untag-resource` Beispiel werden der `project` Tag-Schlüssel und der zugehörige Wert aus der angegebenen Ressourcenfreigabe entfernt.

```
aws ram untag-resource \  
--tag-keys project \  
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) unter AWS CLI Befehlsreferenz.

update-resource-share

Das folgende Codebeispiel zeigt die Verwendung `update-resource-share`.

AWS CLI

Um eine Ressourcenfreigabe zu aktualisieren

Im folgenden `update-resource-share` Beispiel wird die angegebene Ressourcenfreigabe so geändert, dass externe Prinzipale zugelassen werden, die sich nicht in einer AWS Organisation befinden.

```
aws ram update-resource-share \  
--allow-external-principals \  

```

```
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

Ausgabe:

```
{
  "resourceShare": {
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
    "name": "my-resource-share",
    "owningAccountId": "123456789012",
    "allowExternalPrincipals": true,
    "status": "ACTIVE",
    "creationTime": 1565295733.282,
    "lastUpdatedTime": 1565303080.023
  }
}
```

- API-Einheiten finden Sie unter [UpdateResourceShare AWS CLI](#) Befehlsreferenz.

Resource Explorer-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Resource Explorer Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-default-view

Das folgende Codebeispiel zeigt die Verwendung `associate-default-view`.

AWS CLI

So legen Sie eine Resource Explorer-Ansicht als Standard für die zugehörige AWS Region fest

Im folgenden `associate-default-view` Beispiel wird eine Ansicht, wie in `its` angegebenARN, als Standardansicht für die AWS Region festgelegt, in der Sie die Operation aufrufen.

```
aws resource-explorer-2 associate-default-view \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Ausgabe:

```
{  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Einrichten einer Standardansicht in einer AWS Region](#) im AWS Resource Explorer-Benutzerhandbuch.

- APIEinzelheiten finden Sie [AssociateDefaultView](#) unter AWS CLI Befehlsreferenz.

batch-get-view

Das folgende Codebeispiel zeigt die Verwendung `batch-get-view`.

AWS CLI

Um Details zu mehreren Resource Explorer-Ansichten abzurufen

Im folgenden `batch-get-view` Beispiel werden die Details zu zwei Ansichten angezeigt, die durch ihre angegeben sindARNs. Verwenden Sie Leerzeichen, um das Vielfache ARNs im Parameter `--view-arn` zu trennen.

```
aws resource-explorer-2 batch-get-view \  
  --view-arns arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222, \  
arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Ausgabe:

```
{
  "Views": [
    {
      "Filters": {
        "FilterString": "service:ec2"
      },
      "IncludedProperties": [
        {
          "Name": "tags"
        }
      ],
      "LastUpdatedAt": "2022-07-13T21:33:45.249000+00:00",
      "Owner": "123456789012",
      "Scope": "arn:aws:iam::123456789012:root",
      "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    },
    {
      "Filters": {
        "FilterString": ""
      },
      "IncludedProperties": [
        {
          "Name": "tags"
        }
      ],
      "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",
      "Owner": "123456789012",
      "Scope": "arn:aws:iam::123456789012:root",
      "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
    }
  ]
  "Errors": []
}
```

Weitere Informationen zu Ansichten finden Sie unter [Über Resource Explorer-Ansichten im AWS Resource Explorer-Benutzerhandbuch](#).

- API-Einheiten finden Sie [BatchGetView](#) in der AWS CLI Befehlsreferenz.

create-index

Das folgende Codebeispiel zeigt die Verwendung `create-index`.

AWS CLI

So aktivieren Sie den Resource Explorer in einer AWS Region, indem Sie einen Index erstellen

Im folgenden `create-index` Beispiel wird ein lokaler Index in der AWS Region erstellt, in der die Operation aufgerufen wird. Der generiert AWS CLI automatisch einen zufälligen `client-token` Parameterwert und schließt ihn in den Aufruf von ein, AWS falls Sie keinen Wert angeben.

```
aws resource-explorer-2 create-index \  
  --region us-east-1
```

Ausgabe:

```
{  
  "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE22222c",  
  "CreatedAt": "2022-11-01T20:00:59.149Z",  
  "State": "CREATING"  
}
```

Nachdem Sie einen lokalen Index erstellt haben, können Sie ihn in den Aggregatorindex für das Konto konvertieren, indem Sie den [update-index-type](#) Befehl ausführen.

Weitere Informationen finden Sie im Resource Explorer-Benutzerhandbuch [unter Resource Explorer in einer AWS Region aktivieren, um Ihre Ressourcen zu indizieren](#).AWS

- API-Einheiten finden Sie [CreateIndex](#) in der AWS CLI Befehlsreferenz.

create-view

Das folgende Codebeispiel zeigt die Verwendung `create-view`.

AWS CLI

Beispiel 1: Um eine ungefilterte Ansicht für den Index in einer AWS Region zu erstellen

Im folgenden `create-view` Beispiel wird eine Ansicht in der angegebenen AWS Region erstellt, die alle Ergebnisse in der Region ohne jegliche Filterung zurückgibt. Die Ansicht enthält das

optionale Feld Tags für die zurückgegebenen Ergebnisse. Da diese Ansicht in der Region erstellt wird, die den Aggregatorindex enthält, kann sie Ergebnisse aus allen Regionen des Kontos enthalten, die einen Resource Explorer-Index enthalten.

```
aws resource-explorer-2 create-view \  
  --view-name My-Main-View \  
  --included-properties Name=tags \  
  --region us-east-1
```

Ausgabe:

```
{  
  "View": {  
    "Filters": {  
      "FilterString": ""  
    },  
    "IncludedProperties": [  
      {  
        "Name": "tags"  
      }  
    ],  
    "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",  
    "Owner": "123456789012",  
    "Scope": "arn:aws:iam::123456789012:root",  
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
  }  
}
```

Beispiel 2: So erstellen Sie eine Ansicht, die nur Ressourcen zurückgibt, die mit Amazon verknüpft sind EC2

Im Folgenden `create-view` wird eine Ansicht in AWS Region `us-east-1` erstellt, die nur die Ressourcen in der Region zurückgibt, die mit dem EC2 Amazon-Service verknüpft sind. Die Ansicht enthält das optionale `Tags` Feld für zurückgegebene Ergebnisse. Da diese Ansicht in der Region erstellt wird, die den Aggregatorindex enthält, kann sie Ergebnisse aus allen Regionen des Kontos enthalten, die einen Resource Explorer-Index enthalten.

```
aws resource-explorer-2 create-view \  
  --view-name My-EC2-Only-View \  
  --included-properties Name=tags \  
  --region us-east-1
```

```
--filters FilterString="service:ec2" \  
--region us-east-1
```

Ausgabe:

```
{  
  "View": {  
    "Filters": {  
      "FilterString": "service:ec2"  
    },  
    "IncludedProperties": [  
      {  
        "Name": "tags"  
      }  
    ],  
    "LastUpdatedAt": "2022-07-13T21:35:09.059Z",  
    "Owner": "123456789012",  
    "Scope": "arn:aws:iam::123456789012:root",  
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-  
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"  
  }  
}
```

Weitere Informationen finden Sie unter [Erstellen von Ansichten für die Suche](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateView](#) unter AWS CLI Befehlsreferenz.

delete-index

Das folgende Codebeispiel zeigt die Verwendung `delete-index`.

AWS CLI

Um den Resource Explorer in einer AWS Region durch Löschen des zugehörigen Indexes auszuschalten

Im folgenden `delete-index` Beispiel wird der angegebene Resource Explorer-Index in der AWS Region gelöscht, in der Sie die Anforderung stellen.

```
aws resource-explorer-2 delete-index \  
--arn arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE22222 \  

```

```
--region us-west-2
```

Ausgabe:

```
{
  "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222",
  "State": "DELETING"
}
```

Weitere Informationen zum Löschen eines Indexes finden Sie unter [Ausschalten des AWS Resource Explorer-Explorers in einer AWS Region](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteIndex](#) in der AWS CLI Befehlsreferenz.

delete-view

Das folgende Codebeispiel zeigt die Verwendung `delete-view`.

AWS CLI

Um eine Resource Explorer-Ansicht zu löschen

Im folgenden `delete-view` Beispiel wird eine durch ihre ARN angegebene Ansicht gelöscht.

```
aws resource-explorer-2 delete-view \  
--view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Ausgabe:

```
{
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Weitere Informationen finden Sie unter [Löschen von Ansichten](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteView](#) in der AWS CLI Befehlsreferenz.

disassociate-default-view

Das folgende Codebeispiel zeigt die Verwendung `disassociate-default-view`.

AWS CLI

Um die Resource Explorer-Standardansicht für eine AWS Region zu entfernen

Im Folgenden `disassociate-default-view` wird die Resource Explorer-Standardansicht für die AWS Region entfernt, in der Sie den Vorgang aufrufen. Nach der Ausführung dieses Vorgangs müssen alle Suchvorgänge in der Region explizit eine Ansicht angeben. Andernfalls schlägt der Vorgang fehl.

```
aws resource-explorer-2 disassociate-default-view
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten einer Standardansicht in einer AWS Region](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateDefaultView](#) unter AWS CLI Befehlsreferenz.

get-default-view

Das folgende Codebeispiel zeigt die Verwendung `get-default-view`.

AWS CLI

Um die Resource Explorer-Ansicht abzurufen, die die Standardansicht für die entsprechende AWS Region ist

Im folgenden `get-default-view` Beispiel wird die ARN Ansicht abgerufen, die die Standardansicht für die AWS Region ist, in der Sie die Operation aufrufen.

```
aws resource-explorer-2 get-default-view
```

Ausgabe:

```
{
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Weitere Informationen finden Sie unter [Einrichten einer Standardansicht für eine AWS Region](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [GetDefaultView](#) unter AWS CLI Befehlsreferenz.

get-index

Das folgende Codebeispiel zeigt die Verwendung `get-index`.

AWS CLI

Beispiel 1: Um die Details für einen Resource Explorer-Aggregatorindex abzurufen

Im folgenden `get-index` Beispiel werden die Details für den Resource Explorer-Index in der angegebenen AWS Region angezeigt. Da die angegebene Region den Aggregatorindex für das Konto enthält, werden in der Ausgabe die Regionen aufgeführt, die Daten in den Index dieser Region replizieren.

```
aws resource-explorer-2 get-index \  
  --region us-east-1
```

Ausgabe:

```
{  
  "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE11111",  
  "CreatedAt": "2022-07-12T18:59:10.503000+00:00",  
  "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",  
  "ReplicatingFrom": [  
    "ap-south-1",  
    "us-west-2"  
  ],  
  "State": "ACTIVE",  
  "Tags": {},  
  "Type": "AGGREGATOR"  
}
```

Beispiel 2: Um die Details für einen lokalen Resource Explorer-Index abzurufen

Im folgenden `get-index` Beispiel werden die Details für den Resource Explorer-Index in der angegebenen AWS Region angezeigt. Da die angegebene Region einen lokalen Index enthält,

wird in der Ausgabe die Region aufgeführt, in die Daten aus dem Index dieser Region repliziert werden.

```
aws resource-explorer-2 get-index \  
  --region us-west-2
```

Ausgabe:

```
{  
  "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE22222",  
  "CreatedAt": "2022-07-12T18:59:10.503000+00:00",  
  "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",  
  "ReplicatingTo": [  
    "us-west-2"  
  ],  
  "State": "ACTIVE",  
  "Tags": {},  
  "Type": "LOCAL"  
}
```

Weitere Informationen zu Indizes finden Sie im [Resource Explorer-Benutzerhandbuch unter Überprüfen, in welchen AWS Regionen der AWS Resource Explorer aktiviert ist](#).

- API-Einheiten finden Sie [GetIndex](#) in der AWS CLI Befehlsreferenz.

get-view

Das folgende Codebeispiel zeigt die Verwendung `get-view`.

AWS CLI

Um Details zu einer Resource Explorer-Ansicht abzurufen

Im folgenden `get-view` Beispiel werden die Details zu einer Ansicht angezeigt, die in `its` angegeben ist ARN.

```
aws resource-explorer-2 get-view \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Ausgabe:

```
{
  "Tags" : {},
  "View" : {
    "Filters" : {
      "FilterString" : "service:ec2"
    },
    "IncludedProperties" : [
      {
        "Name" : "tags"
      }
    ],
    "LastUpdatedAt" : "2022-07-13T21:33:45.249Z",
    "Owner" : "123456789012",
    "Scope" : "arn:aws:iam::123456789012:root",
    "ViewArn" : "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }
}
```

Weitere Informationen zu Ansichten finden Sie unter [Über Resource Explorer-Ansichten](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [GetView](#) in der AWS CLI Befehlsreferenz.

list-indexes

Das folgende Codebeispiel zeigt die Verwendung `list-indexes`.

AWS CLI

Um die AWS Regionen aufzulisten, in denen Resource Explorer Indizes hat

Das folgende `list-indexes` Beispiel listet die Indizes für alle Regionen auf, in denen Resource Explorer über einen Index verfügt. Die Antwort gibt den Typ jedes Indexes, seine AWS Region und seinen ARN an.

```
aws resource-explorer-2 list-indexes
```

Ausgabe:

```
{
```

```

    "Indexes": [
      {
        "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
        "Region": "us-west-2",
        "Type": "AGGREGATOR"
      },
      {
        "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
        "Region": "us-east-1",
        "Type": "LOCAL"
      },
      {
        "Arn": "arn:aws:resource-explorer-2:us-east-2:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333",
        "Region": "us-east-2",
        "Type": "LOCAL"
      },
      {
        "Arn": "arn:aws:resource-explorer-2:us-west-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE44444",
        "Region": "us-west-1",
        "Type": "LOCAL"
      }
    ]
  }

```

Weitere Informationen zu Indizes finden Sie im [Resource Explorer-Benutzerhandbuch unter Überprüfen, in welchen AWS Regionen der AWS Resource Explorer aktiviert ist](#).

- API-Einheiten finden Sie [ListIndexes](#) in der AWS CLI Befehlsreferenz.

list-supported-resource-types

Das folgende Codebeispiel zeigt die Verwendung `list-supported-resource-types`.

AWS CLI

Um die AWS Regionen aufzulisten, in denen Resource Explorer Indizes hat

Das folgende `list-supported-resource-types` Beispiel listet alle Ressourcentypen auf, die derzeit von `&AREXlong`; unterstützt werden. Die Beispielantwort enthält einen `NextToken` Wert,

der angibt, dass mehr Ausgaben verfügbar sind, die mit zusätzlichen Aufrufen abgerufen werden können.

```
aws resource-explorer-2 list-supported-resource-types \  
--max-items 10
```

Ausgabe:

```
{  
  "ResourceTypes": [  
    {  
      "ResourceType": "cloudfront:cache-policy",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudfront:distribution",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudfront:function",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudfront:origin-access-identity",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudfront:origin-request-policy",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudfront:realtime-log-config",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudfront:response-headers-policy",  
      "Service": "cloudfront"  
    },  
    {  
      "ResourceType": "cloudwatch:alarm",  
      "Service": "cloudwatch"  
    },  
    {
```

```

    "ResourceType": "cloudwatch:dashboard",
    "Service": "cloudwatch"
  },
  {
    "ResourceType": "cloudwatch:insight-rule",
    "Service": "cloudwatch"
  }
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}

```

Um den nächsten Teil der Ausgabe abzurufen, rufen Sie den Vorgang erneut auf und übergeben den NextToken Antwortwert des vorherigen Aufrufs als Wert für `--starting-token`. Wiederholen Sie den Vorgang, bis NextToken der Eintrag in der Antwort fehlt.

```

aws resource-explorer-2 list-supported-resource-types \
  --max-items 10 \
  --starting-token eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=

```

Ausgabe:

```

{
  "ResourceTypes": [
    {
      "ResourceType": "cloudwatch:metric-stream",
      "Service": "cloudwatch"
    },
    {
      "ResourceType": "dynamodb:table",
      "Service": "dynamodb"
    },
    {
      "ResourceType": "ec2:capacity-reservation",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:capacity-reservation-fleet",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:client-vpn-endpoint",

```

```

    "Service": "ec2"
  },
  {
    "ResourceType": "ec2:customer-gateway",
    "Service": "ec2"
  },
  {
    "ResourceType": "ec2:dedicated-host",
    "Service": "ec2"
  },
  {
    "ResourceType": "ec2:dhcp-options",
    "Service": "ec2"
  },
  {
    "ResourceType": "ec2:egress-only-internet-gateway",
    "Service": "ec2"
  },
  {
    "ResourceType": "ec2:elastic-gpu",
    "Service": "ec2"
  }
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyMH0="
}

```

Weitere Informationen zu Indizes finden Sie im [Resource Explorer-Benutzerhandbuch unter Überprüfen, in welchen AWS Regionen der AWS Resource Explorer aktiviert ist](#).

- API-Einheiten finden Sie [ListSupportedResourceTypes](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags aufzulisten, die einer Resource Explorer-Ansicht oder einem Index zugeordnet sind

Das folgende `list-tags-for-resource` Beispiel listet die Tag-Schlüssel- und Wertepaare auf, die der Ansicht mit den angegebenen Werten zugeordnet sind. Sie müssen den Vorgang von der AWS Region aus aufrufen, die die Ressource enthält.

```
aws resource-explorer-2 list-tags-for-resource \  
  --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

Ausgabe:

```
{  
  "Tags": {  
    "application": "MainCorpApp",  
    "department": "1234"  
  }  
}
```

Weitere Informationen zum Taggen von Ansichten finden Sie unter [Tagging Views für die Zugriffskontrolle](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-views

Das folgende Codebeispiel zeigt die Verwendung `list-views`.

AWS CLI

Um die in einer AWS Region verfügbaren Resource Explorer-Ansichten aufzulisten

Im folgenden `list-views` Beispiel werden alle Ansichten aufgeführt, die in der Region verfügbar sind, in der Sie den Vorgang aufrufen.

```
aws resource-explorer-2 list-views
```

Ausgabe:

```
{  
  "Views": [  
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",  
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Default-All-  
Resources-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",  
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Production-Only-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333"  
  ]  
}
```

```
}
```

Weitere Informationen zu Ansichten finden Sie unter [Über Resource Explorer-Ansichten](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [ListViews](#) in der AWS CLI Befehlsreferenz.

search

Das folgende Codebeispiel zeigt die Verwendung `search`.

AWS CLI

Beispiel 1: Um mit der Standardansicht zu suchen

Im folgenden `search` Beispiel werden alle Ressourcen in der angegebenen Liste angezeigt, die dem Dienst zugeordnet sind. Die Suche verwendet die Standardansicht für die Region. Die Beispielantwort enthält einen `NextToken` Wert, der angibt, dass mehr Ausgaben verfügbar sind, die mit zusätzlichen Aufrufen abgerufen werden können.

```
aws resource-explorer-2 search \  
  --query-string "service:iam"
```

Ausgabe:

```
{  
  "Count": {  
    "Complete": true,  
    "TotalResources": 55  
  },  
  "NextToken":  
  "AG9V0EF1KLEXAMPLE0hJHVwo5chEXAMPLER5XiEpNrgsEXAMPLE...b0Cm0F0ryHEXAMPLE",  
  "Resources": [{  
    "Arn": "arn:aws:iam::123456789012:policy/service-role/Some-Policy-For-A-  
Service-Role",  
    "LastReportedAt": "2022-07-21T12:34:42Z",  
    "OwningAccountId": "123456789012",  
    "Properties": [],  
    "Region": "global",  
    "ResourceType": "iam:policy",  
    "Service": "iam"  
  }, {
```



```

    "Arn": "arn:aws:iam::123456789012:policy/service-role/Another-Policy-For-A-
Service-Role",
    "LastReportedAt": "2022-07-21T12:34:42Z",
    "OwningAccountId": "123456789012",
    "Properties": [],
    "Region": "global",
    "ResourceType": "iam:policy",
    "Service": "iam"
  }, {
    ... TRUNCATED FOR BREVITY ...
  }],
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/my-default-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}

```

Beispiel 2: Um mit einer bestimmten Ansicht zu suchen

In der folgenden `search` Beispielsuche werden alle Ressourcen („*“) in der angegebenen AWS Region angezeigt, die in der angegebenen Ansicht sichtbar sind. Die Ergebnisse enthalten EC2 aufgrund der Filter, die der Ansicht zugewiesen sind, nur Ressourcen, die mit Amazon verknüpft sind.

```

aws resource-explorer-2 search \
  -- query-string "*" \
  -- view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-view/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222

```

Ausgabe:

```

HTTP/1.1 200 OK
Date: Tue, 01 Nov 2022 20:00:59 GMT
Content-Type: application/json
Content-Length: <PayloadSizeBytes>

{
  "Count": {
    "Complete": true,
    "TotalResources": 67
  },
  "Resources": [{
    "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/acl-1a2b3c4d",
    "LastReportedAt": "2022-07-21T18:52:02Z",

```

```
"OwningAccountId": "123456789012",
"Properties": [{
  "Data": [{
    "Key": "Department",
    "Value": "AppDevelopment"
  }, {
    "Key": "Environment",
    "Value": "Production"
  }],
  "LastReportedAt": "2021-11-15T14:48:29Z",
  "Name": "tags"
}],
"Region": "us-east-1",
"ResourceType": "ec2:network-acl",
"Service": "ec2"
}, {
  "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-1a2b3c4d",
  "LastReportedAt": "2022-07-21T21:22:23Z",
  "OwningAccountId": "123456789012",
  "Properties": [{
    "Data": [{
      "Key": "Department",
      "Value": "AppDevelopment"
    }, {
      "Key": "Environment",
      "Value": "Production"
    }],
    "LastReportedAt": "2021-07-29T19:02:39Z",
    "Name": "tags"
  }],
  "Region": "us-east-1",
  "ResourceType": "ec2:subnet",
  "Service": "ec2"
}, {
  "Arn": "arn:aws:ec2:us-east-1:123456789012:dhcp-options/dopt-1a2b3c4d",
  "LastReportedAt": "2022-07-21T06:08:53Z",
  "OwningAccountId": "123456789012",
  "Properties": [{
    "Data": [{
      "Key": "Department",
      "Value": "AppDevelopment"
    }, {
      "Key": "Environment",
      "Value": "Production"
    }],
    "LastReportedAt": "2021-07-29T19:02:39Z",
    "Name": "tags"
  }],
  "Region": "us-east-1",
  "ResourceType": "ec2:dhcp-options",
  "Service": "ec2"
}
```

```

        }],
        "LastReportedAt": "2021-11-15T15:11:05Z",
        "Name": "tags"
    }],
    "Region": "us-east-1",
    "ResourceType": "ec2:dhcptions",
    "Service": "ec2"
}, {
    ... TRUNCATED FOR BREVITY ...
}],
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
}

```

Weitere Informationen finden Sie unter [Verwenden von AWS Resource Explorer zur Suche nach Ressourcen](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie unter In der AWS CLI Befehlsreferenz [suchen](#).

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Resource Explorer-Ansicht zu taggen

Im folgenden `tag-resource` Beispiel wird der Ansicht mit dem angegebenen Wert der Tag-Schlüssel „environment“ mit dem Wert „production“ hinzugefügtARN.

```

aws resource-explorer-2 tag-resource \
  --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//  

EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
  --tags environment=production

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Views for Access Control](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie unter [TagResource AWS CLI](#) Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einer Resource Explorer-Ansicht zu entfernen

Im folgenden `untag-resource` Beispiel werden alle Tags mit dem Schlüsselnamen „environment“ aus der Ansicht mit dem angegebenen Wert entferntARN.

```
aws resource-explorer-2 untag-resource \  
  --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \  
  --tag-keys environment
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Views for Access Control](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UntagResource AWS CLI Befehlsreferenz](#).

update-index-type

Das folgende Codebeispiel zeigt die Verwendung `update-index-type`.

AWS CLI

Um den Typ eines Resource Explorer-Indexes zu ändern

Im folgenden `update-index-type` Beispiel wird der angegebene Index von Typ `local` zu Typ `aggregator`, um die Möglichkeit zu aktivieren, in allen AWS Regionen des Kontos nach Ressourcen zu suchen. Sie müssen die Anforderung an die AWS Region senden, die den Index enthält, den Sie aktualisieren möchten.

```
aws resource-explorer-2 update-index-type \  
  --arn arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE11111 \  
  --type aggregator \  
  --region us-east-1
```

Ausgabe:

```
{
  "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE11111",
  "LastUpdatedAt": "2022-07-13T18:41:58.799Z",
  "State": "updating",
  "Type": "aggregator"
}
```

Weitere Informationen zum Ändern des Indextyps finden Sie [unter Aktivieren der regionsübergreifenden Suche durch Erstellen eines Aggregatorindexes](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UpdateIndexType AWS CLI Befehlsreferenz](#).

update-view

Das folgende Codebeispiel zeigt die Verwendung `update-view`.

AWS CLI

Beispiel 1: Um das `IncludedProperties` Feld für eine Resource Explorer-Ansicht zu aktualisieren

Im folgenden `update-view` Beispiel wird die angegebene Ansicht aktualisiert, indem `tags` die optionale Ansicht erweitert wird `IncludedProperties`. Nach dem Ausführen dieses Vorgangs enthalten Suchvorgänge, die diese Ansicht verwenden, Informationen zu den Tags, die den Ressourcen zugeordnet sind und in den Ergebnissen erscheinen.

```
aws resource-explorer-2 update-view \
  --included-properties Name=tags \
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

Ausgabe:

```
{
  "View": {
    "Filters": {
      "FilterString": ""
    },

```

```

    "IncludedProperties": [
      {
        "Name": "tags"
      }
    ],
    "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",
    "Owner": "123456789012",
    "Scope": "arn:aws:iam::123456789012:root",
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }
}

```

Beispiel 2: Um die Filter zu aktualisieren, die einer Ansicht zugeordnet sind

Im folgenden `update-view` Beispiel wird die angegebene Ansicht aktualisiert, sodass sie einen Filter verwendet, der die Ergebnisse nur auf Ressourcentypen beschränkt, die mit dem EC2 Amazon-Service verknüpft sind.

```

aws resource-explorer-2 update-view \
  --filters FilterString="service:ec2" \
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222

```

Ausgabe:

```

{
  "View": {
    "Filters": {
      "FilterString": "service:ec2"
    },
    "IncludedProperties": [],
    "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",
    "Owner": "123456789012",
    "Scope": "arn:aws:iam::123456789012:root",
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
  }
}

```

Weitere Informationen zu Ansichten finden Sie unter [Über Resource Explorer-Ansichten](#) im AWS Resource Explorer-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateView](#) in der AWS CLI Befehlsreferenz.

Beispiele für Resource Groups mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Resource Groups Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-group

Das folgende Codebeispiel zeigt die Verwendung `create-group`.

AWS CLI

Beispiel 1: Um eine tagbasierte Ressourcengruppe zu erstellen

Im folgenden `create-group` Beispiel wird eine Tag-basierte Ressourcengruppe von EC2 Amazon-Instances in der aktuellen Region erstellt. Es basiert auf einer Abfrage nach Ressourcen, die mit dem Schlüssel `Name` und dem Wert `WebServers` gekennzeichnet sind. Der Gruppenname ist `tbq-WebServer`. Die Abfrage befindet sich in einer separaten JSON Datei, die an den Befehl übergeben wird.

```
aws resource-groups create-group \  
  --name tbq-WebServer \  
  --resource-query file://query.json
```

Inhalt von `query.json`:

```
{
```

```

    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [ { \"Key\": \"Name\", \"Values\": [ \"WebServers\" ] } ] }"
  }

```

Ausgabe:

```

{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Name": "tbq-WebServer"
  },
  "ResourceQuery": {
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [ { \"Key\": \"Name\", \"Values\": [ \"WebServers\" ] } ] }"
  }
}

```

Beispiel 2: So erstellen Sie eine CloudFormation stapelbasierte Ressourcengruppe

Im folgenden `create-group` Beispiel wird eine AWS CloudFormation stapelbasierte Ressourcengruppe mit dem Namen `sampleCFNstackgroup` erstellt. Die Abfrage umfasst alle Ressourcen im angegebenen CloudFormation Stack, die von AWS Resource Groups unterstützt werden.

```

aws resource-groups create-group \
  --name cbq-CFNstackgroup \
  --resource-query file://query.json

```

Inhalt von `query.json`:

```

{
  "Type": "CLOUDFORMATION_STACK_1_0",
  "Query": "{\"ResourceTypeFilters\": [\"AWS::AllSupported\"], \"StackIdentifier\": \"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-z39z-11z8-97z5-500z212zz6fz\"}"
}

```

Ausgabe:


```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNstackgroup",
    "Name": "cbq-CFNstackgroup"
  },
  "ResourceQuery": {
    "Type": "CLOUDFORMATION_STACK_1_0",
    "Query": "{\"ResourceTypeFilters\": [\"AWS::AllSupported\"], \"StackIdentifier
\": \"arn:aws:cloudformation:us-east-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
  }
}
```

Weitere Informationen finden Sie unter [Create Groups](#) im AWS Resource Groups User Guide.

- API-Einheiten finden Sie [CreateGroup](#) in der AWS CLI Befehlsreferenz.

delete-group

Das folgende Codebeispiel zeigt die Verwendung `delete-group`.

AWS CLI

Um die Beschreibung für eine Ressourcengruppe zu aktualisieren

Im folgenden `delete-group` Beispiel wird die angegebene Ressourcengruppe aktualisiert.

```
aws resource-groups delete-group \
  --group-name tbq-WebServer
```

Ausgabe:

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-
WebServer",
    "Name": "tbq-WebServer"
  }
}
```

Weitere Informationen finden Sie unter [Gruppen löschen](#) im AWS Resource Groups User Guide.

- API-Einheiten finden Sie [DeleteGroup](#) in der AWS CLI Befehlsreferenz.

get-group-query

Das folgende Codebeispiel zeigt die Verwendung `get-group-query`.

AWS CLI

Um die Abfrage an eine Ressourcengruppe anzuhängen

Im folgenden `get-group-query` Beispiel wird die Abfrage angezeigt, die der angegebenen Ressourcengruppe zugeordnet ist.

```
aws resource-groups get-group-query \  
  --group-name tbq-WebServer
```

Ausgabe:

```
{  
  "GroupQuery": {  
    "GroupName": "tbq-WebServer",  
    "ResourceQuery": {  
      "Type": "TAG_FILTERS_1_0",  
      "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\  
\": [{\"Key\": \"Name\", \"Values\": [\"WebServers\"]}]}"  
    }  
  }  
}
```

- API-Einheiten finden Sie [GetGroupQuery](#) unter AWS CLI Befehlsreferenz.

get-group

Das folgende Codebeispiel zeigt die Verwendung `get-group`.

AWS CLI

Um Informationen über eine Ressourcengruppe zu erhalten

Im folgenden `get-group` Beispiel werden Details zur angegebenen Ressourcengruppe angezeigt. Um die Abfrage an die Gruppe anzuhängen, verwenden Sie `get-group-query`.

```
aws resource-groups get-group \  
  --group-name tbq-WebServer
```

Ausgabe:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-  
WebServer",  
    "Name": "tbq-WebServer",  
    "Description": "A tag-based query resource group of WebServers."  
  }  
}
```

- API-Einheiten finden Sie [GetGroup](#) in der AWS CLI Befehlsreferenz.

get-tags

Das folgende Codebeispiel zeigt die Verwendung `get-tags`.

AWS CLI

Um die einer Ressourcengruppe angehängten Tags abzurufen

Im folgenden `get-tags` Beispiel werden die Tag-Schlüssel- und Wertepaare angezeigt, die der angegebenen Ressourcengruppe (der Gruppe selbst, nicht ihren Mitgliedern) zugeordnet sind.

```
aws resource-groups get-tags \  
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

Ausgabe:

```
{  
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",  
  "Tags": {  
    "QueryType": "tags",  
    "QueryResources": "ec2-instances"  
  }  
}
```

- API-Einheiten finden Sie [GetTags](#) unter AWS CLI Befehlsreferenz.

list-group-resources

Das folgende Codebeispiel zeigt die Verwendung `list-group-resources`.

AWS CLI

Um alle Ressourcen in einer Ressourcengruppe aufzulisten

Beispiel 1: Das folgende `list-resource-groups` Beispiel listet alle Ressourcen auf, die Teil der angegebenen Ressourcengruppe sind.

```
aws resource-groups list-group-resources \  
  --group-name tbq-WebServer
```

Ausgabe:

```
{  
  "ResourceIdentifiers": [  
    {  
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/  
i-09f77fa38c12345ab",  
      "ResourceType": "AWS::EC2::Instance"  
    }  
  ]  
}
```

Beispiel 2: Das folgende Beispiel listet alle Ressourcen in der Gruppe auf, die auch den 'Ressourcentyp'AWS:: EC2 :Instanz 'haben . :

```
aws resource-groups list-group-resources --group-name tbq- WebServer --filters  
name=Ressourcentyp, Values=:: :Instanz AWS EC2
```

- APIEinzelheiten finden Sie [ListGroupResources](#) in AWS CLI der Befehlsreferenz.

list-groups

Das folgende Codebeispiel zeigt die Verwendung `list-groups`.

AWS CLI

Um die verfügbaren Ressourcengruppen aufzulisten

Im folgenden `list-groups` Beispiel wird eine Liste aller Ressourcengruppen angezeigt.

```
aws resource-groups list-groups
```

Ausgabe:

```
{
  "GroupIdentifiers": [
    {
      "GroupName": "tbq-WebServer",
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer3"
    },
    {
      "GroupName": "cbq-CFNStackQuery",
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNStackQuery"
    }
  ],
  "Groups": [
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
      "Name": "tbq-WebServer"
    },
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNStackQuery",
      "Name": "cbq-CFNStackQuery"
    }
  ]
}
```

- API-Einheiten finden Sie [ListGroups](#) unter AWS CLI Befehlsreferenz.

list-resource-groups

Das folgende Codebeispiel zeigt die Verwendung `list-resource-groups`.

AWS CLI

Um alle Ressourcen in einer Ressourcengruppe aufzulisten

Das folgende `list-resource-groups` Beispiel listet alle Ressourcen auf, die Teil der angegebenen Ressourcengruppe sind.

```
aws resource-groups list-group-resources \  
  --group-name tbq-WebServer
```

Ausgabe:

```
{  
  "ResourceIdentifiers": [  
    {  
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/  
i-09f77fa38c12345ab",  
      "ResourceType": "AWS::EC2::Instance"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListResourceGroups](#) unter AWS CLI Befehlsreferenz.

put-group-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-group-configuration`.

AWS CLI

Um eine Dienstkonfiguration an eine Ressourcengruppe anzuhängen

Beispiel 1: Das folgende `put-group-configuration` Beispiel gibt an, dass die Ressourcengruppe nur EC2 Amazon-Kapazitätsreservierungen für Instances der M5 Familien C5 oder enthalten soll.

```
aws resource-groups put-group-configuration \  
  --group MyTestGroup \  
  --configuration file://config.json
```

Inhalt von `config.json`:

```
[  
  {  
    "Type": "AWS::EC2::HostManagement",
```

```

    "Parameters": [
      {
        "Name": "allowed-host-families",
        "Values": [ "c5", "m5" ]
      },
      {
        "Name": "any-host-based-license-configuration",
        "Values": [ "true" ]
      }
    ]
  },
  {
    "Type": "AWS::ResourceGroups::Generic",
    "Parameters": [
      {
        "Name": "allowed-resource-types",
        "Values": [ "AWS::EC2::Host" ]
      },
      {
        "Name": "deletion-protection",
        "Values": [ "UNLESS_EMPTY" ]
      }
    ]
  }
]

```

Dieser Befehl erzeugt bei Erfolg keine Ausgabe.

Weitere Informationen finden Sie unter [Dienstkonfigurationen für Resource Groups](#) im APIReferenzhandbuch für Ressourcengruppen.

- APIEinzelheiten finden Sie [PutGroupConfiguration](#) in der AWS CLI Befehlsreferenz.

search-resources

Das folgende Codebeispiel zeigt die Verwendung `search-resources`.

AWS CLI

Um Ressourcen zu finden, die einer Abfrage entsprechen

Im folgenden `search-resources` Beispiel wird eine Liste aller AWS Ressourcen abgerufen, die der angegebenen Abfrage entsprechen.

```
aws resource-groups search-resources \
  --resource-query file://query.json
```

Inhalt von `query.json`:

```
{
  "Type": "TAG_FILTERS_1_0",
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [ { \"Key\": \"Patch Group\", \"Values\": [\"Dev\"] } ]}"
}
```

Ausgabe:

```
{
  "ResourceIdentifiers": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-01a23bc45d67890ef",
      "ResourceType": "AWS::EC2::Instance"
    }
  ]
}
```

- API-Einheiten finden Sie unter [SearchResources AWS CLI Befehlsreferenz](#).

tag

Das folgende Codebeispiel zeigt die Verwendung `tag`.

AWS CLI

Um ein Tag an eine Ressourcengruppe anzuhängen

Im folgenden `tag` Beispiel werden die angegebenen Tag-Schlüssel- und Wertepaare an die angegebene Ressourcengruppe angehängt (die Gruppe selbst, nicht ihre Mitglieder).

```
aws resource-groups tag \
  --tags QueryType=tags, QueryResources=ec2-instances \
  --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer
```


Ausgabe:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
  "Tags": {
    "QueryType": "tags",
    "QueryResources": "ec2-instances"
  }
}
```

Weitere Informationen finden Sie unter [Tags verwalten](#) im AWS Resource Groups User Guide.

- API-Einheiten finden Sie unter [Tag](#) in der AWS CLI Befehlsreferenz.

untag

Das folgende Codebeispiel zeigt die Verwendung `untag`.

AWS CLI

Um Tags aus einer Ressourcengruppe zu entfernen

Im folgenden `untags` Beispiel werden alle Tags mit dem angegebenen Schlüssel aus der Ressourcengruppe selbst entfernt, nicht aus ihren Mitgliedern.

```
aws resource-groups untag \
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \
  --keys QueryType
```

Ausgabe:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
  "Keys": [
    "QueryType"
  ]
}
```

Weitere Informationen finden Sie unter [Tags verwalten](#) im AWS Resource Groups User Guide.

- API-Einheiten finden Sie unter [Untag](#) in der AWS CLI Befehlsreferenz.

update-group-query

Das folgende Codebeispiel zeigt die Verwendung `update-group-query`.

AWS CLI

Beispiel 1: Um die Abfrage für eine tagbasierte Ressourcengruppe zu aktualisieren

Im folgenden `update-group-query` Beispiel wird die Abfrage aktualisiert, die an die angegebene tagbasierte Ressourcengruppe angehängt ist.

```
aws resource-groups update-group-query \  
  --group-name tbq-WebServer \  
  --resource-query '{"Type":"TAG_FILTERS_1_0", "Query":{"ResourceTypeFilters":["AWS::EC2::Instance"], "TagFilters":[{"Key":"Name", "Values":["WebServers"]}]}'}'
```

Ausgabe:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-east-2:123456789012:group/tbq-WebServer",  
    "Name": "tbq-WebServer"  
  },  
  "ResourceQuery": {  
    "Type": "TAG_FILTERS_1_0",  
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"  
  }  
}
```

Weitere Informationen finden Sie unter [Gruppen aktualisieren](#) im AWS Resource Groups User Guide.

Beispiel 2: So aktualisieren Sie die Abfrage für eine CloudFormation stapelbasierte Ressourcengruppe

Im folgenden `update-group-query` Beispiel wird die Abfrage aktualisiert, die an die angegebene AWS CloudFormation stapelbasierte Ressourcengruppe angehängt ist.

```
aws resource-groups update-group-query \  
  --group-name cbq-CFNstackgroup \  
  --resource-query '{"Type":"TAG_FILTERS_1_0", "Query":{"ResourceTypeFilters":["AWS::CloudFormation::Stack"], "TagFilters":[{"Key":"Name", "Values":["CFNStackGroup"]}]}'}'
```

```
--resource-query '{"Type": "CLOUDFORMATION_STACK_1_0", "Query":  
"{\"ResourceTypeFilters\": [\"AWS::AllSupported\"], \"StackIdentifier\":  
\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-  
z39z-11z8-97z5-500z212zz6fz\"}"}
```

Ausgabe:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-  
CFNstackgroup",  
    "Name": "cbq-CFNstackgroup"  
  },  
  "ResourceQuery": {  
    "Type": "CLOUDFORMATION_STACK_1_0",  
    "Query": "{\"ResourceTypeFilters\": [\"AWS::AllSupported\"], \"StackIdentifier  
\": \"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-  
z39z-11z8-97z5-500z212zz6fz\"}"  
  }  
}
```

Weitere Informationen finden Sie unter [Gruppen aktualisieren](#) im AWS Resource Groups User Guide.

- API-Einheiten finden Sie [UpdateGroupQuery](#) in der AWS CLI Befehlsreferenz.

update-group

Das folgende Codebeispiel zeigt die Verwendung `update-group`.

AWS CLI

Um die Beschreibung für eine Ressourcengruppe zu aktualisieren

Im folgenden `update-group` Beispiel wird die Beschreibung für die angegebene Ressourcengruppe aktualisiert.

```
aws resource-groups update-group \  
  --group-name tbq-WebServer \  
  --description "Resource group for all web server resources."
```

Ausgabe:

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
    "Name": "tbq-WebServer"
    "Description": "Resource group for all web server resources."
  }
}
```

Weitere Informationen finden Sie unter [Gruppen aktualisieren](#) im AWS Resource Groups User Guide.

- API-Einheiten finden Sie [UpdateGroup](#) in der AWS CLI Befehlsreferenz.

API-Beispiele für das Tagging von Resource Groups mithilfe von AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Resource Groups Tagging API-Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-resources

Das folgende Codebeispiel zeigt die Verwendung `get-resources`.

AWS CLI

Um eine Liste der markierten Ressourcen abzurufen

Im folgenden `get-resources` Beispiel wird eine Liste der Ressourcen im Konto angezeigt, die mit dem angegebenen Schlüsselnamen und Wert gekennzeichnet sind.

```
aws resourcegroupstaggingapi get-resources \  
  --tag-filters Key=Environment,Values=Production \  
  --tags-per-page 100
```

Ausgabe:

```
{  
  "ResourceTagMappingList": [  
    {  
      "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-  
nvgVhaxX/template/0-7sbz2Kz0",  
      "Tags": [  
        {  
          "Key": "Environment",  
          "Value": "Production"  
        }  
      ]  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [GetResources](#) Resource Groups Tagging API Reference.

- API-Einheiten finden Sie unter [GetResources AWS CLI](#) Befehlsreferenz.

get-tag-keys

Das folgende Codebeispiel zeigt die Verwendung `get-tag-keys`.

AWS CLI

Um eine Liste aller Tag-Schlüssel zu erhalten

Im folgenden `get-tag-keys` Beispiel wird die Liste aller Tag-Schlüsselnamen abgerufen, die von Ressourcen im Konto verwendet werden.

```
aws resourcegroupstaggingapi get-tag-keys
```

Ausgabe:

```
{
```

```
"TagKeys": [
  "Environment",
  "CostCenter",
  "Department"
]
```

Weitere Informationen finden Sie unter [GetTagKeys](#) Resource Groups Tagging API Reference.

- API-Einheiten finden Sie unter [GetTagKeys AWS CLI](#) Befehlsreferenz.

get-tag-values

Das folgende Codebeispiel zeigt die Verwendung `get-tag-values`.

AWS CLI

Um eine Liste aller Tag-Werte zu erhalten

Im folgenden `get-tag-values` Beispiel werden alle Werte angezeigt, die für den angegebenen Schlüssel verwendet wurden, für alle Ressourcen in

```
aws resourcegroupstaggingapi get-tag-values \
  --key=Environment
```

Ausgabe:

```
{
  "TagValues": [
    "Alpha",
    "Gamma",
    "Production"
  ]
}
```

Weitere Informationen finden Sie unter [GetTagValues](#) Resource Groups Tagging API Reference.

- API-Einheiten finden Sie unter [GetTagValues AWS CLI](#) Befehlsreferenz.

tag-resources

Das folgende Codebeispiel zeigt die Verwendung `tag-resources`.

AWS CLI

Um ein Tag an eine Ressource anzuhängen

Im folgenden `tag-resources` Beispiel wird die angegebene Ressource mit einem Schlüsselnamen und einem Schlüsselwert versehen.

```
aws resourcegroupstaggingapi tag-resources \  
  --resource-arn-list arn:aws:s3:::MyProductionBucket \  
  --tags Environment=Production, CostCenter=1234
```

Ausgabe:

```
{  
  "FailedResourcesMap": {}  
}
```

Weitere Informationen finden Sie unter [TagResources](#) Resource Groups Tagging API Reference.

- API-Einheiten finden Sie unter [TagResources AWS CLI](#) Befehlsreferenz.

untag-resources

Das folgende Codebeispiel zeigt die Verwendung `untag-resources`.

AWS CLI

Um ein Tag aus einer Ressource zu entfernen

Im folgenden `untag-resources` Beispiel werden die angegebenen Tag-Schlüssel und alle zugehörigen Werte aus der angegebenen Ressource entfernt.

```
aws resourcegroupstaggingapi untag-resources \  
  --resource-arn-list arn:aws:s3:::awsexamplebucket \  
  --tag-keys Environment CostCenter
```

Ausgabe:

```
{  
  "FailedResourcesMap": {}  
}
```

Weitere Informationen finden Sie unter [UntagResources](#) Resource Groups Tagging API Reference.

- API Einheiten finden Sie unter [UntagResources AWS CLI](#) Befehlsreferenz.

AWS RoboMaker Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS RoboMaker.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-describe-simulation-job

Das folgende Codebeispiel zeigt die Verwendung `batch-describe-simulation-job`.

AWS CLI

Um Simulationsjobs stapelweise zu beschreiben

Im folgenden `batch-describe-simulation-job` Beispiel werden Details für die drei angegebenen Simulationsaufträge abgerufen.

Befehl:

```
aws robomaker batch-describe-simulation-job \  
--job arn:aws:robomaker:us-west-2:111111111111:simulation-job/  
sim-66bbb3gpxm8x arn:aws:robomaker:us-west-2:111111111111:simulation-job/  
sim-p0cpdrrwng2n arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-  
g8h6tg1mblgw
```


Ausgabe:

```
{
  "jobs": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x",
      "status": "Completed",
      "lastUpdatedAt": 1548959178.0,
      "failureBehavior": "Continue",
      "clientRequestToken": "6020408e-b05c-4310-9f13-4ed71c5221ed",
      "outputLocation": {
        "s3Bucket": "awsrobomakerobjecttracker-111111111-bundlesbucket-2lk584kiq1oa",
        "s3Prefix": "output"
      },
      "maxJobDurationInSeconds": 3600,
      "simulationTimeMillis": 0,
      "iamRole": "arn:aws:iam::111111111111:role/AWSRoboMakerObjectTracker-154895-SimulationJobRole-14D5ASA7PQE3A",
      "simulationApplications": [
        {
          "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
          "applicationVersion": "$LATEST",
          "launchConfig": {
            "packageName": "object_tracker_simulation",
            "launchFile": "local_training.launch",
            "environmentVariables": {
              "MARKOV_PRESET_FILE": "object_tracker.py",
              "MODEL_S3_BUCKET": "awsrobomakerobjecttracker-111111111-bundlesbucket-2lk584kiq1oa",
              "MODEL_S3_PREFIX": "model-store",
              "ROS_AWS_REGION": "us-west-2"
            }
          }
        }
      ]
    },
    {
      "tags": {},
      "vpcConfig": {
        "subnets": [
          "subnet-716dd52a",
          "subnet-43c22325",
        ]
      }
    }
  ]
}
```

```

        "subnet-3f526976"
      ],
      "securityGroups": [
        "sg-3fb40545"
      ],
      "vpcId": "vpc-99895eff",
      "assignPublicIp": true
    }
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
p0cpdrrwng2n",
    "status": "Completed",
    "lastUpdatedAt": 1548168817.0,
    "failureBehavior": "Continue",
    "clientRequestToken": "e4a23e75-f9a7-411d-835f-21881c82c58b",
    "outputLocation": {
      "s3Bucket": "awsrobomakercloudwatch-111111111111-
bundlesbucket-14e5s9jvwtmv7",
      "s3Prefix": "output"
    },
    "maxJobDurationInSeconds": 3600,
    "simulationTimeMillis": 0,
    "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G00BWTQ8YBG6",
    "robotApplications": [
      {
        "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerCloudWatch-1547663411642_NZbpqEJ3T/1547663517377",
        "applicationVersion": "$LATEST",
        "launchConfig": {
          "packageName": "cloudwatch_robot",
          "launchFile": "await_commands.launch",
          "environmentVariables": {
            "LAUNCH_ID": "1548168752173",
            "ROS_AWS_REGION": "us-west-2"
          }
        }
      }
    ]
  },
  "simulationApplications": [
    {

```

```

        "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LI6D1h6/1547663521470",
        "applicationVersion": "$LATEST",
        "launchConfig": {
            "packageName": "cloudwatch_simulation",
            "launchFile": "bookstore_turtlebot_navigation.launch",
            "environmentVariables": {
                "LAUNCH_ID": "1548168752173",
                "ROS_AWS_REGION": "us-west-2",
                "TURTLEBOT3_MODEL": "waffle_pi"
            }
        }
    },
    "tags": {},
    "vpcConfig": {
        "subnets": [
            "subnet-716dd52a",
            "subnet-43c22325",
            "subnet-3f526976"
        ],
        "securityGroups": [
            "sg-3fb40545"
        ],
        "vpcId": "vpc-99895eff",
        "assignPublicIp": true
    }
},
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tglmblgw",
    "status": "Canceled",
    "lastUpdatedAt": 1546543442.0,
    "failureBehavior": "Fail",
    "clientRequestToken": "d796bbb4-2a2c-1abc-f2a9-0d9e547d853f",
    "outputLocation": {
        "s3Bucket": "sample-bucket",
        "s3Prefix": "SimulationLog_115490482698"
    },
    "maxJobDurationInSeconds": 28800,
    "simulationTimeMillis": 0,
    "iamRole": "arn:aws:iam:111111111111:role/RoboMakerSampleTheFirst",
    "robotApplications": [

```

```

    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_robot",
        "launchFile": "rotate.launch"
      }
    },
    "simulationApplications": [
      {
        "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
        "applicationVersion": "$LATEST",
        "launchConfig": {
          "packageName": "hello_world_simulation",
          "launchFile": "empty_world.launch"
        }
      }
    ],
    "tags": {}
  }
],
"unprocessedJobs": []
}

```

- API-Einheiten finden Sie unter [BatchDescribeSimulationJob AWS CLI Befehlsreferenz](#).

cancel-simulation-job

Das folgende Codebeispiel zeigt die Verwendung `cancel-simulation-job`.

AWS CLI

Um einen Simulationsjob abzubrechen

Im folgenden `cancel-simulation-job` Beispiel wird der angegebene Simulationsjob abgebrochen.

```
aws robomaker cancel-simulation-job \
```

```
--job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- API-Einheiten finden Sie [CancelSimulationJob](#) in der AWS CLI Befehlsreferenz.

create-deployment-job

Das folgende Codebeispiel zeigt die Verwendung `create-deployment-job`.

AWS CLI

Um einen Bereitstellungsauftrag zu erstellen

In diesem Beispiel wird ein Bereitstellungsauftrag für die Flotte erstellt `MyFleet`. Es enthält eine Umgebungsvariable mit dem Namen `"ENVIRONMENT"`. Außerdem wird ein Tag mit dem Namen `„Region“` angehängt.

Befehl:

```
aws robomaker create-deployment-job --deployment-
config concurrentDeploymentPercentage=20, failureThresholdPercentage=25
--fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711 --tags Region=West --deployment-application-
configs application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
RoboMakerVoiceInteractionRobot/1546537110575, applicationVersion=1, launchConfig={environmentV
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/sim-0974h36s4v0t",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentApplicationConfigs": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerVoiceInteractionRobot/1546537110575",
      "applicationVersion": "1",
      "launchConfig": {
        "packageName": "voice_interaction_robot",
        "launchFile": "await_commands.launch",
        "environmentVariables": {
```

```

        "ENVIRONMENT": "Beta"
      }
    }
  ],
  "createdAt": 1550770236.0,
  "deploymentConfig": {
    "concurrentDeploymentPercentage": 20,
    "failureThresholdPercentage": 25
  },
  "tags": {
    "Region": "West"
  }
}

```

- API-Einheiten finden Sie [CreateDeploymentJob](#) in der AWS CLI Befehlsreferenz.

create-fleet

Das folgende Codebeispiel zeigt die Verwendung `create-fleet`.

AWS CLI

Um eine Flotte zu erstellen

In diesem Beispiel wird eine Flotte erstellt. Es fügt ein Tag mit dem Namen `Region` hinzu.

Befehl:

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

Ausgabe:

```

{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyOtherFleet/1550771394395",
  "name": "MyFleet",
  "createdAt": 1550771394.0,
  "tags": {
    "Region": "East"
  }
}

```

```
}
```

- API-Einheiten finden Sie [CreateFleet](#) in der AWS CLI Befehlsreferenz.

create-robot-application-version

Das folgende Codebeispiel zeigt die Verwendung `create-robot-application-version`.

AWS CLI

Um eine Version einer Roboter-Anwendung zu erstellen

In diesem Beispiel wird eine Robot-Anwendungsversion erstellt.

Befehl:

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "1",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-robot-application.tar.gz",
      "etag": "f8cf5526f1c6e7b3a72c3ed3f79c5493-70",
      "architecture": "ARMHF"
    }
  ],
  "robotSoftwareSuite": {
    "name": "ROS",
    "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "9986bb8d-a695-4ab4-8810-9f4a74d1aa00"
  "tags": {}
}
```

- API-Einheiten finden Sie [CreateRobotApplicationVersion](#) in der AWS CLI Befehlsreferenz.

create-robot-application

Das folgende Codebeispiel zeigt die Verwendung `create-robot-application`.

AWS CLI

Um eine Roboteranwendung zu erstellen

In diesem Beispiel wird eine Roboteranwendung erstellt.

Befehl:

```
aws robomaker create-robot-application --name MyRobotApplication --  
sources s3Bucket=my-bucket,s3Key=my-robot-application.tar.gz,architecture=X86_64 --  
robot-software-suite name=ROS,version=Kinetic
```

Ausgabe:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MyRobotApplication/1551201873931",  
  "name": "MyRobotApplication",  
  "version": "$LATEST",  
  "sources": [  
    {  
      "s3Bucket": "my-bucket",  
      "s3Key": "my-robot-application.tar.gz",  
      "architecture": "ARMHF"  
    }  
  ],  
  "robotSoftwareSuite": {  
    "name": "ROS",  
    "version": "Kinetic"  
  },  
  "lastUpdatedAt": 1551201873.0,  
  "revisionId": "1f3cb539-9239-4841-a656-d3efcffa07e1",  
  "tags": {}  
}
```

- API-Einheiten finden Sie [CreateRobotApplication](#) in der AWS CLI Befehlsreferenz.

create-robot

Das folgende Codebeispiel zeigt die Verwendung `create-robot`.

AWS CLI

Um einen Roboter zu erstellen

In diesem Beispiel wird ein Roboter erstellt. Es verwendet die ARMHF Architektur. Außerdem wird ein Tag mit dem Namen `Region` angehängt.

Befehl:

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
  "createdAt": 1550772325.0,
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "architecture": "ARMHF",
  "tags": {
    "Region": "East"
  }
}
```

- API-Einheiten finden Sie [CreateRobot](#) in der AWS CLI Befehlsreferenz.

create-simulation-application-version

Das folgende Codebeispiel zeigt die Verwendung `create-simulation-application-version`.

AWS CLI

Um eine Version einer Simulationsanwendung zu erstellen

In diesem Beispiel wird eine Roboteranwendungsversion erstellt.

Befehl:

```
aws robomaker create-simulation-application-version --  
application arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MySimulationApplication/1551203427605
```

Ausgabe:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/  
MyRobotApplication/1551203427605",  
  "name": "MyRobotApplication",  
  "version": "1",  
  "sources": [  
    {  
      "s3Bucket": "my-bucket",  
      "s3Key": "my-simulation-application.tar.gz",  
      "etag": "00d8a94ff113856688c4fce618ae0f45-94",  
      "architecture": "X86_64"  
    }  
  ],  
  "simulationSoftwareSuite": {  
    "name": "Gazebo",  
    "version": "7"  
  },  
  "robotSoftwareSuite": {  
    "name": "ROS",  
    "version": "Kinetic"  
  },  
  "renderingEngine": {  
    "name": "OGRE",  
    "version": "1.x"  
  },  
  "lastUpdatedAt": 1551203853.0,  
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",  
  "tags": {}  
}
```

- API-Einheiten finden Sie [CreateSimulationApplicationVersion](#) in der AWS CLI Befehlsreferenz.

create-simulation-application

Das folgende Codebeispiel zeigt die Verwendung `create-simulation-application`.

AWS CLI

Um eine Simulationsanwendung zu erstellen

In diesem Beispiel wird eine Simulationsanwendung erstellt.

Befehl:

```
aws robomaker create-simulation-application --name MyRobotApplication --  
sources s3Bucket=my-bucket,s3Key=my-simulation-application.tar.gz,architecture=ARMHF  
--robot-software-suite name=ROS,version=Kinetic --simulation-software-  
suite name=Gazebo,version=7 --rendering-engine name=OGRE,version=1.x
```

Ausgabe:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/  
MyRobotApplication/1551203301792",  
  "name": "MyRobotApplication",  
  "version": "$LATEST",  
  "sources": [  
    {  
      "s3Bucket": "my-bucket",  
      "s3Key": "my-simulation-application.tar.gz",  
      "architecture": "X86_64"  
    }  
  ],  
  "simulationSoftwareSuite": {  
    "name": "Gazebo",  
    "version": "7"  
  },  
  "robotSoftwareSuite": {  
    "name": "ROS",  
    "version": "Kinetic"  
  },  
  "renderingEngine": {  
    "name": "OGRE",  
    "version": "1.x"  
  },  
  "lastUpdatedAt": 1551203301.0,  
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",  
  "tags": {}  
}
```

- API-Einheiten finden Sie [CreateSimulationApplication](#) in der AWS CLI Befehlsreferenz.

create-simulation-job

Das folgende Codebeispiel zeigt die Verwendung `create-simulation-job`.

AWS CLI

Um einen Simulationsjob zu erstellen

In diesem Beispiel wird ein Simulationsjob erstellt. Es verwendet eine Roboteranwendung und eine Simulationsanwendung.

Befehl:

```
aws robomaker create-simulation-job --max-job-duration-  
in-seconds 3600 --iam-role arn:aws:iam::111111111111:role/  
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G00BWTQ8YBG6 --robot-  
applications application=arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MyRobotApplication/1551203485821,launchConfig={packageName=hello_world_robot,launchFile=rota  
--simulation-applications application=arn:aws:robomaker:us-  
west-2:111111111111:simulation-application/  
MySimulationApplication/1551203427605,launchConfig={packageName=hello_world_simulation,laun  
--tags Region=North
```

Ausgabe:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-w7m68wpr05h8",  
  "status": "Pending",  
  "lastUpdatedAt": 1551213837.0,  
  "failureBehavior": "Fail",  
  "clientRequestToken": "b283ccce-e468-43ee-8642-be76a9d69f15",  
  "maxJobDurationInSeconds": 3600,  
  "simulationTimeInMillis": 0,  
  "iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",  
  "robotApplications": [  
    {  
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-  
application/MyRobotApplication/1551203485821",  
      "applicationVersion": "$LATEST",  
      "launchConfig": {  
        "packageName": "hello_world_robot",
```

```

        "launchFile": "rotate.launch"
      }
    }
  ],
  "simulationApplications": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-
application/MySimulationApplication/1551203427605",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_simulation",
        "launchFile": "empty_world.launch"
      }
    }
  ],
  "tags": {
    "Region": "North"
  }
}

```

- API-Einheiten finden Sie [CreateSimulationJob](#) in der AWS CLI Befehlsreferenz.

delete-fleet

Das folgende Codebeispiel zeigt die Verwendung `delete-fleet`.

AWS CLI

Um eine Flotte zu löschen

In diesem Beispiel wird eine Flotte gelöscht.

Befehl:

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- API-Einheiten finden Sie [DeleteFleet](#) in der AWS CLI Befehlsreferenz.

delete-robot-application

Das folgende Codebeispiel zeigt die Verwendung `delete-robot-application`.

AWS CLI

Um eine Roboteranwendung zu löschen

In diesem Beispiel wird eine Roboteranwendung gelöscht.

Befehl:

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- API-Einheiten finden Sie [DeleteRobotApplication](#) in der AWS CLI Befehlsreferenz.

delete-robot

Das folgende Codebeispiel zeigt die Verwendung `delete-robot`.

AWS CLI

Um einen Roboter zu löschen

In diesem Beispiel wird ein Roboter gelöscht.

Befehl:

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778
```

- API-Einheiten finden Sie [DeleteRobot](#) in der AWS CLI Befehlsreferenz.

delete-simulation-application

Das folgende Codebeispiel zeigt die Verwendung `delete-simulation-application`.

AWS CLI

Um eine Simulationsanwendung zu löschen

In diesem Beispiel wird eine Simulationsanwendung gelöscht.

Befehl:

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- API-Einheiten finden Sie [DeleteSimulationApplication](#) in der AWS CLI Befehlsreferenz.

deregister-robot

Das folgende Codebeispiel zeigt die Verwendung `deregister-robot`.

AWS CLI

Um einen Roboter von einer Flotte abzumelden

In diesem Beispiel wird ein Roboter von einer Flotte abgemeldet.

Befehl:

```
aws robomaker deregister-robot --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Ausgabe:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- API-Einheiten finden Sie [DeregisterRobot](#) in der AWS CLI Befehlsreferenz.

describe-deployment-job

Das folgende Codebeispiel zeigt die Verwendung `describe-deployment-job`.

AWS CLI

Um einen Bereitstellungsjob zu beschreiben

Im folgenden `describe-deployment-job` Beispiel werden die Details zum angegebenen Bereitstellungsauftrag abgerufen.

```
aws robomaker describe-deployment-job \  
  --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-  
xl8qssl6pbcn
```

Ausgabe:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-  
xl8qssl6pbcn",  
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/  
Trek/1539894765711",  
  "status": "InProgress",  
  "deploymentConfig": {  
    "concurrentDeploymentPercentage": 20,  
    "failureThresholdPercentage": 25  
  },  
  "deploymentApplicationConfigs": [  
    {  
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-  
application/RoboMakerHelloWorldRobot/1546541208251",  
      "applicationVersion": "1",  
      "launchConfig": {  
        "packageName": "hello_world_robot",  
        "launchFile": "rotate.launch"  
      }  
    }  
  ],  
  "createdAt": 1551218369.0,  
  "robotDeploymentSummary": [  
    {  
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/  
MyRobot/1540834232469",  
      "deploymentStartTime": 1551218376.0,  
      "status": "Deploying",  
      "progressDetail": {}  
    }  
  ],  
  "tags": {}  
}
```

- API-Einheiten finden Sie unter [DescribeDeploymentJob AWS CLI-Befehlsreferenz](#).

describe-fleet

Das folgende Codebeispiel zeigt die Verwendung `describe-fleet`.

AWS CLI

Um eine Flotte zu beschreiben

Im folgenden `describe-fleet` Beispiel werden die Details für die angegebene Flotte abgerufen.

```
aws robomaker describe-fleet \
  --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907
```

Ausgabe:

```
{
  "name": "MyFleet",
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",
  "robots": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540834232469",
      "createdAt": 1540834232.0
    },
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyOtherRobot/1540829698778",
      "createdAt": 1540829698.0
    }
  ],
  "createdAt": 1539894765.0,
  "lastDeploymentStatus": "Succeeded",
  "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn",
  "lastDeploymentTime": 1551218369.0,
  "tags": {}
}
```

- API-Einheiten finden Sie unter [DescribeFleet AWS CLI-Befehlsreferenz](#).

describe-robot-application

Das folgende Codebeispiel zeigt die Verwendung `describe-robot-application`.

AWS CLI

Um eine Roboteranwendung zu beschreiben

Dieses Beispiel beschreibt eine Roboteranwendung.

Befehl:

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-robot-application.tar.gz",
      "architecture": "X86_64"
    }
  ],
  "robotSoftwareSuite": {
    "name": "ROS",
    "version": "Kinetic"
  },
  "revisionId": "e72efe0d-f44f-4333-b604-f6fa5c6bb50b",
  "lastUpdatedAt": 1551203485.0,
  "tags": {}
}
```

- API-Einheiten finden Sie [DescribeRobotApplication](#) in der AWS CLI Befehlsreferenz.

describe-robot

Das folgende Codebeispiel zeigt die Verwendung `describe-robot`.

AWS CLI

Um einen Roboter zu beschreiben

Dieses Beispiel beschreibt einen Roboter.

Befehl:

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
  "status": "Available",
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "createdAt": 1550772325.0,
  "architecture": "ARMHF",
  "tags": {
    "Region": "East"
  }
}
```

- API-Einheiten finden Sie [DescribeRobot](#) in der AWS CLI Befehlsreferenz.

describe-simulation-application

Das folgende Codebeispiel zeigt die Verwendung `describe-simulation-application`.

AWS CLI

Um eine Simulationsanwendung zu beschreiben

Dieses Beispiel beschreibt eine Simulationsanwendung.

Befehl:

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-simulation-application.tar.gz",
      "architecture": "X86_64"
    }
  ],
  "simulationSoftwareSuite": {
    "name": "Gazebo",
    "version": "7"
  },
  "robotSoftwareSuite": {
    "name": "ROS",
    "version": "Kinetic"
  },
  "renderingEngine": {
    "name": "OGRE",
    "version": "1.x"
  },
  "revisionId": "783674ab-b7b8-42d9-b01f-9373907987e5",
  "lastUpdatedAt": 1551203427.0,
  "tags": {}
}
```

- API-Einheiten finden Sie [DescribeSimulationApplication](#) in der AWS CLI Befehlsreferenz.

describe-simulation-job

Das folgende Codebeispiel zeigt die Verwendung `describe-simulation-job`.

AWS CLI

Um einen Simulationsjob zu beschreiben

Dieses Beispiel beschreibt einen Simulationsjob.

Befehl:

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6",
  "status": "Running",
  "lastUpdatedAt": 1551219349.0,
  "failureBehavior": "Continue",
  "clientRequestToken": "a19ec4b5-e50d-3591-33da-c2e593c60615",
  "outputLocation": {
    "s3Bucket": "my-output-bucket",
    "s3Prefix": "output"
  },
  "maxJobDurationInSeconds": 3600,
  "simulationTimeMillis": 0,
  "iamRole": "arn:aws:iam:111111111111:role/MySimulationRole",
  "robotApplications": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551206341136",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_robot",
        "launchFile": "rotate.launch"
      }
    }
  ],
  "simulationApplications": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551206347967",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_simulation",
        "launchFile": "empty_world.launch"
      }
    }
  ],
  "tags": {}
}
```

```
}
```

- API-Einheiten finden Sie [DescribeSimulationJob](#) in der AWS CLI Befehlsreferenz.

list-deployment-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-deployment-jobs`.

AWS CLI

Um Bereitstellungsaufträge aufzulisten

Im folgenden `list-deployment-jobs` Beispiel wird eine Liste von Bereitstellungsaufträgen abgerufen.

```
aws robomaker list-deployment-jobs
```

Ausgabe:

```
{
  "deploymentJobs": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/sim-6293szzm56rv",
      "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",
      "status": "InProgress",
      "deploymentApplicationConfigs": [
        {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-application/HelloWorldRobot/1546537110575",
          "applicationVersion": "1",
          "launchConfig": {
            "packageName": "hello_world_robot",
            "launchFile": "rotate.launch",
            "environmentVariables": {
              "ENVIRONMENT": "Desert"
            }
          }
        }
      ]
    }
  ],
}
```

```

        "deploymentConfig": {
            "concurrentDeploymentPercentage": 20,
            "failureThresholdPercentage": 25
        },
        "createdAt": 1550689373.0
    },
    {
        "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-4w4g69p25zdb",
        "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
        "status": "Pending",
        "deploymentApplicationConfigs": [
            {
                "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerHelloWorld-1544562726923_YGHM_sh5M/1544562822877",
                "applicationVersion": "1",
                "launchConfig": {
                    "packageName": "fail",
                    "launchFile": "fail"
                }
            }
        ],
        "deploymentConfig": {
            "concurrentDeploymentPercentage": 20,
            "failureThresholdPercentage": 25
        },
        "failureReason": "",
        "failureCode": "",
        "createdAt": 1544719763.0
    }
]
}

```

- API-Einheiten finden Sie [ListDeploymentJobs](#) in der AWS CLI Befehlsreferenz.

list-fleets

Das folgende Codebeispiel zeigt die Verwendung `list-fleets`.

AWS CLI

Um Flotten aufzulisten

In diesem Beispiel werden Flotten aufgeführt. Es werden maximal 20 Flotten zurückgegeben.

Befehl:

```
aws robomaker list-fleets --max-items 20
```

Ausgabe:

```
{
  "fleetDetails": [
    {
      "name": "Trek",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",
      "createdAt": 1539894765.0,
      "lastDeploymentStatus": "Failed",
      "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-4w4g69p25zdb",
      "lastDeploymentTime": 1544719763.0
    }
  ]
}
```

- API-Einheiten finden Sie [ListFleets](#) in der AWS CLI Befehlsreferenz.

list-robot-applications

Das folgende Codebeispiel zeigt die Verwendung `list-robot-applications`.

AWS CLI

Um Roboteranwendungen aufzulisten

In diesem Beispiel werden Roboteranwendungen aufgeführt. Die Ergebnisse sind auf 20 Roboteranwendungen beschränkt.

Befehl:

```
aws robomaker list-robot-applications --max-results 20
```

Ausgabe:


```
{
  "robotApplicationSummaries": [
    {
      "name": "MyRobot",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobot/1546537110575",
      "version": "$LATEST",
      "lastUpdatedAt": 1546540372.0
    },
    {
      "name": "AnotherRobot",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/AnotherRobot/1546541208251",
      "version": "$LATEST",
      "lastUpdatedAt": 1546541208.0
    },
    {
      "name": "MySuperRobot",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MySuperRobot/1547663517377",
      "version": "$LATEST",
      "lastUpdatedAt": 1547663517.0
    }
  ]
}
```

- API-Einheiten finden Sie [ListRobotApplications](#) in der AWS CLI Befehlsreferenz.

list-robots

Das folgende Codebeispiel zeigt die Verwendung `list-robots`.

AWS CLI

Um Roboter aufzulisten

In diesem Beispiel werden Roboter aufgeführt. Es werden maximal 20 Roboter zurückgegeben.

Befehl:

```
aws robomaker list-robots --max-results 20
```

Ausgabe:

```
{
  "robots": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/Robot100/1544035373264",
      "name": "Robot100",
      "status": "Available",
      "createdAt": 1544035373.0,
      "architecture": "X86_64"
    },
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/Robot101/1542146976587",
      "name": "Robot101",
      "status": "Available",
      "createdAt": 1542146976.0,
      "architecture": "X86_64"
    },
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/Robot102/1540834232469",
      "name": "Robot102",
      "fleetArn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711",
      "status": "Available",
      "createdAt": 1540834232.0,
      "architecture": "X86_64",
      "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-jb007b75gl5f",
      "lastDeploymentTime": 1550689533.0
    },
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778",
      "name": "MyRobot",
      "status": "Registered",
      "createdAt": 1540829698.0,
      "architecture": "X86_64"
    }
  ]
}
```

- API-Einheiten finden Sie [ListRobots](#) in der AWS CLI Befehlsreferenz.

list-simulation-applications

Das folgende Codebeispiel zeigt die Verwendung `list-simulation-applications`.

AWS CLI

Um Simulationsanwendungen aufzulisten

In diesem Beispiel werden Simulationsanwendungen aufgeführt. Es werden maximal 20 Simulationsanwendungen zurückgegeben.

Befehl:

```
aws robomaker list-simulation-applications --max-results 20
```

Ausgabe:

```
{
  "simulationApplicationSummaries": [
    {
      "name": "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
      "version": "$LATEST",
      "lastUpdatedAt": 1548959170.0
    },
    {
      "name": "RoboMakerHelloWorldSimulation",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/RoboMakerHelloWorldSimulation/1546541198985",
      "version": "$LATEST",
      "lastUpdatedAt": 1546541198.0
    },
    {
      "name": "RoboMakerObjectTrackerSimulation",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/RoboMakerObjectTrackerSimulation/1545846795615",
      "version": "$LATEST",
      "lastUpdatedAt": 1545847405.0
    },
    {
```

```

    "name": "RoboMakerVoiceInteractionSimulation",
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
RoboMakerVoiceInteractionSimulation/1546537100507",
    "version": "$LATEST",
    "lastUpdatedAt": 1546540352.0
  },
  {
    "name": "AWSRoboMakerCloudWatch-1547663411642_0LI6D1h6",
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LI6D1h6/1547663521470",
    "version": "$LATEST",
    "lastUpdatedAt": 1547663521.0
  },
  {
    "name": "AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-",
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-/1545848370525",
    "version": "$LATEST",
    "lastUpdatedAt": 1545848370.0
  }
]
}

```

- API-Einheiten finden Sie [ListSimulationApplications](#) in der AWS CLI Befehlsreferenz.

list-simulation-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-simulation-jobs`.

AWS CLI

Um Simulationsjobs aufzulisten

In diesem Beispiel werden Simulationsaufträge aufgeführt.

Befehl:

```
aws robomaker list-simulation-jobs
```

Ausgabe:

```
{
  "simulationJobSummaries": [
```

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x",
  "lastUpdatedAt": 1548959178.0,
  "status": "Completed",
  "simulationApplicationNames": [
    "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq"
  ],
  "robotApplicationNames": [
    null
  ]
},
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
b27c4rkrtzcv",
  "lastUpdatedAt": 1543514088.0,
  "status": "Canceled",
  "simulationApplicationNames": [
    "AWSRoboMakerPersonDetection-1543513948280_T8rHW2_lu"
  ],
  "robotApplicationNames": [
    "AWSRoboMakerPersonDetection-1543513948280_EYaMT0mYb"
  ]
},
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-51vxjbzy4q8t",
  "lastUpdatedAt": 1543508858.0,
  "status": "Canceled",
  "simulationApplicationNames": [
    "AWSRoboMakerCloudWatch-1543504747391_1FF9ZQyx6"
  ],
  "robotApplicationNames": [
    "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
  ]
},
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
kgf1fqxflqbx",
  "lastUpdatedAt": 1543504862.0,
  "status": "Completed",
  "simulationApplicationNames": [
    "AWSRoboMakerCloudWatch-1543504747391_1FF9ZQyx6"
  ]
},
```

```

    "robotApplicationNames": [
      "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
    ]
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
vw81vh061nqt",
    "lastUpdatedAt": 1543441430.0,
    "status": "Completed",
    "simulationApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341__yb_Jg961"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
    ]
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
txy5ypxmhz84",
    "lastUpdatedAt": 1543437488.0,
    "status": "Completed",
    "simulationApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341__yb_Jg961"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
    ]
  }
]
}

```

- API-Einheiten finden Sie [ListSimulationJobs](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

In diesem Beispiel werden Tags für eine AWS RoboMaker Ressource aufgelistet.

Befehl:

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```

Ausgabe:

```
{
  "tags": {
    "Region": "North",
    "Stage": "Initial"
  }
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

register-robot

Das folgende Codebeispiel zeigt die Verwendung `register-robot`.

AWS CLI

Um einen Roboter zu registrieren

In diesem Beispiel wird ein Roboter für eine Flotte registriert.

Befehl:

```
aws robomaker register-robot --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Ausgabe:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- API-Einheiten finden Sie [RegisterRobot](#) in der AWS CLI Befehlsreferenz.

restart-simulation-job

Das folgende Codebeispiel zeigt die Verwendung `restart-simulation-job`.

AWS CLI

Um eine Simulation neu zu starten

In diesem Beispiel wird eine Simulation neu gestartet.

Befehl:

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- API-Einheiten finden Sie [RestartSimulationJob](#) in der AWS CLI Befehlsreferenz.

sync-deployment-job

Das folgende Codebeispiel zeigt die Verwendung `sync-deployment-job`.

AWS CLI

Um einen Bereitstellungsauftrag zu synchronisieren

In diesem Beispiel wird ein Bereitstellungsauftrag synchronisiert.

Befehl:

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-09ccxs3tlfms",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentConfig": {
    "concurrentDeploymentPercentage": 20,
    "failureThresholdPercentage": 25
  }
}
```



```
},
  "deploymentApplicationConfigs": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1546541208251",
      "applicationVersion": "1",
      "launchConfig": {
        "packageName": "hello_world_simulation",
        "launchFile": "empty_world.launch"
      }
    }
  ],
  "createdAt": 1551286954.0
}
```

- API-Einheiten finden Sie [SyncDeploymentJob](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um eine Ressource zu taggen

In diesem Beispiel wird eine Ressource markiert. Es fügt zwei Tags hinzu: Region und Phase.

Befehl:

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung einer Ressource aufzuheben

In diesem Beispiel wird ein Tag aus einer Ressource entfernt. Es entfernt das Region-Tag.

Befehl:

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-robot-application

Das folgende Codebeispiel zeigt die Verwendung `update-robot-application`.

AWS CLI

Um eine Roboteranwendung zu aktualisieren

In diesem Beispiel wird eine Roboteranwendung aktualisiert.

Befehl:

```
aws robomaker update-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821 --sources s3Bucket=my-bucket,s3Key=my-robot-application.tar.gz,architecture=X86_64 --robot-software-suite name=ROS,version=Kinetic
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-robot-application.tar.gz",
      "architecture": "X86_64"
    }
  ],
  "robotSoftwareSuite": {
    "name": "ROS",
    "version": "Kinetic"
  },
  "lastUpdatedAt": 1551287993.0,
}
```

```
"revisionId": "20b5e331-24fd-4504-8b8c-531afe5f4c94"
}
```

- API-Einheiten finden Sie [UpdateRobotApplication](#) in der AWS CLI Befehlsreferenz.

update-simulation-application

Das folgende Codebeispiel zeigt die Verwendung `update-simulation-application`.

AWS CLI

Um eine Simulationsanwendung zu aktualisieren

In diesem Beispiel wird eine Simulationsanwendung aktualisiert.

Befehl:

```
aws robomaker update-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605 --sources s3Bucket=my-bucket,s3Key=my-simulation-application.tar.gz,architecture=X86_64 --robot-software-suite name=ROS,version=Kinetic --simulation-software-suite name=Gazebo,version=7 --rendering-engine name=OGRE,version=1.x
```

Ausgabe:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-simulation-application.tar.gz",
      "architecture": "X86_64"
    }
  ],
  "simulationSoftwareSuite": {
    "name": "Gazebo",
    "version": "7"
  },
  "robotSoftwareSuite": {
```

```
    "name": "ROS",
    "version": "Kinetic"
  },
  "renderingEngine": {
    "name": "OGRE",
    "version": "1.x"
  },
  "lastUpdatedAt": 1551289361.0,
  "revisionId": "4a22cb5d-93c5-4cef-9311-52bdd119b79e"
}
```

- API-Einheiten finden Sie [UpdateSimulationApplication](#) in der AWS CLI Befehlsreferenz.

Route 53-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Route 53 Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

change-resource-record-sets

Das folgende Codebeispiel zeigt die Verwendung `change-resource-record-sets`.

AWS CLI

Um einen Ressourcendatensatz zu erstellen, zu aktualisieren oder zu löschen

Der folgende `change-resource-record-sets` Befehl erstellt einen Ressourcendatensatz unter Verwendung der `hosted-zone-id` `Z1R8UBAEXAMPLE` und der JSON-formatierten Konfiguration in der Datei: `C:\awscli\route53\change-resource-record-sets.json`

```
aws route53 change-resource-record-sets --hosted-zone-id Z1R8UBAEXAMPLE --change-batch file://C:\awscli\route53\change-resource-record-sets.json
```

Weitere Informationen finden Sie POST ChangeResourceRecordSets in der Amazon Route 53 API 53-Referenz.

Die Konfiguration in der JSON Datei hängt von der Art des Ressourceneintragssatzes ab, den Sie erstellen möchten:

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover Alias

Grundlegende Syntax:

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ]
      }
    },
    {...}
  ]
}
```

Gewichtete Syntax:

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
```

```

"ResourceRecordSet": {
  "Name": "DNS domain name",
  "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
  "SetIdentifier": "unique description for this resource record set",
  "Weight": value between 0 and 255,
  "TTL": time to live in seconds,
  "ResourceRecords": [
    {
      "Value": "applicable value for the record type"
    },
    {...}
  ],
  "HealthCheckId": "optional ID of an Amazon Route 53 health check"
}
},
{...}
]
}

```

Alias-Syntax:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
          S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
          bucket, Elastic Load Balancing load balancer, or another resource record set in
          this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

Gewichtete Alias-Syntax:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
bucket, Elastic Load Balancing load balancer, or another resource record set in
this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

Latenz-Syntax:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          }
        ]
      }
    }
  ]
}

```

```

    },
    {...}
  ],
  "HealthCheckId": "optional ID of an Amazon Route 53 health check"
}
},
{...}
]
}

```

Latenz-Alias-Syntax:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
          S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
          bucket, Elastic Load Balancing load balancer, or another resource record set in
          this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

Failover-Syntax:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {

```



```

"Action": "CREATE"|"DELETE"|"UPSERT",
"ResourceRecordSet": {
  "Name": "DNS domain name",
  "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
  "SetIdentifier": "unique description for this resource record set",
  "Failover": "PRIMARY" | "SECONDARY",
  "TTL": time to live in seconds,
  "ResourceRecords": [
    {
      "Value": "applicable value for the record type"
    },
    {...}
  ],
  "HealthCheckId": "ID of an Amazon Route 53 health check"
}
},
{...}
]
}

```

Syntax des Failover-Alias:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Failover": "PRIMARY" | "SECONDARY",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
bucket, Elastic Load Balancing load balancer, or another resource record set in
this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

```
{...}  
]  
}
```

- API-Einheiten finden Sie [ChangeResourceRecordSets](#) in der AWS CLI Befehlsreferenz.

change-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `change-tags-for-resource`.

AWS CLI

Mit dem folgenden Befehl wird einer durch die ID angegebenen Healthcheck-Ressource ein Tag mit dem Namen `owner` hinzugefügt:

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner,Value=myboss
```

Mit dem folgenden Befehl wird ein Tag mit dem Namen `owner` aus einer Ressource in der gehosteten Zone entfernt, die mit der ID angegeben ist:

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-id Z1523434445 --remove-tag-keys owner
```

- API-Einheiten finden Sie [ChangeTagsForResource](#) in der AWS CLI Befehlsreferenz.

create-health-check

Das folgende Codebeispiel zeigt die Verwendung `create-health-check`.

AWS CLI

Um einen Gesundheitscheck zu erstellen

Mit dem folgenden `create-health-check` Befehl wird mithilfe der Anruferreferenz `2014-04-01-18:47` und der JSON-formatierten Konfiguration in der Datei eine Integritätsprüfung erstellt: `C:\awscli\route53\create-health-check.json`

```
aws route53 create-health-check --caller-reference 2014-04-01-18:47 --health-check-config file://C:\awscli\route53\create-health-check.json
```

JSONSyntax:

```
{
  "IPAddress": "IP address of the endpoint to check",
  "Port": port on the endpoint to check--required when Type is "TCP",
  "Type": "HTTP"|"HTTPS"|"HTTP_STR_MATCH"|"HTTPS_STR_MATCH"|"TCP",
  "ResourcePath": "path of the file that you want Amazon Route 53 to request--all
Types except TCP",
  "FullyQualifiedDomainName": "domain name of the endpoint to check--all Types
except TCP",
  "SearchString": "if Type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string to
search for in the response body from the specified resource",
  "RequestInterval": 10 | 30,
  "FailureThreshold": integer between 1 and 10
}
```

Verwenden Sie den `change-resource-record-sets` Befehl, um die Integritätsprüfung zu einem Route 53 53-Ressourcendatensatz hinzuzufügen.

Weitere Informationen finden Sie unter Amazon Route 53 Health Checks and DNS Failover im Amazon Route 53 Developer Guide.

- APIEinzelheiten finden Sie unter [CreateHealthCheck AWS CLI](#) Befehlsreferenz.

create-hosted-zone

Das folgende Codebeispiel zeigt die Verwendung `create-hosted-zone`.

AWS CLI

So erstellen Sie eine gehostete Zone

Der folgende `create-hosted-zone` Befehl fügt eine gehostete Zone hinzu, die `example.com` mithilfe der Anruferreferenz `2014-04-01-18:47` benannt wird. Der optionale Kommentar enthält ein Leerzeichen und muss daher in Anführungszeichen gesetzt werden:

```
aws route53 create-hosted-zone --name example.com --caller-
reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

Weitere Informationen finden Sie unter Working with Hosted Zones im Amazon Route 53 53-Entwicklerhandbuch.

- APIEinzelheiten finden Sie [CreateHostedZone](#) in der AWS CLI Befehlsreferenz.

delete-health-check

Das folgende Codebeispiel zeigt die Verwendung `delete-health-check`.

AWS CLI

Um einen Gesundheitscheck zu löschen

Der folgende `delete-health-check` Befehl löscht die Integritätsprüfung mit einem `health-check-id` von `e75b48d9-547a-4c3d-88a5-ae4002397608`:

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-ae4002397608
```

- API-Einheiten finden Sie [DeleteHealthCheck](#) in der AWS CLI Befehlsreferenz.

delete-hosted-zone

Das folgende Codebeispiel zeigt die Verwendung `delete-hosted-zone`.

AWS CLI

Um eine gehostete Zone zu löschen

Der folgende `delete-hosted-zone` Befehl löscht die Hosting-Zone mit dem Wert `id` von `Z36KTIQEXAMPLE`:

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- API-Einheiten finden Sie [DeleteHostedZone](#) in der AWS CLI Befehlsreferenz.

get-change

Das folgende Codebeispiel zeigt die Verwendung `get-change`.

AWS CLI

Um den Status einer Änderung an Ressourcendatensätzen abzurufen

Mit dem folgenden `get-change` Befehl werden der Status und weitere Informationen zu der `change-resource-record-sets` Anfrage abgerufen, die den Wert `Id` von `hat/change/CWPIK4URU2I5S`:

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- API-Einheiten finden Sie [GetChange](#) in der AWS CLI Befehlsreferenz.

get-health-check

Das folgende Codebeispiel zeigt die Verwendung `get-health-check`.

AWS CLI

Um Informationen über einen Gesundheitscheck zu erhalten

Mit dem folgenden `get-health-check` Befehl werden Informationen zur Integritätsprüfung abgerufen, die einen Wert `health-check-id` von `hat02ec8401-9879-4259-91fa-04e66d094674`:

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- API-Einheiten finden Sie [GetHealthCheck](#) in der AWS CLI Befehlsreferenz.

get-hosted-zone

Das folgende Codebeispiel zeigt die Verwendung `get-hosted-zone`.

AWS CLI

Um Informationen über eine gehostete Zone abzurufen

Mit dem folgenden `get-hosted-zone` Befehl werden Informationen über die Hosting-Zone mit dem Wert `id` von `Z1R8UBAEXAMPLE` abgerufen:

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- API-Einheiten finden Sie [GetHostedZone](#) in der AWS CLI Befehlsreferenz.

list-health-checks

Das folgende Codebeispiel zeigt die Verwendung `list-health-checks`.

AWS CLI

Um die mit dem AWS Girokonto verknüpften Gesundheitschecks aufzulisten

Der folgende `list-health-checks` Befehl listet detaillierte Informationen zu den ersten 100 Zustandsprüfungen auf, die mit dem aktuellen AWS Konto verknüpft sind. :

```
aws route53 list-health-checks
```

Wenn Sie mehr als 100 Zustandsprüfungen haben oder wenn Sie diese in Gruppen mit weniger als 100 auflisten möchten, geben Sie den `--max-items` Parameter an. Um beispielsweise die Zustandsprüfungen einzeln aufzulisten, verwenden Sie den folgenden Befehl:

```
aws route53 list-health-checks --max-items 1
```

Um die nächste Integritätsprüfung anzuzeigen, nehmen Sie den Wert von `NextToken` aus der Antwort auf den vorherigen Befehl und fügen ihn in den `--starting-token` Parameter ein, zum Beispiel:

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- API-Einheiten finden Sie [ListHealthChecks](#) in der AWS CLI Befehlsreferenz.

list-hosted-zones-by-name

Das folgende Codebeispiel zeigt die Verwendung `list-hosted-zones-by-name`.

AWS CLI

Der folgende Befehl listet bis zu 100 gehostete Zonen auf, sortiert nach Domainnamen:

```
aws route53 list-hosted-zones-by-name
```

Ausgabe:

```
{
  "HostedZones": [
    {
      "ResourceRecordSetCount": 2,
      "CallerReference": "test20150527-2",

```

```

    "Config": {
      "Comment": "test2",
      "PrivateZone": false
    },
    "Id": "/hostedzone/Z119WBBTVP5WFX",
    "Name": "2.example.com."
  },
  {
    "ResourceRecordSetCount": 2,
    "CallerReference": "test20150527-1",
    "Config": {
      "Comment": "test",
      "PrivateZone": false
    },
    "Id": "/hostedzone/Z3P5QSUBK4POTI",
    "Name": "www.example.com."
  }
],
"IsTruncated": false,
"MaxItems": "100"
}

```

Der folgende Befehl listet die Hosting-Zonen nach Namen geordnet auf, beginnend mit `www.example.com`:

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

Ausgabe:

```

{
  "HostedZones": [
    {
      "ResourceRecordSetCount": 2,
      "CallerReference": "mwunderl20150527-1",
      "Config": {
        "Comment": "test",
        "PrivateZone": false
      },
      "Id": "/hostedzone/Z3P5QSUBK4POTI",
      "Name": "www.example.com."
    }
  ],
  "DNSName": "www.example.com",

```

```
"IsTruncated": false,  
"MaxItems": "100"  
}
```

- API-Einheiten finden Sie [ListHostedZonesByName](#) in der AWS CLI Befehlsreferenz.

list-hosted-zones

Das folgende Codebeispiel zeigt die Verwendung `list-hosted-zones`.

AWS CLI

Um die Hosting-Zonen aufzulisten, die dem aktuellen AWS Konto zugeordnet sind

Der folgende `list-hosted-zones` Befehl listet zusammenfassende Informationen zu den ersten 100 Hostzonen auf, die dem aktuellen AWS Konto zugeordnet sind. :

```
aws route53 list-hosted-zones
```

Wenn Sie mehr als 100 gehostete Zonen haben oder wenn Sie sie in Gruppen von weniger als 100 auflisten möchten, fügen Sie den Parameter `--max-items` ein. Um zum Beispiel eine gehostete Zone nach der anderen aufzulisten, verwenden Sie den folgenden Befehl:

```
aws route53 list-hosted-zones --max-items 1
```

Um Informationen über die nächste gehostete Zone anzuzeigen, übernehmen Sie den Wert von `NextToken` aus der Antwort auf den vorherigen Befehl und fügen ihn in den Parameter `--starting-token` ein, zum Beispiel:

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- API-Einheiten finden Sie [ListHostedZones](#) in der AWS CLI Befehlsreferenz.

list-query-logging-configs

Das folgende Codebeispiel zeigt die Verwendung `list-query-logging-configs`.

AWS CLI

Um Konfigurationen für die Abfrageprotokollierung aufzulisten

Im folgenden `list-query-logging-configs` Beispiel werden Informationen zu den ersten 100 Konfigurationen für die Abfrageprotokollierung in Ihrem AWS Konto für die gehostete Zone aufgeführt `Z10X3WQEXAMPLE`.

```
aws route53 list-query-logging-configs \
  --hosted-zone-id Z10X3WQEXAMPLE
```

Ausgabe:

```
{
  "QueryLoggingConfigs": [
    {
      "Id": "964ff34e-ae03-4f06-80a2-9683cexample",
      "HostedZoneId": "Z10X3WQEXAMPLE",
      "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/route53/example.com:*"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Logging DNS queries](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [ListQueryLoggingConfigs](#) in der AWS CLI Befehlsreferenz.

list-resource-record-sets

Das folgende Codebeispiel zeigt die Verwendung `list-resource-record-sets`.

AWS CLI

Um die Ressourcendatensätze in einer gehosteten Zone aufzulisten

Der folgende `list-resource-record-sets` Befehl listet zusammenfassende Informationen zu den ersten 100 Ressourcendatensätzen in einer angegebenen Hostzone auf. :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

Wenn die gehostete Zone mehr als 100 Ressourcendatensätze enthält oder wenn Sie sie in Gruppen mit weniger als 100 auflisten möchten, geben Sie den `--max-items` Parameter

an. Verwenden Sie beispielsweise den folgenden Befehl, um Ressourcendatensätze einzeln aufzulisten:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

Um Informationen über den nächsten Ressourceneintrag in der Hosting-Zone anzuzeigen, nehmen Sie den Wert von `NextToken` aus der Antwort auf den vorherigen Befehl und fügen ihn in den `--starting-token` Parameter ein, zum Beispiel:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1  
--starting-token Z3M3LMPEXAMPLE
```

Um alle Ressourcendatensätze mit einem bestimmten Namen anzuzeigen, verwenden Sie den `--query` Parameter, um sie herauszufiltern. Beispielsweise:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --  
query "ResourceRecordSets[?Name == 'example.domain.']"
```

- API-Einheiten finden Sie [ListResourceRecordSets](#) unter AWS CLI Befehlsreferenz.

Beispiele für die Route 53-Domainregistrierung mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe der Domänenregistrierung AWS Command Line Interface mit Route 53 Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

check-domain-availability

Das folgende Codebeispiel zeigt die Verwendung `check-domain-availability`.

AWS CLI

Um festzustellen, ob Sie einen Domainnamen mit Route 53 registrieren können

Der folgende `check-domain-availability` Befehl gibt Informationen darüber zurück, ob der Domainname für die Registrierung über Route 53 verfügbar `example.com` ist.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains check-domain-availability \
  --region us-east-1 \
  --domain-name example.com
```

Ausgabe:

```
{
  "Availability": "UNAVAILABLE"
}
```

Route 53 unterstützt eine große Anzahl von Top-Level-Domains (TLDs), z. B. `.com` und `.jp`, aber wir unterstützen nicht alle verfügbaren TLDs Domains. Wenn Sie die Verfügbarkeit einer Domain überprüfen und Route 53 das nicht unterstützt TLD, wird die folgende Meldung `check-domain-availability` zurückgegeben.

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability
operation: <top-level domain> tld is not supported.
```

Eine Liste der Domains TLDs, die Sie bei der Registrierung einer Domain bei Route 53 verwenden können, finden Sie unter [Domains, die Sie bei Amazon Route 53 registrieren können](#) im Amazon Route 53 53-Entwicklerhandbuch. Weitere Informationen zur Registrierung von Domains bei Amazon Route 53 finden Sie unter [Registrierung einer neuen Domain](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API Einheiten finden Sie [CheckDomainAvailability](#) in der AWS CLI Befehlsreferenz.

check-domain-transferability

Das folgende Codebeispiel zeigt die Verwendung `check-domain-transferability`.

AWS CLI

Um festzustellen, ob eine Domain auf Route 53 übertragen werden kann

Der folgende `check-domain-transferability` Befehl gibt Informationen darüber zurück, ob Sie den Domainnamen `example.com` auf Route 53 übertragen können.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains check-domain-transferability \
  --region us-east-1 \
  --domain-name example.com
```

Ausgabe:

```
{
  "Transferability": {
    "Transferable": "UNTRANSFERABLE"
  }
}
```

Weitere Informationen finden Sie unter [Übertragung der Registrierung für eine Domain auf Amazon Route 53](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [CheckDomainTransferability](#) in der AWS CLI Befehlsreferenz.

delete-tags-for-domain

Das folgende Codebeispiel zeigt die Verwendung `delete-tags-for-domain`.

AWS CLI

Um Tags für eine Domain zu löschen

Der folgende `delete-tags-for-domain` Befehl löscht drei Tags aus der angegebenen Domäne. Beachten Sie, dass Sie nur den Tag-Schlüssel angeben, nicht den Tag-Wert.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains delete-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --tags-to-delete accounting-key hr-key engineering-key
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Um zu bestätigen, dass die Tags gelöscht wurden, können Sie ausführen [list-tags-for-domain](#). Weitere Informationen finden Sie unter [Tagging Amazon Route 53 53-Ressourcen](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DeleteTagsForDomain AWS CLI Befehlsreferenz](#).

disable-domain-auto-renew

Das folgende Codebeispiel zeigt die Verwendung `disable-domain-auto-renew`.

AWS CLI

Um die automatische Verlängerung einer Domain zu deaktivieren

Mit dem folgenden `disable-domain-auto-renew` Befehl wird Route 53 so konfiguriert, dass die Domain nicht automatisch erneuert wird, `example.com` bevor die Registrierung für die Domain abläuft.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains disable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Um zu bestätigen, dass die Einstellung geändert wurde, können Sie ausführen [get-domain-detail](#). Wenn die automatische Verlängerung deaktiviert ist, `AutoRenew` ist der Wert von `False`. Weitere Informationen zur automatischen Verlängerung finden Sie unter Erneuern der Registrierung für

eine Domain < <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html> im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [DisableDomainAutoRenew](#) in der AWS CLI Befehlsreferenz.

disable-domain-transfer-lock

Das folgende Codebeispiel zeigt die Verwendung `disable-domain-transfer-lock`.

AWS CLI

Um die Transfersperre für eine Domain zu deaktivieren

Mit dem folgenden `disable-domain-transfer-lock` Befehl wird die Übertragungssperre für die Domain aufgehoben, `example.com` sodass die Domain an einen anderen Registrar übertragen werden kann. Dieser Befehl ändert den `clientTransferProhibited` Status.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains disable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

Ausgabe:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Um zu bestätigen, dass die Übertragungssperre geändert wurde, können Sie ausführen [get-domain-detail](#). Wenn die Übertragungssperre deaktiviert ist, schließt der Wert von `StatusList` nicht ein `clientTransferProhibited`.

Weitere Informationen zum Übertragungsprozess finden Sie unter [Übertragung einer Domain von Amazon Route 53 zu einem anderen Registrar](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [DisableDomainTransferLock](#) in der AWS CLI Befehlsreferenz.

enable-domain-auto-renew

Das folgende Codebeispiel zeigt die Verwendung `enable-domain-auto-renew`.

AWS CLI

Um die automatische Verlängerung einer Domain zu aktivieren

Mit dem folgenden `enable-domain-auto-renew` Befehl wird Route 53 so konfiguriert, dass die Domain automatisch erneuert wird, `example.com` bevor die Registrierung für die Domain abläuft.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains enable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um zu bestätigen, dass die Einstellung geändert wurde, können Sie ausführen [get-domain-detail](#). Wenn die automatische Verlängerung aktiviert ist, `AutoRenew` ist der Wert von `True`.

Weitere Informationen zur automatischen Verlängerung finden Sie unter Erneuern der Registrierung für eine Domain < <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html> im Amazon Route 53 Developer Guide.

- API Einheiten finden Sie [EnableDomainAutoRenew](#) in der AWS CLI Befehlsreferenz.

`enable-domain-transfer-lock`

Das folgende Codebeispiel zeigt die Verwendung `enable-domain-transfer-lock`.

AWS CLI

Um die Transfersperre für eine Domain zu aktivieren

Der folgende `enable-domain-transfer-lock` Befehl sperrt die angegebene Domain, sodass sie nicht an einen anderen Registrar übertragen werden kann. Dieser Befehl ändert den `clientTransferProhibited` Status.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains enable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

Ausgabe:

```
{
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

Um zu bestätigen, dass die Übertragungssperre geändert wurde, können Sie ausführen [get-domain-detail](#). Wenn die Übertragungssperre aktiviert ist, enthält der Wert von `StatusList` `IncludesClientTransferProhibited`.

Weitere Informationen zum Übertragungsprozess finden Sie unter [Übertragung einer Domain von Amazon Route 53 zu einem anderen Registrar](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [EnableDomainTransferLock](#) in der AWS CLI Befehlsreferenz.

get-contact-reachability-status

Das folgende Codebeispiel zeigt die Verwendung `get-contact-reachability-status`.

AWS CLI

Um festzustellen, ob der Kontakt des Registranten auf eine Bestätigungs-E-Mail geantwortet hat

Der folgende `get-contact-reachability-status` Befehl gibt Informationen darüber zurück, ob der Kontakt des Registranten für die angegebene Domain auf eine Bestätigungs-E-Mail geantwortet hat.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains get-contact-reachability-status \
  --region us-east-1 \
  --domain-name example.com
```

Ausgabe:

```
{
  "domainName": "example.com",
  "status": "DONE"
}
```


Weitere Informationen finden Sie unter [Erneutes Senden von Autorisierungs- und Bestätigungs-E-Mails](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [GetContactReachabilityStatus AWS CLI Befehlsreferenz](#).

get-domain-detail

Das folgende Codebeispiel zeigt die Verwendung `get-domain-detail`.

AWS CLI

Um detaillierte Informationen zu einer bestimmten Domain zu erhalten

Der folgende `get-domain-detail` Befehl zeigt detaillierte Informationen über die angegebene Domäne an.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains get-domain-detail \
  --region us-east-1 \
  --domain-name example.com
```

Ausgabe:

```
{
  "DomainName": "example.com",
  "Nameservers": [
    {
      "Name": "ns-2048.awsdns-64.com",
      "GlueIps": []
    },
    {
      "Name": "ns-2049.awsdns-65.net",
      "GlueIps": []
    },
    {
      "Name": "ns-2050.awsdns-66.org",
      "GlueIps": []
    },
    {
      "Name": "ns-2051.awsdns-67.co.uk",
      "GlueIps": []
    }
  ]
}
```

```
    }
  ],
  "AutoRenew": true,
  "AdminContact": {
    "FirstName": "Saanvi",
    "LastName": "Sarkar",
    "ContactType": "COMPANY",
    "OrganizationName": "Example",
    "AddressLine1": "123 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "ssarkar@example.com",
    "ExtraParams": []
  },
  "RegistrantContact": {
    "FirstName": "Alejandro",
    "LastName": "Rosalez",
    "ContactType": "COMPANY",
    "OrganizationName": "Example",
    "AddressLine1": "123 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "arosalez@example.com",
    "ExtraParams": []
  },
  "TechContact": {
    "FirstName": "Wang",
    "LastName": "Xiulan",
    "ContactType": "COMPANY",
    "OrganizationName": "Example",
    "AddressLine1": "123 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "wxiulan@example.com",
    "ExtraParams": []
  }
}
```

```
},
  "AdminPrivacy": true,
  "RegistrantPrivacy": true,
  "TechPrivacy": true,
  "RegistrarName": "Amazon Registrar, Inc.",
  "WhoIsServer": "whois.registrar.amazon.com",
  "RegistrarUrl": "http://registrar.amazon.com",
  "AbuseContactEmail": "abuse@registrar.amazon.com",
  "AbuseContactPhone": "+1.2062661000",
  "CreationDate": 1444934889.601,
  "ExpirationDate": 1602787689.0,
  "StatusList": [
    "clientTransferProhibited"
  ]
}
```

- API-Einheiten finden Sie unter [GetDomainDetail AWS CLI Befehlsreferenz](#).

get-domain-suggestions

Das folgende Codebeispiel zeigt die Verwendung `get-domain-suggestions`.

AWS CLI

Um eine Liste der vorgeschlagenen Domainnamen zu erhalten

Der folgende `get-domain-suggestions` Befehl zeigt eine Liste mit vorgeschlagenen Domainnamen an, die auf dem Domainnamen basieren `example.com`. Die Antwort enthält nur Domainnamen, die verfügbar sind. Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains get-domain-suggestions \
  --region us-east-1 \
  --domain-name example.com \
  --suggestion-count 10 \
  --only-available
```

Ausgabe:

```
{
```

```
"SuggestionsList": [  
  {  
    "DomainName": "egzaampal.com",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "examplelaw.com",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "examplehouse.net",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "homeexample.net",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "examplelist.com",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "examplenews.net",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "officeexample.com",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "exampleworld.com",  
    "Availability": "AVAILABLE"  
  },  
  {  
    "DomainName": "exampleart.com",  
    "Availability": "AVAILABLE"  
  }  
]  
}
```

- API-Einheiten finden Sie unter [GetDomainSuggestions AWS CLI-Befehlsreferenz](#).

get-operation-detail

Das folgende Codebeispiel zeigt die Verwendung `get-operation-detail`.

AWS CLI

Um den aktuellen Status einer Operation abzurufen

Einige Domainregistrierungsvorgänge werden asynchron ausgeführt und geben eine Antwort zurück, bevor sie abgeschlossen sind. Diese Operationen geben eine Vorgangs-ID zurück, mit der Sie den aktuellen Status abrufen können. Der folgende `get-operation-detail` Befehl gibt den Status der angegebenen Operation zurück.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains get-operation-detail \  
  --region us-east-1 \  
  --operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

Ausgabe:

```
{  
  "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",  
  "Status": "SUCCESSFUL",  
  "DomainName": "example.com",  
  "Type": "DOMAIN_LOCK",  
  "SubmittedDate": 1573749367.864  
}
```

- API-Einheiten finden Sie unter [GetOperationDetail AWS CLI Befehlsreferenz](#).

list-domains

Das folgende Codebeispiel zeigt die Verwendung `list-domains`.

AWS CLI

Um die Domains aufzulisten, die mit dem AWS Girokonto registriert sind

Der folgende `list-domains` Befehl listet zusammenfassende Informationen zu den Domänen auf, die mit dem aktuellen AWS Konto registriert sind.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf `us-east-1` eingestellt ist, können Sie den `region` Parameter weglassen.

```
aws route53domains list-domains
  --region us-east-1
```

Ausgabe:

```
{
  "Domains": [
    {
      "DomainName": "example.com",
      "AutoRenew": true,
      "TransferLock": true,
      "Expiry": 1602712345.0
    },
    {
      "DomainName": "example.net",
      "AutoRenew": true,
      "TransferLock": true,
      "Expiry": 1602723456.0
    },
    {
      "DomainName": "example.org",
      "AutoRenew": true,
      "TransferLock": true,
      "Expiry": 1602734567.0
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListDomains AWS CLI Befehlsreferenz](#).

list-operations

Das folgende Codebeispiel zeigt die Verwendung `list-operations`.

AWS CLI

Um den Status von Vorgängen aufzulisten, die eine Vorgangs-ID zurückgeben

Einige Domainregistrierungsvorgänge werden asynchron ausgeführt und geben eine Antwort zurück, bevor sie abgeschlossen sind. Diese Operationen geben eine Vorgangs-ID zurück, mit der Sie den aktuellen Status abrufen können. Der folgende `list-operations` Befehl listet zusammenfassende Informationen, einschließlich des Status, zu den aktuellen Domänenregistrierungsvorgängen auf.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains list-operations
  --region us-east-1
```

Ausgabe:

```
{
  "Operations": [
    {
      "OperationId": "aab9822f-1da0-4bf3-8a15-fd4e0example",
      "Status": "SUCCESSFUL",
      "Type": "DOMAIN_LOCK",
      "SubmittedDate": 1455321739.986
    },
    {
      "OperationId": "c24379ed-76be-42f8-bdad-9379bexample",
      "Status": "SUCCESSFUL",
      "Type": "UPDATE_NAMESERVER",
      "SubmittedDate": 1468960475.109
    },
    {
      "OperationId": "f47e1297-ef9e-4c2b-ae1e-a5fcbexample",
      "Status": "SUCCESSFUL",
      "Type": "RENEW_DOMAIN",
      "SubmittedDate": 1473561835.943
    },
    {
      "OperationId": "75584f23-b15f-459e-aed7-dc6f5example",
      "Status": "SUCCESSFUL",
      "Type": "UPDATE_DOMAIN_CONTACT",
      "SubmittedDate": 1547501003.41
    }
  ]
}
```

Die Ausgabe umfasst alle Operationen, die eine Vorgangs-ID zurückgeben und die Sie für alle Domains ausgeführt haben, die Sie jemals mit dem AWS Girokonto registriert haben. Wenn Sie nur die Operationen abrufen möchten, die Sie nach einem bestimmten Datum eingereicht haben, können Sie den `submitted-since` Parameter einbeziehen und ein Datum im Unix-Format und in Coordinated Universal Time (UTC) angeben. Der folgende Befehl ruft den Status aller Operationen ab, die UTC am 1. Januar 2020 nach 12:00 Uhr übermittelt wurden.

```
aws route53domains list-operations \  
  --submitted-since 1577836800
```

- API-Einheiten finden Sie [ListOperations](#) unter AWS CLI Befehlsreferenz.

list-tags-for-domain

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-domain`.

AWS CLI

Um Tags für eine Domain aufzulisten

Der folgende `list-tags-for-domain` Befehl listet die Tags auf, die derzeit der angegebenen Domäne zugeordnet sind.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains list-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

Ausgabe:

```
{  
  "TagList": [  
    {  
      "Key": "key1",  
      "Value": "value1"  
    },  
    {  
      "Key": "key2",  
      "Value": "value2"  
    }  
  ]  
}
```



```
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Tagging Amazon Route 53 53-Ressourcen](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [ListTagsForDomain AWS CLI-Befehlsreferenz](#).

register-domain

Das folgende Codebeispiel zeigt die Verwendung `register-domain`.

AWS CLI

Um eine Domain zu registrieren

Der folgende `register-domain` Befehl registriert eine Domäne und ruft alle Parameterwerte aus einer mit JSON-formatierten Datei ab.

Dieser Befehl wird nur in der Region ausgeführt. `us-east-1` Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains register-domain \  
  --region us-east-1 \  
  --cli-input-json file://register-domain.json
```

Inhalt von `register-domain.json`:

```
{  
  "DomainName": "example.com",  
  "DurationInYears": 1,  
  "AutoRenew": true,  
  "AdminContact": {  
    "FirstName": "Martha",  
    "LastName": "Rivera",  
    "ContactType": "PERSON",  
    "OrganizationName": "Example",  
    "AddressLine1": "1 Main Street",  
    "City": "Anytown",  
    "State": "WA",  
    "CountryCode": "US",
```

```
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "mrivera@example.com"
  },
  "RegistrantContact": {
    "FirstName": "Li",
    "LastName": "Juan",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "ljuan@example.com"
  },
  "TechContact": {
    "FirstName": "Mateo",
    "LastName": "Jackson",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "mjackson@example.com"
  },
  "PrivacyProtectAdminContact": true,
  "PrivacyProtectRegistrantContact": true,
  "PrivacyProtectTechContact": true
}
```

Ausgabe:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen `get-operation-detail`. Weitere Informationen finden Sie unter [get-operation-detail](#).

Weitere Informationen finden Sie unter [Registrieren einer neuen Domäne](#) im Amazon Route 53-Entwicklerhandbuch.

Informationen darüber, für welche Top-Level-Domains (TLDs) Werte erforderlich sind `ExtraParams` und welche Werte gültig sind, finden Sie [ExtraParam](#) in der Amazon Route 53 API 53-Referenz.

- API-Einheiten finden Sie unter [RegisterDomain AWS CLI](#) Befehlsreferenz.

renew-domain

Das folgende Codebeispiel zeigt die Verwendung `renew-domain`.

AWS CLI

Um eine Domain zu erneuern

Mit dem folgenden `renew-domain` Befehl wird die angegebene Domain um fünf Jahre verlängert. Um den Wert für `abzurufen-current-expiry-year`, verwenden Sie den `get-domain-detail` Befehl und konvertieren Sie den Wert von `ExpirationDate` aus dem Unix-Format.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains renew-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --duration-in-years 5 \  
  --current-expiry-year 2020
```

Ausgabe:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen `get-operation-detail`. Weitere Informationen finden Sie unter [get-operation-detail](#).

Die Registrierung für jede Top-Level-Domain (TLD), z. B. `.com` oder `.org`, legt fest, für wie viele Jahre Sie eine Domain maximal verlängern können. Informationen zum maximalen

Verlängerungszeitraum für Ihre Domain finden Sie TLD im Abschnitt „Registrierungs- und Verlängerungszeitraum“ unter [Domains, die Sie bei Amazon Route 53 registrieren können](#) im Amazon Route 53 53-Entwicklerhandbuch.

Weitere Informationen finden Sie unter [Erneuern der Registrierung für eine Domain](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [RenewDomain AWS CLI](#) Befehlsreferenz.

resend-contact-reachability-email

Das folgende Codebeispiel zeigt die Verwendung `resend-contact-reachability-email`.

AWS CLI

Um die Bestätigungs-E-Mail erneut an die aktuelle E-Mail-Adresse des Registranten zu senden, wenden Sie sich an

Mit dem folgenden `resend-contact-reachability-email` Befehl wird die Bestätigungs-E-Mail erneut an die aktuelle E-Mail-Adresse des Registrantenkontakts für die Domäne `example.com` gesendet.

Dieser Befehl wird nur in der Region ausgeführt. `us-east-1` Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains resend-contact-reachability-email \  
  --region us-east-1 \  
  --domain-name example.com
```

Ausgabe:

```
{  
  "domainName": "example.com",  
  "emailAddress": "moliveira@example.com",  
  "isAlreadyVerified": true  
}
```

Wenn der Wert von `isAlreadyVerified` ist `true`, wie in diesem Beispiel, hat der Kontakt des Registranten bereits bestätigt, dass die angegebene E-Mail-Adresse erreichbar ist.

Weitere Informationen finden Sie unter [Erneutes Senden von Autorisierungs- und Bestätigungs-E-Mails](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [ResendContactReachabilityEmail AWS CLI](#) Befehlsreferenz.

retrieve-domain-auth-code

Das folgende Codebeispiel zeigt die Verwendung `retrieve-domain-auth-code`.

AWS CLI

Um den Autorisierungscode für eine Domain zu erhalten, damit Sie die Domain an einen anderen Registrar übertragen können

Mit dem folgenden `retrieve-domain-auth-code` Befehl wird der aktuelle Autorisierungscode für die Domain `example.com` abgerufen. Sie geben diesen Wert einem anderen Domain-Registrar, wenn Sie die Domain an diesen Registrar übertragen möchten.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains retrieve-domain-auth-code \  
  --region us-east-1 \  
  --domain-name example.com
```

Ausgabe:

```
{  
  "AuthCode": ")o!v3dJeXampLe"  
}
```

Weitere Informationen finden Sie unter [Übertragung einer Domain von Amazon Route 53 zu einem anderen Registrar](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [RetrieveDomainAuthCode](#) in der AWS CLI Befehlsreferenz.

transfer-domain

Das folgende Codebeispiel zeigt die Verwendung `transfer-domain`.

AWS CLI

Um eine Domain zu Amazon Route 53 zu übertragen

Mit dem folgenden `transfer-domain` Befehl wird eine Domäne mit den in der JSON - formatierten Datei bereitgestellten Parametern an Route 53 übertragen. `C:\temp\transfer-domain.json`

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains transfer-domain \  
  --region us-east-1 \  
  --cli-input-json file://C:\temp\transfer-domain.json
```

Inhalt von `transfer-domain.json`:

```
{  
  "DomainName": "example.com",  
  "DurationInYears": 1,  
  "Nameservers": [  
    {  
      "Name": "ns-2048.awsdns-64.com"  
    },  
    {  
      "Name": "ns-2049.awsdns-65.net"  
    },  
    {  
      "Name": "ns-2050.awsdns-66.org"  
    },  
    {  
      "Name": "ns-2051.awsdns-67.co.uk"  
    }  
  ],  
  "AuthCode": ")o!v3dJeXampLe",  
  "AutoRenew": true,  
  "AdminContact": {  
    "FirstName": "Martha",  
    "LastName": "Rivera",  
    "ContactType": "PERSON",  
    "OrganizationName": "Example",  
    "AddressLine1": "1 Main Street",  
    "City": "Anytown",  
    "State": "WA",  
    "CountryCode": "US",  
    "ZipCode": "98101",
```

```
    "PhoneNumber": "+1.8005551212",
    "Email": "mrivera@example.com"
  },
  "RegistrantContact": {
    "FirstName": "Li",
    "LastName": "Juan",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "ljuan@example.com"
  },
  "TechContact": {
    "FirstName": "Mateo",
    "LastName": "Jackson",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "mjackson@example.com"
  },
  "PrivacyProtectAdminContact": true,
  "PrivacyProtectRegistrantContact": true,
  "PrivacyProtectTechContact": true
}
```

Ausgabe:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen `get-operation-detail`. Weitere Informationen finden Sie unter [get-operation-detail](#).

Weitere Informationen finden Sie unter [Übertragung der Registrierung für eine Domain auf Amazon Route 53](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [TransferDomain](#) in der AWS CLI Befehlsreferenz.

update-domain-contact-privacy

Das folgende Codebeispiel zeigt die Verwendung `update-domain-contact-privacy`.

AWS CLI

Um die Datenschutzeinstellungen für die Kontakte einer Domain zu aktualisieren

Mit dem folgenden `update-domain-contact-privacy` Befehl wird der Datenschutz für den Administratorkontakt für die Domäne `example.com` deaktiviert. Dieser Befehl wird nur in der `us-east-1` Region ausgeführt.

Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains update-domain-contact-privacy \
  --region us-east-1 \
  --domain-name example.com \
  --no-admin-privacy
```

Ausgabe:

```
{
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen `get-operation-detail`. Weitere Informationen finden Sie unter [get-operation-detail](#).

Weitere Informationen finden Sie unter [Aktivieren oder Deaktivieren des Datenschutzes für Kontaktinformationen für eine Domain](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateDomainContactPrivacy](#) in der AWS CLI Befehlsreferenz.

update-domain-contact

Das folgende Codebeispiel zeigt die Verwendung `update-domain-contact`.

AWS CLI

Um die Kontaktinformationen für eine Domain zu aktualisieren

Der folgende `update-domain-contact` Befehl aktualisiert die Kontaktinformationen für eine Domain und ruft die Parameter aus der JSON -formatierten Datei ab. `C:\temp\update-domain-contact.json`

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains update-domain-contact \
  --region us-east-1 \
  --cli-input-json file://C:\temp\update-domain-contact.json
```

Inhalt von `update-domain-contact.json`:

```
{
  "AdminContact": {
    "AddressLine1": "101 Main Street",
    "AddressLine2": "Suite 1a",
    "City": "Seattle",
    "ContactType": "COMPANY",
    "CountryCode": "US",
    "Email": "w.xiulan@example.com",
    "FirstName": "Wang",
    "LastName": "Xiulan",
    "OrganizationName": "Example",
    "PhoneNumber": "+1.8005551212",
    "State": "WA",
    "ZipCode": "98101"
  },
  "DomainName": "example.com",
  "RegistrantContact": {
    "AddressLine1": "101 Main Street",
    "AddressLine2": "Suite 1a",
    "City": "Seattle",
    "ContactType": "COMPANY",
    "CountryCode": "US",
    "Email": "w.xiulan@example.com",
    "FirstName": "Wang",
    "LastName": "Xiulan",
    "OrganizationName": "Example",
```

```
    "PhoneNumber": "+1.8005551212",
    "State": "WA",
    "ZipCode": "98101"
  },
  "TechContact": {
    "AddressLine1": "101 Main Street",
    "AddressLine2": "Suite 1a",
    "City": "Seattle",
    "ContactType": "COMPANY",
    "CountryCode": "US",
    "Email": "w.xiulan@example.com",
    "FirstName": "Wang",
    "LastName": "Xiulan",
    "OrganizationName": "Example",
    "PhoneNumber": "+1.8005551212",
    "State": "WA",
    "ZipCode": "98101"
  }
}
```

Ausgabe:

```
{
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen [get-domain-detail](#). Weitere Informationen finden Sie unter [Aktualisieren der Kontaktinformationen für eine Domain](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateDomainContact](#) unter AWS CLI Befehlsreferenz.

update-domain-nameservers

Das folgende Codebeispiel zeigt die Verwendung `update-domain-nameservers`.

AWS CLI

Um die Nameserver für eine Domain zu aktualisieren

Der folgende `update-domain-nameservers` Befehl aktualisiert die Nameserver für eine Domain.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains update-domain-nameservers \
  --region us-east-1 \
  --domain-name example.com \
  --
nameservers Name=ns-1.awsdns-01.org Name=ns-2.awsdns-02.co.uk Name=ns-3.awsdns-03.net Name=
```

Ausgabe:

```
{
  "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

Um zu bestätigen, dass der Vorgang erfolgreich war, können Sie ihn ausführen [get-domain-detail](#).

Weitere Informationen finden Sie unter [Hinzufügen oder Ändern von Nameservern und Glue-Datensätzen für eine Domain](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateDomainNameservers](#) in der AWS CLI Befehlsreferenz.

update-tags-for-domain

Das folgende Codebeispiel zeigt die Verwendung `update-tags-for-domain`.

AWS CLI

Um Tags für eine Domain hinzuzufügen oder zu aktualisieren

Mit dem folgenden `update-tags-for-domain` Befehl werden zwei Schlüssel und die entsprechenden Werte für die Domäne `example.com` hinzugefügt oder aktualisiert. Um den Wert für einen Schlüssel zu aktualisieren, geben Sie einfach den Schlüssel und den neuen Wert an. Sie können jeweils nur in einer Domain Tags hinzufügen oder aktualisieren.

Dieser Befehl wird nur in der `us-east-1` Region ausgeführt. Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains update-tags-for-domain \
  --region us-east-1 \
  --domain-name example.com \
```

```
--tags-to-update "Key=key1,Value=value1" "Key=key2,Value=value2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um zu bestätigen, dass die Tags hinzugefügt oder aktualisiert wurden, können Sie ausführen [list-tags-for-domain](#).

Weitere Informationen finden Sie unter [Tagging Amazon Route 53 53-Ressourcen](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [UpdateTagsForDomain AWS CLI-Befehlsreferenz](#).

view-billing

Das folgende Codebeispiel zeigt die Verwendung `view-billing`.

AWS CLI

Um Rechnungsinformationen für die Gebühren für die Domainregistrierung für das AWS Girokonto abzurufen

Mit dem folgenden `view-billing` Befehl werden alle domänenbezogenen Abrechnungsdatensätze für das Girokonto für den Zeitraum vom 1. Januar 2018 (1514764800 in Unix-Zeit) bis Mitternacht am 31. Dezember 2019 (1577836800 in Unix-Zeit) zurückgegeben.

Dieser Befehl wird nur in der Region ausgeführt. `us-east-1` Wenn Ihre Standardregion auf eingestellt ist `us-east-1`, können Sie den `region` Parameter weglassen.

```
aws route53domains view-billing \  
  --region us-east-1 \  
  --start-time 1514764800 \  
  --end-time 1577836800
```

Ausgabe:

```
{  
  "BillingRecords": [  
    {  
      "DomainName": "example.com",  
      "Operation": "RENEW_DOMAIN",  
      "InvoiceId": "149962827",  
      "BillDate": 1536618063.181,  
      "Price": 12.0
```

```
    },  
    {  
      "DomainName": "example.com",  
      "Operation": "RENEW_DOMAIN",  
      "InvoiceId": "290913289",  
      "BillDate": 1568162630.884,  
      "Price": 12.0  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ViewBilling](#) in der Amazon Route 53 API 53-Referenz.

- API-Einheiten finden Sie [ViewBilling](#) unter AWS CLI Befehlsreferenz.

Beispiele für Route 53 53-Profile mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Route 53 53-Profilen Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-profile

Das folgende Codebeispiel zeigt die Verwendung `associate-profile`.

AWS CLI

Um ein Profil zuzuordnen

Das folgende `associate-profile` Beispiel ordnet ein Profil einem zuVPC.

```
aws route53profiles associate-profile \  
  --name test-association \  
  --profile-id rp-4987774726example \  
  --resource-id vpc-0af3b96b3example
```

Ausgabe:

```
{  
  "ProfileAssociation": {  
    "CreationTime": 1710851336.527,  
    "Id": "rpassoc-489ce212fexample",  
    "ModificationTime": 1710851336.527,  
    "Name": "test-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceId": "vpc-0af3b96b3example",  
    "Status": "CREATING",  
    "StatusMessage": "Creating Profile Association"  
  }  
}
```

Weitere Informationen finden Sie unter [Using Profiles](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [AssociateProfile](#) in der AWS CLI Befehlsreferenz.

associate-resource-to-profile

Das folgende Codebeispiel zeigt die Verwendung `associate-resource-to-profile`.

AWS CLI

Um eine Ressource einem Profil zuzuordnen

Im folgenden `associate-resource-to-profile` Beispiel wird eine DNS Firewall-Regelgruppe mit der Priorität 102 einem Profil zugeordnet.

```
aws route53profiles associate-resource-to-profile \  
  --name test-resource-association \  
  --profile-id rp-4987774726example \  
  --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-  
group/rslvr-frg-cfe7f72example \  
  --resource-properties '{"priority": 102}'
```

Ausgabe:

```
{
  "ProfileResourceAssociation": {
    "CreationTime": 1710851216.613,
    "Id": "rpr-001913120a7example",
    "ModificationTime": 1710851216.613,
    "Name": "test-resource-association",
    "OwnerId": "123456789012",
    "ProfileId": "rp-4987774726example",
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
    "ResourceProperties": "{\"priority\":102}",
    "ResourceType": "FIREWALL_RULE_GROUP",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Profile to DNS Firewall rule group
association"
  }
}
```

- API-Einheiten finden Sie [AssociateResourceToProfile](#) unter AWS CLI Befehlsreferenz.

create-profile

Das folgende Codebeispiel zeigt die Verwendung `create-profile`.

AWS CLI

Um ein Profil zu erstellen

Im folgenden `create-profile` Beispiel wird ein Profil erstellt.

```
aws route53profiles create-profile \
  --name test
```

Ausgabe:

```
{
  "Profile": {
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
    "ClientToken": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
```

```
"CreationTime": 1710850903.578,  
"Id": "rp-6ffe47d5example",  
"ModificationTime": 1710850903.578,  
"Name": "test",  
"OwnerId": "123456789012",  
"ShareStatus": "NOT_SHARED",  
"Status": "COMPLETE",  
"StatusMessage": "Created Profile"  
}  
}
```

- API-Einheiten finden Sie [CreateProfile](#) in der AWS CLI Befehlsreferenz.

delete-profile

Das folgende Codebeispiel zeigt die Verwendung `delete-profile`.

AWS CLI

Um ein Profil zu löschen

Im folgenden `delete-profile` Beispiel wird ein Profil gelöscht.

```
aws route53profiles delete-profile \  
--profile-id rp-6ffe47d5example
```

Ausgabe:

```
{  
  "Profile": {  
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/  
rp-6ffe47d5example",  
    "ClientToken": "0a15fec0-05d9-4f78-bec0-EXAMPLE11111",  
    "CreationTime": 1710850903.578,  
    "Id": "rp-6ffe47d5example",  
    "ModificationTime": 1710850903.578,  
    "Name": "test",  
    "OwnerId": "123456789012",  
    "ShareStatus": "NOT_SHARED",  
    "Status": "DELETED",  
    "StatusMessage": "Deleted Profile"  
  }  
}
```



```
}
```

- API-Einheiten finden Sie [DeleteProfile](#) in der AWS CLI Befehlsreferenz.

disassociate-profile

Das folgende Codebeispiel zeigt die Verwendung `disassociate-profile`.

AWS CLI

Um die Zuordnung eines Profils aufzuheben

Im folgenden `disassociate-profile` Beispiel wird die Zuordnung eines Profils von einem getrennt. VPC

```
aws route53profiles disassociate-profile \  
  --profile-id rp-4987774726example \  
  --resource-id vpc-0af3b96b3example
```

Ausgabe:

```
{  
  "ProfileAssociation": {  
    "CreationTime": 1710851336.527,  
    "Id": "rpassoc-489ce212fexample",  
    "ModificationTime": 1710851401.362,  
    "Name": "test-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceId": "vpc-0af3b96b3example",  
    "Status": "DELETING",  
    "StatusMessage": "Deleting Profile Association"  
  }  
}
```

- API-Einheiten finden Sie [DisassociateProfile](#) in der AWS CLI Befehlsreferenz.

disassociate-resource-from-profile

Das folgende Codebeispiel zeigt die Verwendung `disassociate-resource-from-profile`.

AWS CLI

Um die Zuordnung einer Ressource zu einem Profil aufzuheben

Im folgenden `disassociate-resource-from-profile` Beispiel wird die Zuordnung einer DNS Firewall-Regelgruppe zu einem Profil aufgehoben.

```
aws route53profiles disassociate-resource-from-profile \  
  --profile-id rp-4987774726example \  
  --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-  
group/rslvr-frg-cfe7f72example
```

Ausgabe:

```
{  
  "ProfileResourceAssociation": {  
    "CreationTime": 1710851216.613,  
    "Id": "rpr-001913120a7example",  
    "ModificationTime": 1710852624.36,  
    "Name": "test-resource-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-  
rule-group/rslvr-frg-cfe7f72example",  
    "ResourceProperties": "{\"priority\":105}",  
    "ResourceType": "FIREWALL_RULE_GROUP",  
    "Status": "DELETING",  
    "StatusMessage": "Deleting the Profile to DNS Firewall rule group  
association"  
  }  
}
```

- API-Einheiten finden Sie [DisassociateResourceFromProfile](#) in der AWS CLI Befehlsreferenz.

get-profile-association

Das folgende Codebeispiel zeigt die Verwendung `get-profile-association`.

AWS CLI

Um Informationen über eine Profilverknüpfung abzurufen

Im Folgenden werden Informationen zur angegebenen Profilverknüpfung `get-profile-association` zurückgegeben.

```
aws route53profiles get-profile-association \  
  --profile-association-id rpassoc-489ce212fexample
```

Ausgabe:

```
{  
  "ProfileAssociation": {  
    "CreationTime": 1709338817.148,  
    "Id": "rrpassoc-489ce212fexample",  
    "ModificationTime": 1709338974.772,  
    "Name": "test-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceId": "vpc-0af3b96b3example",  
    "Status": "COMPLETE",  
    "StatusMessage": "Created Profile Association"  
  }  
}
```

- API-Einheiten finden Sie [GetProfileAssociation](#) unter AWS CLI Befehlsreferenz.

get-profile-resource-association

Das folgende Codebeispiel zeigt die Verwendung `get-profile-resource-association`.

AWS CLI

Um Informationen über eine Ressource abzurufen, die einem Profil zugeordnet ist

Im Folgenden werden Informationen über die angegebene Ressourcenzuweisung zu einem Profil `get-profile-resource-association` zurückgegeben.

```
aws route53profiles get-profile-resource-association \  
  --profile-resource-association-id rpr-001913120a7example
```

Ausgabe:

```
{
```

```

    "ProfileResourceAssociation": {
      "CreationTime": 1710851216.613,
      "Id": "rpr-001913120a7example",
      "ModificationTime": 1710852303.798,
      "Name": "test-resource-association",
      "OwnerId": "123456789012",
      "ProfileId": "rp-4987774726example",
      "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
      "ResourceProperties": "{\"priority\":105}",
      "ResourceType": "FIREWALL_RULE_GROUP",
      "Status": "COMPLETE",
      "StatusMessage": "Completed creation of Profile to DNS Firewall rule group
association"
    }
  }
}

```

- API-Einheiten finden Sie [GetProfileResourceAssociation](#) unter AWS CLI Befehlsreferenz.

get-profile

Das folgende Codebeispiel zeigt die Verwendung `get-profile`.

AWS CLI

Um Informationen über ein Profil zu erhalten

Im Folgenden werden Informationen über das angegebene Profil `get-profile` zurückgegeben.

```

aws route53profiles get-profile \
  --profile-id rp-4987774726example

```

Ausgabe:

```

{
  "Profile": {
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
    "ClientToken": "0cbc5ae7-4921-4204-bea9-EXAMPLE11111",
    "CreationTime": 1710851044.288,
    "Id": "rp-4987774726example",
    "ModificationTime": 1710851044.288,

```

```
    "Name": "test",
    "OwnerId": "123456789012",
    "ShareStatus": "NOT_SHARED",
    "Status": "COMPLETE",
    "StatusMessage": "Created Profile"
  }
}
```

- API-Einheiten finden Sie [GetProfile](#) unter AWS CLI Befehlsreferenz.

list-profile-associations

Das folgende Codebeispiel zeigt die Verwendung `list-profile-associations`.

AWS CLI

Um Profilzuordnungen aufzulisten

Im Folgenden `list-profile-associations` werden die Profilzuordnungen in Ihrem AWS Konto aufgeführt.

```
aws route53profiles list-profile-associations
```

Ausgabe:

```
{
  "ProfileAssociations": [
    {
      "CreationTime": 1709338817.148,
      "Id": "rpassoc-489ce212fexample",
      "ModificationTime": 1709338974.772,
      "Name": "test-association",
      "OwnerId": "123456789012",
      "ProfileId": "rp-4987774726example",
      "ResourceId": "vpc-0af3b96b3example",
      "Status": "COMPLETE",
      "StatusMessage": "Created Profile Association"
    }
  ]
}
```

- API-Einheiten finden Sie [ListProfileAssociations](#) in der AWS CLI Befehlsreferenz.

list-profile-resource-associations

Das folgende Codebeispiel zeigt die Verwendung `list-profile-resource-associations`.

AWS CLI

Um Profilressourcenzuordnungen aufzulisten

Im `list-profile-resource-associations` Folgenden werden die Profilressourcenzuordnungen für das angegebene Profil aufgeführt.

```
aws route53profiles list-profile-resource-associations \
  --profile-id rp-4987774726example
```

Ausgabe:

```
{
  "ProfileResourceAssociations": [
    {
      "CreationTime": 1710851216.613,
      "Id": "rpr-001913120a7example",
      "ModificationTime": 1710851216.613,
      "Name": "test-resource-association",
      "OwnerId": "123456789012",
      "ProfileId": "rp-4987774726example",
      "ResourceArn": "arn:aws:route53resolver:us-
east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example",
      "ResourceProperties": "{\"priority\":102}",
      "ResourceType": "FIREWALL_RULE_GROUP",
      "Status": "COMPLETE",
      "StatusMessage": "Completed creation of Profile to DNS Firewall rule
group association"
    }
  ]
}
```

- API-Einheiten finden Sie [ListProfileResourceAssociations](#) in der AWS CLI Befehlsreferenz.

list-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-profiles`.

AWS CLI

Um Profile aufzulisten

Im Folgenden `list-profiles` werden die Profile in Ihrem AWS Konto aufgeführt und zusätzliche Informationen zu ihnen angezeigt.

```
aws route53profiles list-profiles
```

Ausgabe:

```
{
  "ProfileSummaries": [
    {
      "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
      "Id": "rp-4987774726example",
      "Name": "test",
      "ShareStatus": "NOT_SHARED"
    }
  ]
}
```

- API-Einheiten finden Sie [ListProfiles](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine Ressource aufzulisten

Im Folgenden `list-tags-for-resource` werden die Tags für die angegebene Ressource aufgeführt.

```
aws route53profiles list-tags-for-resource \
  --resource-arn arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example
```

Ausgabe:

```
{
  "Tags": {
    "my-key-2": "my-value-2",
    "my-key-1": "my-value-1"
  }
}
```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

update-profile-resource-association

Das folgende Codebeispiel zeigt die Verwendung `update-profile-resource-association`.

AWS CLI

Um eine Ressource zu aktualisieren, die einem Profil zugeordnet ist

Im Folgenden wird die Priorität einer DNS Firewall-Regelgruppe `update-profile-resource-association` aktualisiert, die dem Profil zugeordnet ist.

```
aws route53profiles update-profile-resource-association \
  --profile-resource-association-id rpr-001913120a7example \
  --resource-properties "{\"priority\": 105}"
```

Ausgabe:

```
{
  "ProfileResourceAssociation": {
    "CreationTime": 1710851216.613,
    "Id": "rpr-001913120a7example",
    "ModificationTime": 1710852303.798,
    "Name": "test-resource-association",
    "OwnerId": "123456789012",
    "ProfileId": "rp-4987774726example",
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
    "ResourceProperties": "{\"priority\":105}",
    "ResourceType": "FIREWALL_RULE_GROUP",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Profile to DNS Firewall rule group
association"
```



```
}  
}
```

- API-Einheiten finden Sie [UpdateProfileResourceAssociation](#) in der AWS CLI Befehlsreferenz.

Beispiele für Route 53 Resolver mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Route 53 Resolver Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-firewall-rule-group

Das folgende Codebeispiel zeigt die Verwendung `associate-firewall-rule-group`.

AWS CLI

Um eine Firewall-Regelgruppe einem zuzuordnen VPC

Im folgenden `associate-firewall-rule-group` Beispiel wird eine DNS Firewall-Regelgruppe einem Amazon zugeordnet VPC.

```
aws route53resolver associate-firewall-rule-group \  
  --name test-association \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --vpc-id vpc-31e92222 \  
  --priority 101
```

Ausgabe:

```
{
  "FirewallRuleGroupAssociation": {
    "Id": "rslvr-frgassoc-57e8873d7example",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "VpcId": "vpc-31e92222",
    "Name": "test-association",
    "Priority": 101,
    "MutationProtection": "DISABLED",
    "Status": "UPDATING",
    "StatusMessage": "Creating Firewall Rule Group Association",
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:47:48.755768Z",
    "ModificationTime": "2021-05-25T21:47:48.755768Z"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung von Verknüpfungen zwischen Ihren VPC und Route 53 Resolver DNS Firewall-Regelgruppen](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [AssociateFirewallRuleGroup](#) in der AWS CLI Befehlsreferenz.

associate-resolver-endpoint-ip-address

Das folgende Codebeispiel zeigt die Verwendung `associate-resolver-endpoint-ip-address`.

AWS CLI

Um einem Resolver-Endpoint eine andere IP-Adresse zuzuordnen

Im folgenden `associate-resolver-endpoint-ip-address` Beispiel wird einem eingehenden Resolver-Endpoint eine weitere IP-Adresse zugeordnet. Wenn Sie nur eine Subnetz-ID angeben und die IP-Adresse nicht im `--ip-address` Parameter angeben, wählt Resolver aus den verfügbaren IP-Adressen im angegebenen Subnetz eine IP-Adresse für Sie aus.

```
aws route53resolver associate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-497098ad5example \
```

```
--ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"
```

Ausgabe:

```
{
  "ResolverEndpoint": {
    "Id": "rslvr-in-497098ad5example",
    "CreatorRequestId": "AWSConsole.25.0123456789",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad5example",
    "Name": "my-inbound-endpoint",
    "SecurityGroupIds": [
      "sg-05cd7b25d6example"
    ],
    "Direction": "INBOUND",
    "IpAddressCount": 3,
    "HostVPCId": "vpc-304bexam",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Resolver Endpoint",
    "CreationTime": "2020-01-02T23:25:45.538Z",
    "ModificationTime": "2020-01-02T23:25:45.538Z"
  }
}
```

Weitere Informationen finden Sie unter [Werte, die Sie angeben, wenn Sie eingehende Endpunkte erstellen oder bearbeiten](#), im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie unter [AssociateResolverEndpointIpAddress AWS CLI-Befehlsreferenz](#).

associate-resolver-rule

Das folgende Codebeispiel zeigt die Verwendung `associate-resolver-rule`.

AWS CLI

Um eine Resolver-Regel mit einem zu verknüpfen VPC

Das folgende `associate-resolver-rule` Beispiel verknüpft eine Resolver-Regel mit einem Amazon VPC. Nachdem Sie den Befehl ausgeführt haben, leitet DNS Resolver basierend auf den Einstellungen in der Regel, z. B. dem Domainnamen der weitergeleiteten Anfragen, Anfragen an Ihr Netzwerk weiter.

```
aws route53resolver associate-resolver-rule \  
  --name my-resolver-rule-association \  
  --resolver-rule-id rslvr-rr-42b60677c0example \  
  --vpc-id vpc-304bexam
```

Ausgabe:

```
{  
  "ResolverRuleAssociation": {  
    "Id": "rslvr-rrassoc-d61cbb2c8bexample",  
    "ResolverRuleId": "rslvr-rr-42b60677c0example",  
    "Name": "my-resolver-rule-association",  
    "VPCId": "vpc-304bexam",  
    "Status": "CREATING",  
    "StatusMessage": "[Trace id: 1-5dc5a8fa-ec2cc480d2ef07617example] Creating  
the association."  
  }  
}
```

Weitere Informationen finden Sie unter [Weiterleiten ausgehender DNS Anfragen an Ihr Netzwerk](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [AssociateResolverRule AWS CLI](#) Befehlsreferenz.

create-firewall-domain-list

Das folgende Codebeispiel zeigt die Verwendung `create-firewall-domain-list`.

AWS CLI

So erstellen Sie eine Route 53 Resolver DNS Firewall-Domänenliste

Im folgenden `create-firewall-domain-list` Beispiel wird in Ihrem AWS Konto eine Route 53 Resolver DNS Firewall-Domänenliste mit dem Namen `test` erstellt.

```
aws route53resolver create-firewall-domain-list \  
  --creator-request-id my-request-id \  
  --name test
```

Ausgabe:

```
{
  "FirewallDomainList": {
    "Id": "rslvr-fdl-d61cbb2cbexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/rslvr-fdl-d61cbb2cbexample",
    "Name": "test",
    "DomainCount": 0,
    "Status": "COMPLETE",
    "StatusMessage": "Created Firewall Domain List",
    "CreatorRequestId": "my-request-id",
    "CreationTime": "2021-05-25T15:55:51.115365Z",
    "ModificationTime": "2021-05-25T15:55:51.115365Z"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer eigenen Domainlisten](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateFirewallDomainList](#) in der AWS CLI Befehlsreferenz.

create-firewall-rule-group

Das folgende Codebeispiel zeigt die Verwendung `create-firewall-rule-group`.

AWS CLI

Um eine Firewall-Regelgruppe zu erstellen

Im folgenden `create-firewall-rule-group` Beispiel wird eine DNS Firewall-Regelgruppe erstellt.

```
aws route53resolver create-firewall-rule-group \
  --creator-request-id my-request-id \
  --name test
```

Ausgabe:

```
{
  "FirewallRuleGroup": {
    "Id": "rslvr-frg-47f93271fexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/rslvr-frg-47f93271fexample",
```

```

    "Name": "test",
    "RuleCount": 0,
    "Status": "COMPLETE",
    "StatusMessage": "Created Firewall Rule Group",
    "OwnerId": "123456789012",
    "CreatorRequestId": "my-request-id",
    "ShareStatus": "NOT_SHARED",
    "CreationTime": "2021-05-25T18:59:26.490017Z",
    "ModificationTime": "2021-05-25T18:59:26.490017Z"
  }
}

```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [CreateFirewallRuleGroup](#) in der AWS CLI Befehlsreferenz.

create-firewall-rule

Das folgende Codebeispiel zeigt die Verwendung `create-firewall-rule`.

AWS CLI

Um eine Firewallregel zu erstellen

Im folgenden `create-firewall-rule` Beispiel wird eine Firewallregel in einer DNS Firewallregel für Domänen erstellt, die in einer DNS Firewall-Domänenliste aufgeführt sind.

```

aws route53resolver create-firewall-rule \
  --name allow-rule \
  --firewall-rule-group-id rslvr-frg-47f93271fexample \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
  --priority 101 \
  --action ALLOW

```

Ausgabe:

```

{
  "FirewallRule": {
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
    "Name": "allow-rule",

```

```

    "Priority": 101,
    "Action": "ALLOW",
    "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:44:00.346093Z",
    "ModificationTime": "2021-05-25T21:44:00.346093Z"
  }
}

```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [CreateFirewallRule](#) in der AWS CLI Befehlsreferenz.

create-resolver-endpoint

Das folgende Codebeispiel zeigt die Verwendung `create-resolver-endpoint`.

AWS CLI

Um einen Resolver-Endpunkt für eingehenden Datenverkehr zu erstellen

Im folgenden `create-resolver-endpoint` Beispiel wird ein Resolver-Endpunkt für eingehende Anrufe erstellt. Sie können denselben Befehl verwenden, um sowohl eingehende als auch ausgehende Endpunkte zu erstellen.

```
aws route53resolver create-resolver-endpoint --name my-inbound-endpoint -- creator-request-id 2020-01-01-18:47 -- security-group-ids „sg-f62bexam“ --direction --ip-addresses =subnet-ba47exam, Ip=192.0.2.255 =subnet-12d8exam, Ip=192.0.2.254 INBOUND SubnetId SubnetId
```

Ausgabe:

```

{
  "ResolverEndpoint": {
    "Id": "rslvr-in-f9ab8a03f1example",
    "CreatorRequestId": "2020-01-01-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/rslvr-in-f9ab8a03f1example",
    "Name": "my-inbound-endpoint",
    "SecurityGroupIds": [
      "sg-f62bexam"
    ],
    "Direction": "INBOUND",
  }
}

```

```

    "IpAddressCount": 2,
    "HostVPCId": "vpc-304examp",
    "Status": "CREATING",
    "StatusMessage": "[Trace id: 1-5dc1ff84-f3477826e4a190025example] Creating
the Resolver Endpoint",
    "CreationTime": "2020-01-01T23:02:29.583Z",
    "ModificationTime": "2020-01-01T23:02:29.583Z"
  }
}

```

Um einen Outbound-Resolver-Endpoint zu erstellen

Im folgenden `create-resolver-endpoint` Beispiel wird mithilfe der Werte im -formatierten Dokument ein Outbound-Resolver-Endpoint erstellt. JSON `create-outbound-resolver-endpoint.json`

```

aws route53resolver create-resolver-endpoint \
  --cli-input-json file://c:\temp\create-outbound-resolver-endpoint.json

```

Inhalt von `create-outbound-resolver-endpoint.json`:

```

{
  "CreatorRequestId": "2020-01-01-18:47",
  "Direction": "OUTBOUND",
  "IpAddresses": [
    {
      "Ip": "192.0.2.255",
      "SubnetId": "subnet-ba47exam"
    },
    {
      "Ip": "192.0.2.254",
      "SubnetId": "subnet-12d8exam"
    }
  ],
  "Name": "my-outbound-endpoint",
  "SecurityGroupIds": [ "sg-05cd7b25d6example" ],
  "Tags": [
    {
      "Key": "my-key-name",
      "Value": "my-key-value"
    }
  ]
}

```



```
}
```

Weitere Informationen finden Sie unter [Auflösen von DNS Abfragen zwischen VPCs und Ihrem Netzwerk](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [CreateResolverEndpoint AWS CLI Befehlsreferenz](#).

create-resolver-rule

Das folgende Codebeispiel zeigt die Verwendung `create-resolver-rule`.

AWS CLI

Um eine Resolver-Regel zu erstellen

Im folgenden `create-resolver-rule` Beispiel wird eine Resolver-Weiterleitungsregel erstellt. Die Regel verwendet den ausgehenden Endpunkt `rslvr-out-d5e5920e37example`, um DNS-Anfragen an die IP-Adressen `10.24.8.75` und `10.24.8.156` weiterzuleiten. `example.com`

```
aws route53resolver create-resolver-rule \
  --creator-request-id 2020-01-02-18:47 \
  --domain-name example.com \
  --name my-rule \
  --resolver-endpoint-id rslvr-out-d5e5920e37example \
  --rule-type FORWARD \
  --target-ips "Ip=10.24.8.75" "Ip=10.24.8.156"
```

Ausgabe:

```
{
  "ResolverRule": {
    "Status": "COMPLETE",
    "RuleType": "FORWARD",
    "ResolverEndpointId": "rslvr-out-d5e5920e37example",
    "Name": "my-rule",
    "DomainName": "example.com.",
    "CreationTime": "2022-05-10T21:35:30.923187Z",
    "TargetIps": [
      {
        "Ip": "10.24.8.75",
        "Port": 53
      }
    ]
  }
}
```

```

    },
    {
      "Ip": "10.24.8.156",
      "Port": 53
    }
  ],
  "CreatorRequestId": "2022-05-10-16:33",
  "ModificationTime": "2022-05-10T21:35:30.923187Z",
  "ShareStatus": "NOT_SHARED",
  "Arn": "arn:aws:route53resolver:us-east-1:111117012054:resolver-rule/rslvr-rr-b1e0b905e93611111",
  "OwnerId": "111111111111",
  "Id": "rslvr-rr-rslvr-rr-b1e0b905e93611111",
  "StatusMessage": "[Trace id: 1-22222222-3e56afcc71a3724664f22e24]
  Successfully created Resolver Rule."
}
}

```

- API-Einheiten [CreateResolverRule AWS CLI](#) finden Sie in der Befehlsreferenz.

delete-firewall-domain-list

Das folgende Codebeispiel zeigt die Verwendung `delete-firewall-domain-list`.

AWS CLI

So löschen Sie eine Route 53 Resolver DNS Firewall-Domänenliste

Im folgenden `delete-firewall-domain-list` Beispiel wird eine Route 53 Resolver DNS Firewall-Domänenliste mit dem Namen `test` in Ihrem AWS Konto gelöscht.

```
aws route53resolver delete-firewall-domain-list \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Ausgabe:

```

{
  "FirewallDomainList": {
    "Id": "rslvr-fdl-9e956e9ffexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/rslvr-fdl-9e956e9ffexample",
    "Name": "test",
  }
}

```

```

    "DomainCount": 6,
    "Status": "DELETING",
    "StatusMessage": "Deleting the Firewall Domain List",
    "CreatorRequestId": "my-request-id",
    "CreationTime": "2021-05-25T15:55:51.115365Z",
    "ModificationTime": "2021-05-25T18:58:05.588024Z"
  }
}

```

Weitere Informationen finden Sie unter [Verwaltung Ihrer eigenen Domainlisten](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteFirewallDomainList](#) in der AWS CLI Befehlsreferenz.

delete-firewall-rule-group

Das folgende Codebeispiel zeigt die Verwendung `delete-firewall-rule-group`.

AWS CLI

Um eine Firewall-Regelgruppe zu löschen

Im folgenden `delete-firewall-rule-group` Beispiel wird eine Firewall-Regelgruppe gelöscht.

```

aws route53resolver delete-firewall-rule-group \
  --firewall-rule-group-id rslvr-frg-47f93271fexample

```

Ausgabe:

```

{
  "FirewallRuleGroup": {
    "Id": "rslvr-frg-47f93271fexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/rslvr-frg-47f93271fexample",
    "Name": "test",
    "RuleCount": 0,
    "Status": "UPDATING",
    "StatusMessage": "Updating Firewall Rule Group",
    "OwnerId": "123456789012",
    "CreatorRequestId": "my-request-id",
    "ShareStatus": "NOT_SHARED",
  }
}

```

```
    "CreationTime": "2021-05-25T18:59:26.490017Z",  
    "ModificationTime": "2021-05-25T21:51:53.028688Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [DeleteFirewallRuleGroup](#) in der AWS CLI Befehlsreferenz.

delete-firewall-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-firewall-rule`.

AWS CLI

Um eine Firewallregel zu löschen

Im folgenden `delete-firewall-rule` Beispiel wird eine angegebene Firewallregel gelöscht.

```
aws route53resolver delete-firewall-rule \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Ausgabe:

```
{  
  "FirewallRule": {  
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",  
    "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",  
    "Name": "allow-rule",  
    "Priority": 102,  
    "Action": "ALLOW",  
    "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",  
    "CreationTime": "2021-05-25T21:44:00.346093Z",  
    "ModificationTime": "2021-05-25T21:45:59.611600Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [DeleteFirewallRule](#) in der AWS CLI Befehlsreferenz.

delete-resolver-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-resolver-endpoint`.

AWS CLI

Um einen Resolver-Endpunkt zu löschen

Im folgenden `delete-resolver-endpoint` Beispiel wird der angegebene Endpunkt gelöscht.

Wichtig: Wenn Sie einen eingehenden Endpunkt löschen, werden DNS-Anfragen aus Ihrem Netzwerk nicht mehr an den Resolver weitergeleitet VPC, den Sie im Endpunkt angegeben haben. Wenn Sie einen ausgehenden Endpunkt löschen, beendet der Resolver die Weiterleitung von DNS-Anfragen von Ihrem an Ihr Netzwerk VPC nach Regeln, die den gelöschten ausgehenden Endpunkt spezifizieren.

```
aws route53resolver delete-resolver-endpoint \
  --resolver-endpoint-id rslvr-in-497098ad59example
```

Ausgabe:

```
{
  "ResolverEndpoint": {
    "Id": "rslvr-in-497098ad59example",
    "CreatorRequestId": "AWSConsole.25.157290example",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad59example",
    "Name": "my-inbound-endpoint",
    "SecurityGroupIds": [
      "sg-05cd7b25d6example"
    ],
    "Direction": "INBOUND",
    "IpAddressCount": 5,
    "HostVPCId": "vpc-304bexam",
    "Status": "DELETING",
    "StatusMessage": "[Trace id: 1-5dc5b658-811b5be0922bbc382example] Deleting
ResolverEndpoint.",
    "CreationTime": "2020-01-01T23:25:45.538Z",
    "ModificationTime": "2020-01-02T23:25:45.538Z"
  }
}
```

```
}
```

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteResolverEndpoint](#). AWS CLI

delete-resolver-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-resolver-rule`.

AWS CLI

Um eine Resolver-Regel zu löschen

Im folgenden `delete-resolver-rule` Beispiel wird die angegebene Regel gelöscht.

Hinweis Wenn eine Regel mit einer Regel verknüpft ist VPCs, müssen Sie zuerst die Zuordnung der Regel von der Regel trennen, VPCs bevor Sie sie löschen können.

```
aws route53resolver delete-resolver-rule \  
  --resolver-rule-id rslvr-rr-5b3809426bexample
```

Ausgabe:

```
{  
  "ResolverRule": {  
    "Id": "rslvr-rr-5b3809426bexample",  
    "CreatorRequestId": "2020-01-03-18:47",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-  
rr-5b3809426bexample",  
    "DomainName": "zenith.example.com.",  
    "Status": "DELETING",  
    "StatusMessage": "[Trace id: 1-5dc5e05b-602e67b052cb74f05example] Deleting  
Resolver Rule.",  
    "RuleType": "FORWARD",  
    "Name": "my-resolver-rule",  
    "TargetIps": [  
      {  
        "Ip": "192.0.2.50",  
        "Port": 53  
      }  
    ],  
    "ResolverEndpointId": "rslvr-out-d5e5920e3example",  
    "OwnerId": "111122223333",
```

```

    "ShareStatus": "NOT_SHARED"
  }
}

```

- API-Einheiten finden Sie [DeleteResolverRule](#) in der AWS CLI Befehlsreferenz.

disassociate-firewall-rule-group

Das folgende Codebeispiel zeigt die Verwendung `disassociate-firewall-rule-group`.

AWS CLI

Um die Zuordnung einer Firewall-Regelgruppe von einer zu trennen VPC

Im folgenden `disassociate-firewall-rule-group` Beispiel wird die Zuordnung einer DNS Firewall-Regelgruppe zu einer VPC Amazon-Gruppe getrennt.

```

aws route53resolver disassociate-firewall-rule-group \
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example

```

Ausgabe:

```

{
  "FirewallRuleGroupAssociation": {
    "Id": "rslvr-frgassoc-57e8873d7example",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "VpcId": "vpc-31e92222",
    "Name": "test-association",
    "Priority": 103,
    "MutationProtection": "DISABLED",
    "Status": "DELETING",
    "StatusMessage": "Deleting the Firewall Rule Group Association",
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:47:48.755768Z",
    "ModificationTime": "2021-05-25T21:51:02.377887Z"
  }
}

```

Weitere Informationen finden Sie unter [Verwaltung von Verknüpfungen zwischen Ihren VPC und Route 53 Resolver DNS Firewall-Regelgruppen](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [DisassociateFirewallRuleGroup](#) in der AWS CLI Befehlsreferenz.

disassociate-resolver-endpoint-ip-address

Das folgende Codebeispiel zeigt die Verwendung `disassociate-resolver-endpoint-ip-address`.

AWS CLI

Um die Zuordnung einer IP-Adresse zu einem Resolver-Endpoint zu trennen

Im folgenden `disassociate-resolver-endpoint-ip-address` Beispiel wird eine IP-Adresse von einem angegebenen eingehenden oder ausgehenden Resolver-Endpoint entfernt.

Hinweis: Ein Endpoint muss mindestens zwei IP-Adressen haben. Wenn ein Endpoint derzeit nur zwei IP-Adressen hat und Sie eine Adresse durch eine andere Adresse ersetzen möchten, müssen Sie zuerst [associate-resolver-endpoint-ip-address](#) verwenden, um die neue IP-Adresse zuzuordnen. Anschließend können Sie die Zuordnung einer der ursprünglichen IP-Adressen zum Endpoint aufheben.

```
aws route53resolver disassociate-resolver-endpoint-ip-address \  
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example \  
  --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"
```

Ausgabe:

```
{  
  "ResolverEndpoint": {  
    "Id": "rslvr-in-f9ab8a03f1example",  
    "CreatorRequestId": "2020-01-01-18:47",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/  
rslvr-in-f9ab8a03f1example",  
    "Name": "my-inbound-endpoint",  
    "SecurityGroupIds": [  
      "sg-f62bexam"  
    ],  
    "Direction": "INBOUND",  
    "IpAddressCount": 3,  
    "HostVPCId": "vpc-304bexam",  
    "Status": "UPDATING",  
    "StatusMessage": "Updating the Resolver Endpoint",
```



```

    "CreationTime": "2020-01-01T23:02:29.583Z",
    "ModificationTime": "2020-01-05T23:02:29.583Z"
  }
}

```

- API-Einheiten finden Sie [DisassociateResolverEndpointIpAddress](#) in der AWS CLI Befehlsreferenz.

disassociate-resolver-rule

Das folgende Codebeispiel zeigt die Verwendung `disassociate-resolver-rule`.

AWS CLI

So trennen Sie die Zuordnung einer Resolver-Regel zu einer Amazon-Regel VPC

Im folgenden `disassociate-resolver-rule` Beispiel wird die Zuordnung zwischen der angegebenen Resolver-Regel und der angegebenen entfernt. VPC Unter den folgenden Umständen können Sie die Zuordnung einer Regel zu VPC einer aufheben:

Bei DNS-Anfragen, die ihren Ursprung in dieser Regel haben VPC, soll Resolver die Weiterleitung von Anfragen an Ihr Netzwerk für den in der Regel angegebenen Domainnamen beenden. Sie möchten die Weiterleitungsregel löschen. Wenn eine Regel derzeit einer oder mehreren Regeln zugeordnet ist VPCs, müssen Sie die Zuordnung der Regel zu allen Regeln aufheben, VPCs bevor Sie sie löschen können.

```

aws route53resolver disassociate-resolver-rule \
  --resolver-rule-id rslvr-rr-4955cb98ceexample \
  --vpc-id vpc-304bexam

```

Ausgabe:

```

{
  "ResolverRuleAssociation": {
    "Id": "rslvr-rrassoc-322f4e8b9cexample",
    "ResolverRuleId": "rslvr-rr-4955cb98ceexample",
    "Name": "my-resolver-rule-association",
    "VPCId": "vpc-304bexam",
    "Status": "DELETING",
    "StatusMessage": "[Trace id: 1-5dc5ffa2-a26c38004c1f94006example] Deleting Association"
  }
}

```

```
}  
}
```

- API-Einheiten finden Sie [DisassociateResolverRule](#) in der AWS CLI Befehlsreferenz.

get-firewall-config

Das folgende Codebeispiel zeigt die Verwendung `get-firewall-config`.

AWS CLI

Um eine Firewall-Konfiguration für ein zu erhalten VPC

Im folgenden `get-firewall-config` Beispiel wird das DNS Firewall-Verhalten für die angegebene VPC Datei abgerufen.

```
aws route53resolver get-firewall-config \  
  --resource-id vpc-31e92222
```

Ausgabe:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e92222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

Weitere Informationen finden Sie unter [DNSVPCFirewall-Konfiguration](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [GetFirewallConfig](#) in der AWS CLI Befehlsreferenz.

get-firewall-domain-list

Das folgende Codebeispiel zeigt die Verwendung `get-firewall-domain-list`.

AWS CLI

So rufen Sie eine Route 53 Resolver DNS Firewall-Domänenliste ab

Im folgenden `get-firewall-domain-list` Beispiel wird die Domänenliste mit der von Ihnen angegebenen ID abgerufen.

```
aws route53resolver get-firewall-domain-list \  
  --firewall-domain-list-id rslvr-fdl-42b60677cexample
```

Ausgabe:

```
{  
  "FirewallDomainList": {  
    "Id": "rslvr-fdl-9e956e9ffexample",  
    "Arn": "arn:aws:route53resolver:us-west-2:123457689012:firewall-domain-list/  
rslvr-fdl-42b60677cexample",  
    "Name": "test",  
    "DomainCount": 0,  
    "Status": "COMPLETE",  
    "StatusMessage": "Created Firewall Domain List",  
    "CreatorRequestId": "my-request-id",  
    "CreationTime": "2021-05-25T15:55:51.115365Z",  
    "ModificationTime": "2021-05-25T15:55:51.115365Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer eigenen Domainlisten](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetFirewallDomainList](#) in der AWS CLI Befehlsreferenz.

get-firewall-rule-group-association

Das folgende Codebeispiel zeigt die Verwendung `get-firewall-rule-group-association`.

AWS CLI

Um eine Firewall-Regelgruppenzuordnung zu erhalten

Im folgenden `get-firewall-rule-group-association` Beispiel wird eine Firewall-Regelgruppenzuordnung abgerufen.

```
aws route53resolver get-firewall-rule-group-association \  
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

Ausgabe:

```
{
  "FirewallRuleGroupAssociation": {
    "Id": "rslvr-frgassoc-57e8873d7example",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "VpcId": "vpc-31e92222",
    "Name": "test-association",
    "Priority": 101,
    "MutationProtection": "DISABLED",
    "Status": "COMPLETE",
    "StatusMessage": "Finished rule group association update",
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:47:48.755768Z",
    "ModificationTime": "2021-05-25T21:47:48.755768Z"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung von Verknüpfungen zwischen Ihren VPC und Route 53 Resolver DNS Firewall-Regelgruppen](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetFirewallRuleGroupAssociation](#) in der AWS CLI Befehlsreferenz.

get-firewall-rule-group-policy

Das folgende Codebeispiel zeigt die Verwendung `get-firewall-rule-group-policy`.

AWS CLI

Um eine AWS IAM Richtlinie zu erhalten

Im folgenden `get-firewall-rule-group-policy` Beispiel wird die AWS Identity and Access Management (AWS IAM) -Richtlinie für die gemeinsame Nutzung der angegebenen Regelgruppe abgerufen.

```
aws route53resolver get-firewall-rule-group-policy \
  --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/
rslvr-frg-47f93271fexample
```

Ausgabe:

```
{
  "FirewallRuleGroupPolicy": "{ \"Version\": \"2012-10-17\",
  \"Statement\": [{ \"Sid\": \"test\", \"Effect\": \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::AWS_ACCOUNT_ID:root\" }, \"Action\": [ \"route53resolver:GetFirewallRuleGroup\", \"route53resolver:ListFirewallRuleGroups\" ], \"Resource\": \"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-group/rslvr-frg-47f93271fexample\" } ] }"
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [GetFirewallRuleGroupPolicy](#) in der AWS CLI Befehlsreferenz.

get-firewall-rule-group

Das folgende Codebeispiel zeigt die Verwendung `get-firewall-rule-group`.

AWS CLI

Um eine Firewall-Regelgruppe abzurufen

Im folgenden `get-firewall-rule-group` Beispiel werden Informationen über eine DNS Firewall-Regelgruppe mit der von Ihnen angegebenen ID abgerufen.

```
aws route53resolver get-firewall-rule-group \
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Ausgabe:

```
{
  "FirewallRuleGroup": {
    "Id": "rslvr-frg-47f93271fexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/rslvr-frg-47f93271fexample",
    "Name": "test",
    "RuleCount": 0,
    "Status": "COMPLETE",
    "StatusMessage": "Created Firewall Rule Group",
    "OwnerId": "123456789012",
    "CreatorRequestId": "my-request-id",
  }
}
```

```
    "ShareStatus": "NOT_SHARED",
    "CreationTime": "2021-05-25T18:59:26.490017Z",
    "ModificationTime": "2021-05-25T18:59:26.490017Z"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [GetFirewallRuleGroup](#) in der AWS CLI Befehlsreferenz.

get-resolver-endpoint

Das folgende Codebeispiel zeigt die Verwendung `get-resolver-endpoint`.

AWS CLI

Um Informationen über einen Resolver-Endpunkt abzurufen

Im folgenden `get-resolver-endpoint` Beispiel werden Details für den angegebenen ausgehenden Endpunkt angezeigt. Sie können es sowohl `get-resolver-endpoint` für eingehende als auch für ausgehende Endpunkte verwenden, indem Sie die entsprechende Endpunkt-ID angeben.

```
aws route53resolver get-resolver-endpoint \
  --resolver-endpoint-id rslvr-out-d5e5920e37example
```

Ausgabe:

```
{
  "ResolverEndpoint": {
    "Id": "rslvr-out-d5e5920e37example",
    "CreatorRequestId": "2020-01-01-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-out-d5e5920e37example",
    "Name": "my-outbound-endpoint",
    "SecurityGroupIds": [
      "sg-05cd7b25d6example"
    ],
    "Direction": "OUTBOUND",
    "IpAddressCount": 2,
```

```

    "HostVPCId": "vpc-304bexam",
    "Status": "OPERATIONAL",
    "StatusMessage": "This Resolver Endpoint is operational.",
    "CreationTime": "2020-01-01T23:50:50.979Z",
    "ModificationTime": "2020-01-02T23:50:50.979Z"
  }
}

```

Weitere Informationen finden Sie unter [Werte, die Sie angeben, wenn Sie eingehende Endpunkte erstellen oder bearbeiten](#), im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie unter [GetResolverEndpoint AWS CLI Befehlsreferenz](#).

get-resolver-rule-association

Das folgende Codebeispiel zeigt die Verwendung `get-resolver-rule-association`.

AWS CLI

Um Informationen über die Assoziation zwischen einer Resolver-Regel und einem zu erhalten VPC

Im folgenden `get-resolver-rule-association` Beispiel werden Details zur Zuordnung zwischen einer angegebenen Resolver-Regel und einer angezeigt. VPC Sie ordnen eine Resolver-Regel und eine VPC Verwendung zu. [associate-resolver-rule](#)

```

aws route53resolver get-resolver-rule-association \
  --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample

```

Ausgabe:

```

{
  "ResolverRuleAssociation": {
    "Id": "rslvr-rrassoc-d61cbb2c8bexample",
    "ResolverRuleId": "rslvr-rr-42b60677c0example",
    "Name": "my-resolver-rule-association",
    "VPCId": "vpc-304bexam",
    "Status": "COMPLETE",
    "StatusMessage": ""
  }
}

```

- API-Einheiten finden Sie [GetResolverRuleAssociation](#) in der AWS CLI Befehlsreferenz.

get-resolver-rule

Das folgende Codebeispiel zeigt die Verwendung `get-resolver-rule`.

AWS CLI

Um Informationen über eine Resolver-Regel abzurufen

Im folgenden `get-resolver-rule` Beispiel werden Details zur angegebenen Resolver-Regel angezeigt, z. B. der Domänenname, für den die Regel DNS-Abfragen weiterleitet, und die ID des ausgehenden Resolver-Endpunkts, dem die Regel zugeordnet ist.

```
aws route53resolver get-resolver-rule \  
  --resolver-rule-id rslvr-rr-42b60677c0example
```

Ausgabe:

```
{  
  "ResolverRule": {  
    "Id": "rslvr-rr-42b60677c0example",  
    "CreatorRequestId": "2020-01-01-18:47",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-  
rr-42b60677c0example",  
    "DomainName": "example.com.",  
    "Status": "COMPLETE",  
    "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example]  
Successfully created Resolver Rule.",  
    "RuleType": "FORWARD",  
    "Name": "my-rule",  
    "TargetIps": [  
      {  
        "Ip": "192.0.2.45",  
        "Port": 53  
      }  
    ],  
    "ResolverEndpointId": "rslvr-out-d5e5920e37example",  
    "OwnerId": "111122223333",  
    "ShareStatus": "NOT_SHARED"  
  }  
}
```


Weitere Informationen finden Sie im [Amazon Route 53 53-Entwicklerhandbuch unter Werte, die Sie beim Erstellen oder Bearbeiten von Regeln angeben](#).

- API-Einheiten finden Sie [GetResolverRule](#) unter AWS CLI Befehlsreferenz.

import-firewall-domains

Das folgende Codebeispiel zeigt die Verwendung `import-firewall-domains`.

AWS CLI

Um Domains in eine Domainliste zu importieren

Im folgenden `import-firewall-domains` Beispiel wird eine Gruppe von Domänen aus einer Datei in eine von Ihnen angegebene DNS Firewall-Domänenliste importiert.

```
aws route53resolver import-firewall-domains \  
  --firewall-domain-list-id rs1vr-fdl-d61cbb2cbexample \  
  --operation REPLACE \  
  --domain-file-url s3://PATH/TO/YOUR/FILE
```

Ausgabe:

```
{  
  "Id": "rs1vr-fdl-d61cbb2cbexample",  
  "Name": "test",  
  "Status": "IMPORTING",  
  "StatusMessage": "Importing domains from provided file."  
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer eigenen Domainlisten](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [ImportFirewallDomains](#) in der AWS CLI Befehlsreferenz.

list-firewall-configs

Das folgende Codebeispiel zeigt die Verwendung `list-firewall-configs`.

AWS CLI

Um Firewall-Konfigurationen aufzulisten

Das folgende `list-firewall-configs` Beispiel listet Ihre DNS Firewall-Konfigurationen auf.

```
aws route53resolver list-firewall-configs
```

Ausgabe:

```
{
  "FirewallConfigs": [
    {
      "Id": "rslvr-fc-86016850cexample",
      "ResourceId": "vpc-31e92222",
      "OwnerId": "123456789012",
      "FirewallFailOpen": "DISABLED"
    }
  ]
}
```

Weitere Informationen finden Sie unter [DNSVPCFirewall-Konfiguration](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [ListFirewallConfigs](#) in der AWS CLI Befehlsreferenz.

`list-firewall-domain-lists`

Das folgende Codebeispiel zeigt die Verwendung `list-firewall-domain-lists`.

AWS CLI

Um alle Route 53 Resolver DNS Firewall-Domänenlisten aufzulisten

Das folgende `list-firewall-domain-lists` Beispiel listet alle Domänenlisten auf.

```
aws route53resolver list-firewall-domain-lists
```

Ausgabe:

```
{
  "FirewallDomainLists": [
    {
      "Id": "rslvr-fdl-2c46f2ecfexample",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/rslvr-fdl-2c46f2ecfexample",
    }
  ]
}
```

```

        "Name": "AWSManagedDomainsMalwareDomainList",
        "CreatorRequestId": "AWSManagedDomainsMalwareDomainList",
        "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
    },
    {
        "Id": "rslvr-fdl-aa970e9e1example",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-
list/rslvr-fdl-aa970e9e1example",
        "Name": "AWSManagedDomainsBotnetCommandandControl",
        "CreatorRequestId": "AWSManagedDomainsBotnetCommandandControl",
        "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
    },
    {
        "Id": "rslvr-fdl-42b60677cexample",
        "Arn": "arn:aws:route53resolver:us-west-2:123456789111:firewall-domain-
list/rslvr-fdl-42b60677cexample",
        "Name": "test",
        "CreatorRequestId": "my-request-id"
    }
]
}

```

Weitere Informationen finden Sie in den [Domainlisten der Route 53 Resolver DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [ListFirewallDomainLists](#) in der AWS CLI Befehlsreferenz.

list-firewall-domains

Das folgende Codebeispiel zeigt die Verwendung `list-firewall-domains`.

AWS CLI

Um Domains in einer Domainliste aufzulisten

Im folgenden `list-firewall-domains` Beispiel werden die Domänen in einer DNS Firewall-Domänenliste aufgeführt, die Sie angeben.

```

aws route53resolver list-firewall-domains \
  --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample

```

Ausgabe:

```
{
  "Domains": [
    "test1.com.",
    "test2.com.",
    "test3.com."
  ]
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer eigenen Domainlisten](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListFirewallDomains](#) in der AWS CLI Befehlsreferenz.

list-firewall-rule-group-associations

Das folgende Codebeispiel zeigt die Verwendung `list-firewall-rule-group-associations`.

AWS CLI

Um die Zuordnungen von DNS Firewall-Regelgruppen aufzulisten

Das folgende `list-firewall-rule-group-associations` Beispiel listet Ihre DNS Firewall-Regelgruppenzuordnungen mit Amazon auf VPCs.

```
aws route53resolver list-firewall-rule-group-associations
```

Ausgabe:

```
{
  "FirewallRuleGroupAssociations": [
    {
      "Id": "rslvr-frgassoc-57e8873d7example",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
      "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
      "VpcId": "vpc-31e92222",
      "Name": "test-association",
      "Priority": 101,
      "MutationProtection": "DISABLED",
      "Status": "UPDATING",
      "StatusMessage": "Creating Firewall Rule Group Association",
      "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
    }
  ]
}
```

```
        "CreationTime": "2021-05-25T21:47:48.755768Z",
        "ModificationTime": "2021-05-25T21:47:48.755768Z"
    }
]
}
```

Weitere Informationen finden Sie unter [Verwaltung von Verknüpfungen zwischen Ihrer VPC und der Route 53 Resolver DNS Firewall-Regelgruppe](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListFirewallRuleGroupAssociations](#) in der AWS CLI Befehlsreferenz.

list-firewall-rule-groups

Das folgende Codebeispiel zeigt die Verwendung `list-firewall-rule-groups`.

AWS CLI

Um eine Liste Ihrer Firewall-Regelgruppen abzurufen

Das folgende `list-firewall-rule-groups` Beispiel listet Ihre DNS Firewall-Regelgruppen auf.

```
aws route53resolver list-firewall-rule-groups
```

Ausgabe:

```
{
  "FirewallRuleGroups": [
    {
      "Id": "rslvr-frg-47f93271fexample",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/rslvr-frg-47f93271fexample",
      "Name": "test",
      "OwnerId": "123456789012",
      "CreatorRequestId": "my-request-id",
      "ShareStatus": "NOT_SHARED"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [ListFirewallRuleGroups](#) in der AWS CLI Befehlsreferenz.

list-firewall-rules

Das folgende Codebeispiel zeigt die Verwendung `list-firewall-rules`.

AWS CLI

Um Firewallregeln aufzulisten

Im folgenden `list-firewall-rules` Beispiel werden alle Ihre DNS Firewallregeln innerhalb einer Firewall-Regelgruppe aufgeführt.

```
aws route53resolver list-firewall-rules \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Ausgabe:

```
{  
  "FirewallRules": [  
    {  
      "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",  
      "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",  
      "Name": "allow-rule",  
      "Priority": 101,  
      "Action": "ALLOW",  
      "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",  
      "CreationTime": "2021-05-25T21:44:00.346093Z",  
      "ModificationTime": "2021-05-25T21:44:00.346093Z"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [ListFirewallRules](#) in der AWS CLI Befehlsreferenz.

list-resolver-endpoint-ip-addresses

Das folgende Codebeispiel zeigt die Verwendung `list-resolver-endpoint-ip-addresses`.

AWS CLI

Um IP-Adressen für einen bestimmten eingehenden oder ausgehenden Endpunkt aufzulisten

Im folgenden `list-resolver-endpoint-ip-addresses` Beispiel werden Informationen zu den IP-Adressen aufgeführt, die dem eingehenden Endpunkt zugeordnet sind. `rslvr-in-f9ab8a03f1example` Sie können es auch `list-resolver-endpoint-ip-addresses` für ausgehende Endpunkte verwenden, indem Sie die entsprechende Endpunkt-ID angeben.

```
aws route53resolver list-resolver-endpoint-ip-addresses \  
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example
```

Ausgabe:

```
{  
  "MaxResults": 10,  
  "IpAddresses": [  
    {  
      "IpId": "rni-1de60cdbfeexample",  
      "SubnetId": "subnet-ba47exam",  
      "Ip": "192.0.2.44",  
      "Status": "ATTACHED",  
      "StatusMessage": "This IP address is operational.",  
      "CreationTime": "2020-01-03T23:02:29.587Z",  
      "ModificationTime": "2020-01-03T23:03:05.555Z"  
    },  
    {  
      "IpId": "rni-aac7085e38example",  
      "SubnetId": "subnet-12d8exam",  
      "Ip": "192.0.2.45",  
      "Status": "ATTACHED",  
      "StatusMessage": "This IP address is operational.",  
      "CreationTime": "2020-01-03T23:02:29.593Z",  
      "ModificationTime": "2020-01-03T23:02:55.060Z"  
    }  
  ]  
}
```

Weitere Informationen zu den Werten in der Ausgabe finden Sie unter [Werte, die Sie angeben, wenn Sie eingehende Endpunkte erstellen oder bearbeiten](#), und [Werte, die Sie angeben, wenn Sie ausgehende Endpunkte erstellen oder bearbeiten](#), beide im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie unter [ListResolverEndpointIpAddresses](#) Befehlsreferenz. AWS CLI

list-resolver-endpoints

Das folgende Codebeispiel zeigt die Verwendung `list-resolver-endpoints`.

AWS CLI

Um Resolver-Endpunkte in einer Region aufzulisten AWS

Im folgenden `list-resolver-endpoints` Beispiel werden die Resolver-Endpunkte für eingehende und ausgehende Verbindungen aufgeführt, die im aktuellen Konto vorhanden sind.

```
aws route53resolver list-resolver-endpoints
```

Ausgabe:

```
{
  "MaxResults": 10,
  "ResolverEndpoints": [
    {
      "Id": "rslvr-in-497098ad59example",
      "CreatorRequestId": "2020-01-01-18:47",
      "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-in-497098ad59example",
      "Name": "my-inbound-endpoint",
      "SecurityGroupIds": [
        "sg-05cd7b25d6example"
      ],
      "Direction": "INBOUND",
      "IpAddressCount": 2,
      "HostVPCId": "vpc-304bexam",
      "Status": "OPERATIONAL",
      "StatusMessage": "This Resolver Endpoint is operational.",
      "CreationTime": "2020-01-01T23:25:45.538Z",
      "ModificationTime": "2020-01-01T23:25:45.538Z"
    },
    {
      "Id": "rslvr-out-d5e5920e37example",
      "CreatorRequestId": "2020-01-01-18:48",
      "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-out-d5e5920e37example",
```



```

        "Name": "my-outbound-endpoint",
        "SecurityGroupIds": [
            "sg-05cd7b25d6example"
        ],
        "Direction": "OUTBOUND",
        "IpAddressCount": 2,
        "HostVPCId": "vpc-304bexam",
        "Status": "OPERATIONAL",
        "StatusMessage": "This Resolver Endpoint is operational.",
        "CreationTime": "2020-01-01T23:50:50.979Z",
        "ModificationTime": "2020-01-01T23:50:50.979Z"
    }
]
}

```

- API-Einheiten finden Sie unter [ListResolverEndpoints](#) Befehlsreferenz. AWS CLI

list-resolver-rule-associations

Das folgende Codebeispiel zeigt die Verwendung `list-resolver-rule-associations`.

AWS CLI

Um Verknüpfungen zwischen Resolver-Regeln und aufzulisten VPCs

Im folgenden `list-resolver-rule-associations` Beispiel werden die Verknüpfungen zwischen den Resolver-Regeln und dem aktuellen VPCs AWS Konto aufgeführt.

```
aws route53resolver list-resolver-rule-associations
```

Ausgabe:

```

{
  "MaxResults": 30,
  "ResolverRuleAssociations": [
    {
      "Id": "rslvr-autodefined-assoc-vpc-304bexam-internet-resolver",
      "ResolverRuleId": "rslvr-autodefined-rr-internet-resolver",
      "Name": "System Rule Association",
      "VPCId": "vpc-304bexam",
      "Status": "COMPLETE",
      "StatusMessage": ""
    }
  ]
}

```

```

    },
    {
      "Id": "rslvr-rrassoc-d61cbb2c8bexample",
      "ResolverRuleId": "rslvr-rr-42b60677c0example",
      "Name": "my-resolver-rule-association",
      "VPCId": "vpc-304bexam",
      "Status": "COMPLETE",
      "StatusMessage": ""
    }
  ]
}

```

Weitere Informationen finden Sie unter [So leitet Route 53 Resolver DNS Abfragen von Ihrem VPCs an Ihr Netzwerk](#) weiter im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListResolverRuleAssociations](#) in der AWS CLI Befehlsreferenz.

list-resolver-rules

Das folgende Codebeispiel zeigt die Verwendung `list-resolver-rules`.

AWS CLI

Um Resolver-Regeln aufzulisten

Das folgende `list-resolver-rules` Beispiel listet alle Resolver-Regeln im aktuellen AWS Konto auf.

```
aws route53resolver list-resolver-rules
```

Ausgabe:

```

{
  "MaxResults": 30,
  "ResolverRules": [
    {
      "Id": "rslvr-autodefined-rr-internet-resolver",
      "CreatorRequestId": "",
      "Arn": "arn:aws:route53resolver:us-west-2::autodefined-rule/rslvr-
autodefined-rr-internet-resolver",
      "DomainName": ".",
      "Status": "COMPLETE",
      "RuleType": "RECURSIVE",

```

```

    "Name": "Internet Resolver",
    "OwnerId": "Route 53 Resolver",
    "ShareStatus": "NOT_SHARED"
  },
  {
    "Id": "rslvr-rr-42b60677c0example",
    "CreatorRequestId": "2020-01-01-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0bc4e299",
    "DomainName": "example.com.",
    "Status": "COMPLETE",
    "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example]
Successfully created Resolver Rule.",
    "RuleType": "FORWARD",
    "Name": "my-rule",
    "TargetIps": [
      {
        "Ip": "192.0.2.45",
        "Port": 53
      }
    ],
    "ResolverEndpointId": "rslvr-out-d5e5920e37example",
    "OwnerId": "111122223333",
    "ShareStatus": "NOT_SHARED"
  }
]
}

```

Weitere Informationen finden Sie unter [So leitet Route 53 Resolver DNS Abfragen von Ihrem VPCs an Ihr Netzwerk](#) weiter im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListResolverRules](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Resolver-Ressource aufzulisten

Im folgenden `list-tags-for-resource` Beispiel werden die Tags aufgeführt, die der angegebenen Resolver-Regel zugewiesen sind.

```
aws route53resolver list-tags-for-resource \  
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/  
rslvr-rr-42b60677c0example"
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "my-key-1",  
      "Value": "my-value-1"  
    },  
    {  
      "Key": "my-key-2",  
      "Value": "my-value-2"  
    }  
  ]  
}
```

Informationen zur Verwendung von Tags für die Kostenzuweisung finden Sie unter [Verwenden von Kostenzuordnungs-Tags](#) im AWS Billing and Cost Management-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

put-firewall-rule-group-policy

Das folgende Codebeispiel zeigt die Verwendung `put-firewall-rule-group-policy`.

AWS CLI

Um eine Richtlinie anzuhängen, um eine AWS IAM Gruppenrichtlinie für eine Firewall-Regel gemeinsam zu nutzen

Im folgenden `put-firewall-rule-group-policy` Beispiel wird eine AWS Identity and Access Management (AWS IAM) -Richtlinie für die gemeinsame Nutzung der Regelgruppe angehängt.

```
aws route53resolver put-firewall-rule-group-policy \  
  --firewall-rule-group-policy "{\"Version\":\"2012-10-17\",  
  \"Statement\":[{\"Sid\":\"test\",\"Effect\":\"Allow\",\"Principal\br/>  \":{\"AWS\":\"arn:aws:iam::AWS_ACCOUNT_ID:root\"},\"Action\":
```

```
[\"route53resolver:GetFirewallRuleGroup\", \"route53resolver:ListFirewallRuleGroups \
\"], \"Resource\": \"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-
group/rslvr-frg-47f93271fexample\"]}]}"
```

Ausgabe:

```
{
  "ReturnValue": true
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [PutFirewallRuleGroupPolicy](#) in der AWS CLI Befehlsreferenz.

put-resolver-rule-policy

Das folgende Codebeispiel zeigt die Verwendung `put-resolver-rule-policy`.

AWS CLI

Um eine Resolver-Regel mit einem anderen AWS Konto zu teilen

Das folgende `put-resolver-rule-policy` Beispiel gibt eine Resolver-Regel an, die Sie mit einem anderen AWS Konto teilen möchten, das Konto, mit dem Sie die Regel teilen möchten, und die regelbezogenen Operationen, die das Konto für die Regeln ausführen soll.

Hinweis: Sie müssen diesen Befehl mit Anmeldeinformationen von demselben Konto ausführen, das die Regel erstellt hat.

```
aws route53resolver put-resolver-rule-policy \
  --region us-east-1 \
  --arn "arn:aws:route53resolver:us-east-1:111122223333:resolver-rule/rslvr-
rr-42b60677c0example" \
  --resolver-rule-policy "{\"Version\": \"2012-10-17\", \
    \"Statement\": [ { \
      \"Effect\" : \"Allow\", \
      \"Principal\" : {\"AWS\" : \"444455556666\" }, \
      \"Action\" : [ \
        \"route53resolver:GetResolverRule\", \
        \"route53resolver:AssociateResolverRule\", \
        \"route53resolver:DisassociateResolverRule\", \
```

```

    \"route53resolver:ListResolverRules\", \
    \"route53resolver:ListResolverRuleAssociations\" ], \
    \"Resource\" : [ \"arn:aws:route53resolver:us-east-1:111122223333:resolver-
rule/rslvr-rr-42b60677c0example\" ] } ] }"

```

Ausgabe:

```

{
  "ReturnValue": true
}

```

Nach der Ausführung `put-resolver-rule-policy` können Sie die folgenden beiden Resource Access Manager (RAM) -Befehle ausführen. Sie müssen das Konto verwenden, mit dem Sie die Regel teilen möchten:

`get-resource-share-invitations` gibt den Wert zurück `resourceShareInvitationArn`. Sie benötigen diesen Wert, um die Einladung zur Verwendung der gemeinsamen Regel anzunehmen. `accept-resource-share-invitation` akzeptiert die Einladung zur Verwendung der gemeinsamen Regel.

Weitere Informationen finden Sie in der folgenden -Dokumentation:

[get-resource-share-invitationsaccept-resource-share-invitationsWeiterleitungsregeln mit anderen AWS Konten teilen und gemeinsame Regeln verwenden](#) im Amazon Route 53 53-Entwicklerhandbuch

- API-Einheiten finden Sie [PutResolverRulePolicy](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um Tags mit einer Resolver-Ressource zu verknüpfen

Im folgenden `tag-resource` Beispiel werden zwei Tag-Schlüssel/Wert-Paare mit der angegebenen Resolver-Regel verknüpft.

```

aws route53resolver tag-resource \
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
rslvr-rr-42b60677c0example" \

```

```
--tags "Key=my-key-1,Value=my-value-1" "Key=my-key-2,Value=my-value-2"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Informationen zur Verwendung von Tags für die Kostenzuweisung finden Sie unter [Verwenden von Kostenzuordnungs-Tags](#) im AWS Billing and Cost Management-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer Resolver-Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden zwei Tags aus der angegebenen Resolver-Regel entfernt.

```
aws route53resolver untag-resource \  
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/  
  rslvr-rr-42b60677c0example" \  
  --tag-keys my-key-1 my-key-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben. Um zu bestätigen, dass die Tags entfernt wurden, können Sie verwenden [list-tags-for-resource](#).

Informationen zur Verwendung von Tags für die Kostenzuweisung finden Sie unter [Verwenden von Kostenzuordnungs-Tags](#) im AWS Billing and Cost Management-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-firewall-config

Das folgende Codebeispiel zeigt die Verwendung `update-firewall-config`.

AWS CLI

Um eine Firewall-Konfiguration zu aktualisieren

Das folgende `update-firewall-config` Beispiel aktualisiert die DNS Firewall-Konfiguration.

```
aws route53resolver update-firewall-config \  
  --resource-id vpc-31e92222 \  
  --firewall-fail-open DISABLED
```

Ausgabe:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e92222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

Weitere Informationen finden Sie unter [DNSVPCFirewall-Konfiguration](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [UpdateFirewallConfig](#) in der AWS CLI Befehlsreferenz.

update-firewall-domains

Das folgende Codebeispiel zeigt die Verwendung `update-firewall-domains`.

AWS CLI

Um eine Domainliste zu aktualisieren

Im folgenden `update-firewall-domains` Beispiel werden die Domänen einer Domänenliste mit der von Ihnen angegebenen ID hinzugefügt.

```
aws route53resolver update-firewall-domains \  
  --firewall-domain-list-id rslvr-fdl-42b60677cexampleb \  
  --operation ADD \  
  --domains test1.com test2.com test3.com
```

Ausgabe:

```
{  
  "Id": "rslvr-fdl-42b60677cexample",  
  "Name": "test",  
  "Status": "UPDATING",
```



```
"StatusMessage": "Updating the Firewall Domain List"
}
```

Weitere Informationen finden Sie unter [Verwaltung Ihrer eigenen Domainlisten](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateFirewallDomains](#) in der AWS CLI Befehlsreferenz.

update-firewall-rule-group-association

Das folgende Codebeispiel zeigt die Verwendung `update-firewall-rule-group-association`.

AWS CLI

Um eine Firewall-Regelgruppenzuordnung zu aktualisieren

Im folgenden `update-firewall-rule-group-association` Beispiel wird eine Firewall-Regelgruppenzuordnung aktualisiert.

```
aws route53resolver update-firewall-rule-group-association \
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \
  --priority 103
```

Ausgabe:

```
{
  "FirewallRuleGroupAssociation": {
    "Id": "rslvr-frgassoc-57e8873d7example",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "VpcId": "vpc-31e92222",
    "Name": "test-association",
    "Priority": 103,
    "MutationProtection": "DISABLED",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Firewall Rule Group Association Attributes",
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:47:48.755768Z",
    "ModificationTime": "2021-05-25T21:50:09.272569Z"
  }
}
```

Weitere Informationen finden Sie unter [Verwaltung von Verknüpfungen zwischen Ihrer VPC und der Route 53 Resolver DNS Firewall-Regelgruppe](#) im Amazon Route 53 53-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateFirewallRuleGroupAssociation](#) in der AWS CLI Befehlsreferenz.

update-firewall-rule

Das folgende Codebeispiel zeigt die Verwendung `update-firewall-rule`.

AWS CLI

Um eine Firewallregel zu aktualisieren

Im folgenden `update-firewall-rule` Beispiel wird eine Firewallregel mit den von Ihnen angegebenen Parametern aktualisiert.

```
aws route53resolver update-firewall-rule \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \  
  --priority 102
```

Ausgabe:

```
{  
  "FirewallRule": {  
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",  
    "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",  
    "Name": "allow-rule",  
    "Priority": 102,  
    "Action": "ALLOW",  
    "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",  
    "CreationTime": "2021-05-25T21:44:00.346093Z",  
    "ModificationTime": "2021-05-25T21:45:59.611600Z"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Regelgruppen und Regeln in der DNS Firewall](#) im Amazon Route 53 Developer Guide.

- API-Einheiten finden Sie [UpdateFirewallRule](#) in der AWS CLI Befehlsreferenz.

update-resolver-endpoint

Das folgende Codebeispiel zeigt die Verwendung `update-resolver-endpoint`.

AWS CLI

Um den Namen eines Resolver-Endpunkts zu aktualisieren

Im folgenden `update-resolver-endpoint` Beispiel wird der Name eines Resolver-Endpunkts aktualisiert. Das Aktualisieren anderer Werte wird nicht unterstützt.

```
aws route53resolver update-resolver-endpoint \  
  --resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \  
  --name my-renamed-inbound-endpoint
```

Ausgabe:

```
{  
  "ResolverEndpoint": {  
    "Id": "rslvr-in-b5d45e32bdexample",  
    "CreatorRequestId": "2020-01-02-18:48",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/  
rslvr-in-b5d45e32bdexample",  
    "Name": "my-renamed-inbound-endpoint",  
    "SecurityGroupIds": [  
      "sg-f62bexam"  
    ],  
    "Direction": "INBOUND",  
    "IpAddressCount": 2,  
    "HostVPCId": "vpc-304bexam",  
    "Status": "OPERATIONAL",  
    "StatusMessage": "This Resolver Endpoint is operational.",  
    "CreationTime": "2020-01-01T18:33:59.265Z",  
    "ModificationTime": "2020-01-08T18:33:59.265Z"  
  }  
}
```

- API-Einheiten finden Sie [UpdateResolverEndpoint](#) in der AWS CLI Befehlsreferenz.

update-resolver-rule

Das folgende Codebeispiel zeigt die Verwendung `update-resolver-rule`.

AWS CLI

Beispiel 1: Um die Einstellungen des Resolver-Endpunkts zu aktualisieren

Im folgenden `update-resolver-rule` Beispiel werden der Name der Regel, die IP-Adressen in Ihrem lokalen Netzwerk, an die DNS Anfragen weitergeleitet werden, und die ID des ausgehenden Resolver-Endpunkts aktualisiert, mit dem Sie Anfragen an Ihr Netzwerk weiterleiten.

Hinweis Bestehende Werte für `TargetIps` werden überschrieben. Sie müssen also alle IP-Adressen angeben, die die Regel nach dem Update haben soll.

```
aws route53resolver update-resolver-rule \  
  --resolver-rule-id rslvr-rr-1247fa64f3example \  
  --config Name="my-2nd-rule",TargetIps=[{Ip=192.0.2.45,Port=53},  
{Ip=192.0.2.46,Port=53}],ResolverEndpointId=rslvr-out-7b89ed0d25example
```

Ausgabe:

```
{  
  "ResolverRule": {  
    "Id": "rslvr-rr-1247fa64f3example",  
    "CreatorRequestId": "2020-01-02-18:47",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-  
rr-1247fa64f3example",  
    "DomainName": "www.example.com.",  
    "Status": "COMPLETE",  
    "StatusMessage": "[Trace id: 1-5dcc90b9-8a8ee860aba1ebd89example]  
Successfully updated Resolver Rule.",  
    "RuleType": "FORWARD",  
    "Name": "my-2nd-rule",  
    "TargetIps": [  
      {  
        "Ip": "192.0.2.45",  
        "Port": 53  
      },  
      {  
        "Ip": "192.0.2.46",  
        "Port": 53  
      }  
    ],  
    "ResolverEndpointId": "rslvr-out-7b89ed0d25example",  
    "OwnerId": "111122223333",
```

```

    "ShareStatus": "NOT_SHARED"
  }
}

```

Beispiel 2: Um die Einstellungen zu aktualisieren, verwenden Sie den Resolver-Endpunkt mithilfe einer Datei für ``config``-Einstellungen

Sie können die config Einstellungen auch in eine JSON Datei aufnehmen und diese Datei dann angeben, wenn Sie sie aufrufen. `update-resolver-rule`

```

aws route53resolver update-resolver-rule \
  --resolver-rule-id rslvr-rr-1247fa64f3example \
  --config file://c:\temp\update-resolver-rule.json

```

Inhalt von `update-resolver-rule.json`.

```

{
  "Name": "my-2nd-rule",
  "TargetIps": [
    {
      "Ip": "192.0.2.45",
      "Port": 53
    },
    {
      "Ip": "192.0.2.46",
      "Port": 53
    }
  ],
  "ResolverEndpointId": "rslvr-out-7b89ed0d25example"
}

```

Weitere Informationen finden [Sie im Amazon Route 53 53-Entwicklerhandbuch unter Werte, die Sie beim Erstellen oder Bearbeiten von Regeln angeben.](#)

- API-Einheiten finden Sie [UpdateResolverRule](#) unter AWS CLI Befehlsreferenz.

Amazon S3 S3-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon S3 Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

abort-multipart-upload

Das folgende Codebeispiel zeigt die Verwendung `abort-multipart-upload`.

AWS CLI

Um den angegebenen mehrteiligen Upload abubrechen

Mit dem folgenden `abort-multipart-upload` Befehl wird ein mehrteiliger Upload für den Schlüssel `multipart/01` im Bucket abgebrochen. `my-bucket`

```
aws s3api abort-multipart-upload \  
  --bucket my-bucket \  
  --key multipart/01 \  
  --upload-  
id dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZlJF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR
```

Die für diesen Befehl benötigte Upload-ID wird von ausgegeben `create-multipart-upload` und kann auch mit abgerufen werden. `list-multipart-uploads`

- API-Einheiten finden Sie [AbortMultipartUpload](#) in der AWS CLI Befehlsreferenz.

complete-multipart-upload

Das folgende Codebeispiel zeigt, wie man es benutzt `complete-multipart-upload`.

AWS CLI

Mit dem folgenden Befehl wird ein mehrteiliger Upload für den Schlüssel `multipart/01` im Bucket `my-bucket` abgeschlossen:

```
aws s3api complete-multipart-upload --multipart-upload file://  
mpustruct --bucket my-bucket --key 'multipart/01' --upload-  
id dfRtDYU0WwCCcH43C3WfbkR0NycyCpTJJvxu2i5GYkZlJF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR
```

Die für diesen Befehl benötigte Upload-ID wird von ausgegeben `create-multipart-upload` und kann auch mit `list-multipart-uploads` abgerufen werden.

Die mehrteilige Upload-Option im obigen Befehl hat eine JSON Struktur, die die Teile des mehrteiligen Uploads beschreibt, die wieder zur vollständigen Datei zusammengesetzt werden sollen. In diesem Beispiel wird das `file://` Präfix verwendet, um die JSON Struktur aus einer Datei im lokalen Ordner mit dem Namen zu laden. `mpustruct`

`mpustruct`:

```
{  
  "Parts": [  
    {  
      "ETag": "e868e0f4719e394144ef36531ee6824c",  
      "PartNumber": 1  
    },  
    {  
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",  
      "PartNumber": 2  
    },  
    {  
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",  
      "PartNumber": 3  
    }  
  ]  
}
```

Der ETag Wert für jeden Teil, der hochgeladen wird, wird jedes Mal ausgegeben, wenn Sie ein Teil mit dem `upload-part` Befehl hochladen. Er kann auch durch Aufrufen abgerufen `list-parts` oder berechnet werden, indem die MD5 Prüfsumme jedes Teils genommen wird.

Ausgabe:

```
{
  "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
  "Bucket": "my-bucket",
  "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
  "Key": "multipart/01"
}
```

- API-Einheiten finden Sie [CompleteMultipartUpload](#) in der AWS CLI Befehlsreferenz.

copy-object

Das folgende Codebeispiel zeigt, wie man es benutzt `copy-object`.

AWS CLI

Der folgende Befehl kopiert ein Objekt von `bucket-1` nach `bucket-2`:

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2
```

Ausgabe:

```
{
  "CopyObjectResult": {
    "LastModified": "2015-11-10T01:07:25.000Z",
    "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
  },
  "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
}
```

- API-Einheiten finden Sie [CopyObject](#) in der AWS CLI Befehlsreferenz.

cp

Das folgende Codebeispiel zeigt, wie man es benutzt `cp`.

AWS CLI

Beispiel 1: Kopieren einer lokalen Datei nach S3

Der folgende `cp` Befehl kopiert eine einzelne Datei in einen angegebenen Bucket und Schlüssel:


```
aws s3 cp test.txt s3://mybucket/test2.txt
```

Ausgabe:

```
upload: test.txt to s3://mybucket/test2.txt
```

Beispiel 2: Kopieren einer lokalen Datei mit einem Ablaufdatum nach S3

Der folgende cp Befehl kopiert eine einzelne Datei in einen angegebenen Bucket und Schlüssel, der zum angegebenen ISO 8601-Zeitstempel abläuft:

```
aws s3 cp test.txt s3://mybucket/test2.txt \  
--expires 2014-10-01T20:30:00Z
```

Ausgabe:

```
upload: test.txt to s3://mybucket/test2.txt
```

Beispiel 3: Kopieren einer Datei von S3 nach S3

Der folgende cp Befehl kopiert ein einzelnes S3-Objekt in einen angegebenen Bucket und Schlüssel:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt
```

Ausgabe:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Beispiel 4: Kopieren eines S3-Objekts in eine lokale Datei

Mit dem folgenden cp Befehl wird ein einzelnes Objekt lokal in eine angegebene Datei kopiert:

```
aws s3 cp s3://mybucket/test.txt test2.txt
```

Ausgabe:

```
download: s3://mybucket/test.txt to test2.txt
```

Beispiel 5: Kopieren eines S3-Objekts von einem Bucket in einen anderen

Der folgende `cp` Befehl kopiert ein einzelnes Objekt in einen angegebenen Bucket, wobei der ursprüngliche Name beibehalten wird:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket2/
```

Ausgabe:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

Beispiel 6: Rekursives Kopieren von S3-Objekten in ein lokales Verzeichnis

Wenn der folgende `cp` Befehl zusammen mit dem Parameter übergeben wird `--recursive`, kopiert er rekursiv alle Objekte unter einem bestimmten Präfix und Bucket in ein bestimmtes Verzeichnis. In diesem Beispiel `mybucket` enthält der Bucket die Objekte `test1.txt` und `test2.txt`:

```
aws s3 cp s3://mybucket . \
--recursive
```

Ausgabe:

```
download: s3://mybucket/test1.txt to test1.txt
download: s3://mybucket/test2.txt to test2.txt
```

Beispiel 7: Rekursives Kopieren lokaler Dateien nach S3

Wenn der folgende `cp` Befehl zusammen mit dem Parameter übergeben wird `--recursive`, kopiert er rekursiv alle Dateien in einem angegebenen Verzeichnis in einen bestimmten Bucket und ein bestimmtes Präfix, wobei einige Dateien mithilfe eines `--exclude` Parameters ausgeschlossen werden. In diesem Beispiel `myDir` enthält das Verzeichnis die Dateien `test1.txt` und `test2.jpg`:

```
aws s3 cp myDir s3://mybucket/ \
--recursive \
--exclude "*.jpg"
```

Ausgabe:

```
upload: myDir/test1.txt to s3://mybucket/test1.txt
```

Beispiel 8: Rekursives Kopieren von S3-Objekten in einen anderen Bucket

Wenn der folgende `cp` Befehl zusammen mit dem Parameter übergeben wird `--recursive`, kopiert er rekursiv alle Objekte unter einem angegebenen Bucket in einen anderen Bucket und schließt dabei einige Objekte mithilfe eines `--exclude` Parameters aus. In diesem Beispiel `mybucket` enthält der Bucket die Objekte `test1.txt` und `another/test1.txt`:

```
aws s3 cp s3://mybucket/ s3://mybucket2/ \  
  --recursive \  
  --exclude "another/*"
```

Ausgabe:

```
copy: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

Sie können `--include` Optionen kombinieren `--exclude`, um nur Objekte zu kopieren, die einem Muster entsprechen, alle anderen ausgenommen:

```
aws s3 cp s3://mybucket/logs/ s3://mybucket2/logs/ \  
  --recursive \  
  --exclude "*" \  
  --include "*.log"
```

Ausgabe:

```
copy: s3://mybucket/logs/test/test.log to s3://mybucket2/logs/test/test.log  
copy: s3://mybucket/logs/test3.log to s3://mybucket2/logs/test3.log
```

Beispiel 9: Einstellung der Zugriffskontrollliste (ACL) beim Kopieren eines S3-Objekts

Der folgende `cp` Befehl kopiert ein einzelnes Objekt in einen angegebenen Bucket und Schlüssel und setzt dabei ACL auf `public-read-write`:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt \  
  --acl public-read-write
```

Ausgabe:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Beachten Sie, dass Sie, wenn Sie die `--acl` Option verwenden, sicherstellen müssen, dass alle zugehörigen IAM Richtlinien die `"s3:PutObjectAcl"` Aktion enthalten:

```
aws iam get-user-policy \
  --user-name myuser \
  --policy-name mypolicy
```

Ausgabe:

```
{
  "UserName": "myuser",
  "PolicyName": "mypolicy",
  "PolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": [
          "s3:PutObject",
          "s3:PutObjectAcl"
        ],
        "Resource": [
          "arn:aws:s3:::mybucket/*"
        ],
        "Effect": "Allow",
        "Sid": "Stmt1234567891234"
      }
    ]
  }
}
```

Beispiel 10: Erteilen von Berechtigungen für ein S3-Objekt

Der folgende `cp` Befehl veranschaulicht die Verwendung der `--grants` Option, um allen Benutzern, die durch ihre Canonical ID identifiziert werden, Lesezugriff zu gewähren URI und einem bestimmten Benutzer, der anhand seiner Canonical ID identifiziert wird, Vollzugriff zu gewähren:

```
aws s3 cp file.txt s3://mybucket/ --grants read=uri=http://acs.amazonaws.com/groups/global/AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Ausgabe:

```
upload: file.txt to s3://mybucket/file.txt
```

Beispiel 11: Einen lokalen Dateistream auf S3 hochladen

PowerShell kann die Kodierung von Eingaben ändern oder ihr eine CRLF hinzufügen.

Der folgende cp Befehl lädt einen lokalen Dateistream von der Standardeingabe in einen bestimmten Bucket und Schlüssel hoch:

```
aws s3 cp - s3://mybucket/stream.txt
```

Beispiel 12: Hochladen eines lokalen Dateistreams, der größer als 50 GB ist, auf S3

Mit dem folgenden cp Befehl wird ein 51 GB großer lokaler Dateistream von der Standardeingabe in einen angegebenen Bucket und Schlüssel hochgeladen. Die `--expected-size` Option muss angegeben werden, andernfalls kann der Upload fehlschlagen, wenn die standardmäßige Teilelimit von 10.000 erreicht wird:

```
aws s3 cp - s3://mybucket/stream.txt --expected-size 54760833024
```

Beispiel 13: Ein S3-Objekt als lokalen Dateistream herunterladen

PowerShell kann die Kodierung der Ausgabe ändern oder eine der über die CRLF Pipeline oder Umleitung weitergeleiteten Ausgabe hinzufügen.

Mit dem folgenden cp Befehl wird ein S3-Objekt lokal als Stream in die Standardausgabe heruntergeladen. Das Herunterladen als Stream ist derzeit nicht mit dem `--recursive` Parameter kompatibel:

```
aws s3 cp s3://mybucket/stream.txt -
```

Beispiel 14: Upload auf einen S3-Zugangspunkt

Der folgende cp Befehl lädt eine einzelne Datei (`mydoc.txt`) auf den Access Point (`myaccesspoint`) am Schlüssel (`mykey`) hoch:

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

Ausgabe:

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

Beispiel 15: Herunterladen von einem S3-Zugangspunkt

Der folgende `cp` Befehl lädt ein einzelnes Objekt (`mykey`) vom Access Point (`myaccesspoint`) in die lokale Datei (`mydoc.txt`) herunter:

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey mydoc.txt
```

Ausgabe:

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to
mydoc.txt
```

- API-Einheiten finden Sie unter [Cp](#) in der AWS CLI Befehlsreferenz.

create-bucket

Das folgende Codebeispiel zeigt, wie man es benutzt `create-bucket`.

AWS CLI

Beispiel 1: Um einen Bucket zu erstellen

Im folgenden `create-bucket` Beispiel wird ein Bucket mit dem Namen `my-bucket` erstellt:

```
aws s3api create-bucket \
  --bucket my-bucket \
  --region us-east-1
```

Ausgabe:

```
{
  "Location": "/my-bucket"
}
```

Weitere Informationen finden Sie unter [Erstellen eines Buckets](#) im Amazon S3 S3-Benutzerhandbuch.

Beispiel 2: Um einen Bucket zu erstellen, bei dem der Besitzer erzwungen wird

Im folgenden `create-bucket` Beispiel wird ein Bucket mit dem Namen `my-bucket` erstellt, der die Einstellung `Bucket Owner enforced` für S3 Object Ownership verwendet.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1 \  
  --object-ownership BucketOwnerEnforced
```

Ausgabe:

```
{  
  "Location": "/my-bucket"  
}
```

Weitere Informationen finden Sie unter [Kontrolle des Besitzes von Objekten und Deaktivierung ACLs](#) im Amazon S3 S3-Benutzerhandbuch.

Beispiel 3: Um einen Bucket außerhalb der Region `us-east-1` zu erstellen

Im folgenden `create-bucket` Beispiel wird ein Bucket mit dem Namen `my-bucket eu-west-1` für Regionen außerhalb von `us-east-1` erstellt. Für Regionen außerhalb von `us-east-1` müssen die entsprechenden `LocationConstraint` Felder angegeben werden, um den Bucket in der gewünschten Region zu erstellen.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region eu-west-1 \  
  --create-bucket-configuration LocationConstraint=eu-west-1
```

Ausgabe:

```
{  
  "Location": "http://my-bucket.s3.amazonaws.com/"  
}
```

Weitere Informationen finden Sie unter [Erstellen eines Buckets](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateBucket](#) in der AWS CLI Befehlsreferenz.

create-multipart-upload

Das folgende Codebeispiel zeigt, wie man es benutzt `create-multipart-upload`.

AWS CLI

Der folgende Befehl erstellt einen mehrteiligen Upload im Bucket `my-bucket` mit dem Schlüssel `multipart/01`:

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Ausgabe:

```
{
  "Bucket": "my-bucket",
  "UploadId":
  "dfRtDYU0WwCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3URC
  "Key": "multipart/01"
}
```

Die fertige Datei wird `01` in einem Ordner namens `multipart` im Bucket `my-bucket` benannt. Speichern Sie die Upload-ID, den Schlüssel und den Bucket-Namen zur Verwendung mit dem `upload-part` Befehl.

- API-Einheiten finden Sie [CreateMultipartUpload](#) in der AWS CLI Befehlsreferenz.

delete-bucket-analytics-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-analytics-configuration`.

AWS CLI

Um eine Analytics-Konfiguration für einen Bucket zu löschen

Im folgenden `delete-bucket-analytics-configuration` Beispiel wird die Analytics-Konfiguration für den angegebenen Bucket und die angegebene ID entfernt.

```
aws s3api delete-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id my-id
```



```
--id 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBucketAnalyticsConfiguration](#) in der AWS CLI Befehlsreferenz.

delete-bucket-cors

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-cors`.

AWS CLI

Der folgende Befehl löscht eine Cross-Origin-Konfiguration für die gemeinsame Nutzung von Ressourcen aus einem Bucket mit dem Namen: `my-bucket`

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- API-Einheiten finden Sie [DeleteBucketCors](#) in der AWS CLI Befehlsreferenz.

delete-bucket-encryption

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-encryption`.

AWS CLI

Um die serverseitige Verschlüsselungskonfiguration eines Buckets zu löschen

Im folgenden `delete-bucket-encryption` Beispiel wird die serverseitige Verschlüsselungskonfiguration des angegebenen Buckets gelöscht.

```
aws s3api delete-bucket-encryption \  
  --bucket my-bucket
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteBucketEncryption AWS CLI](#) Befehlsreferenz.

delete-bucket-intelligent-tiering-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-intelligent-tiering-configuration`.

AWS CLI

Um eine S3 Intelligent-Tiering-Konfiguration aus einem Bucket zu entfernen

Im folgenden `delete-bucket-intelligent-tiering-configuration` Beispiel wird eine S3 Intelligent-Tiering-Konfiguration mit dem Namen, aus einem Bucket entfernt. `ExampleConfig`

```
aws s3api delete-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id ExampleConfig
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwenden von S3 Intelligent-Tiering](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [DeleteBucketIntelligentTieringConfiguration](#).AWS CLI

`delete-bucket-inventory-configuration`

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-inventory-configuration`.

AWS CLI

Um die Inventarkonfiguration eines Buckets zu löschen

Im folgenden `delete-bucket-inventory-configuration` Beispiel wird die Inventarkonfiguration mit der ID 1 für den angegebenen Bucket gelöscht.

```
aws s3api delete-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteBucketInventoryConfiguration AWS CLI](#) Befehlsreferenz.

`delete-bucket-lifecycle`

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-lifecycle`.

AWS CLI

Der folgende Befehl löscht eine Lebenszykluskonfiguration aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- API-Einheiten finden Sie [DeleteBucketLifecycle](#) in der AWS CLI Befehlsreferenz.

`delete-bucket-metrics-configuration`

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-metrics-configuration`.

AWS CLI

Um eine Metrikkonfiguration für einen Bucket zu löschen

Im folgenden `delete-bucket-metrics-configuration` Beispiel wird die Metrikkonfiguration für den angegebenen Bucket und die angegebene ID entfernt.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteBucketMetricsConfiguration](#) in der AWS CLI Befehlsreferenz.

`delete-bucket-ownership-controls`

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-ownership-controls`.

AWS CLI

Um die Bucket-Besitzeinstellungen eines Buckets zu entfernen

Im folgenden `delete-bucket-ownership-controls` Beispiel werden die Einstellungen für den Bucket-Besitz eines Buckets entfernt.

```
aws s3api delete-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Objektbesitz für einen vorhandenen Bucket festlegen](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteBucketOwnershipControls](#) unter AWS CLI Befehlsreferenz.

delete-bucket-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-policy`.

AWS CLI

Der folgende Befehl löscht eine Bucket-Richtlinie aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- API-Einheiten finden Sie [DeleteBucketPolicy](#) in der AWS CLI Befehlsreferenz.

delete-bucket-replication

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-replication`.

AWS CLI

Der folgende Befehl löscht eine Replikationskonfiguration aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- API-Einheiten finden Sie [DeleteBucketReplication](#) in der AWS CLI Befehlsreferenz.

delete-bucket-tagging

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-tagging`.

AWS CLI

Der folgende Befehl löscht eine Tagging-Konfiguration aus einem Bucket mit dem Namen: `my-bucket`

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- API-Einheiten finden Sie [DeleteBucketTagging](#) in der AWS CLI Befehlsreferenz.

delete-bucket-website

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket-website`.

AWS CLI

Der folgende Befehl löscht eine Website-Konfiguration aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket-website --bucket my-bucket
```

- API-Einheiten finden Sie [DeleteBucketWebsite](#) in der AWS CLI Befehlsreferenz.

delete-bucket

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-bucket`.

AWS CLI

Der folgende Befehl löscht einen Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- API-Einheiten finden Sie [DeleteBucket](#) in der AWS CLI Befehlsreferenz.

delete-object-tagging

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-object-tagging`.

AWS CLI

Um die Tag-Sets eines Objekts zu löschen

Im folgenden `delete-object-tagging` Beispiel wird das Tag mit dem angegebenen Schlüssel aus dem Objekt `doc1.rtf` gelöscht.

```
aws s3api delete-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteObjectTagging AWS CLI](#) Befehlsreferenz.

delete-object

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-object`.

AWS CLI

Der folgende Befehl löscht ein Objekt mit dem Namen `test.txt` aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Wenn die Bucket-Versionierung aktiviert ist, enthält die Ausgabe die Versions-ID der Löschmarkierung:

```
{  
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tND1w1Gq",  
  "DeleteMarker": true  
}
```

Weitere Informationen zum Löschen von Objekten finden Sie unter [Objekte löschen im Amazon S3 Developer Guide](#).

- API-Einheiten finden Sie [DeleteObject](#) in der AWS CLI Befehlsreferenz.

delete-objects

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-objects`.

AWS CLI

Der folgende Befehl löscht ein Objekt aus einem Bucket mit dem Namen `my-bucket`:

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json` ist ein JSON Dokument im aktuellen Verzeichnis, das das zu löschende Objekt angibt:

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Ausgabe:

```
{
  "Deleted": [
    {
      "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUP01koHwzU",
      "Key": "test1.txt",
      "DeleteMarker": true
    }
  ]
}
```

- API-Einheiten finden Sie [DeleteObjects](#) in der AWS CLI Befehlsreferenz.

delete-public-access-block

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-public-access-block`.

AWS CLI

Um die Konfiguration „Öffentlichen Zugriff blockieren“ für einen Bucket zu löschen

Im folgenden `delete-public-access-block` Beispiel wird die Konfiguration „Öffentlicher Zugriff blockieren“ für den angegebenen Bucket entfernt.

```
aws s3api delete-public-access-block \
  --bucket my-bucket
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

get-bucket-accelerate-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-accelerate-configuration`.

AWS CLI

Um die Accelerate-Konfiguration eines Buckets abzurufen

Im folgenden `get-bucket-accelerate-configuration` Beispiel wird die Accelerate-Konfiguration für den angegebenen Bucket abgerufen.

```
aws s3api get-bucket-accelerate-configuration \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "Status": "Enabled"  
}
```

- API-Einheiten finden Sie [GetBucketAccelerateConfiguration](#) in der AWS CLI Befehlsreferenz.

get-bucket-acl

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-acl`.

AWS CLI

Der folgende Befehl ruft die Zugriffskontrollliste für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-acl --bucket my-bucket
```

Ausgabe:

```
{  
  "Owner": {  
    "DisplayName": "my-username",
```



```

    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}

```

- API-Einheiten finden Sie [GetBucketAcl](#) in der AWS CLI Befehlsreferenz.

get-bucket-analytics-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-analytics-configuration`.

AWS CLI

Um die Analytics-Konfiguration für einen Bucket mit einer bestimmten ID abzurufen

Im folgenden `get-bucket-analytics-configuration` Beispiel wird die Analytics-Konfiguration für den angegebenen Bucket und die angegebene ID angezeigt.

```

aws s3api get-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1

```

Ausgabe:

```

{
  "AnalyticsConfiguration": {
    "StorageClassAnalysis": {},
    "Id": "1"
  }
}

```

- API-Einheiten finden Sie [GetBucketAnalyticsConfiguration](#) in der AWS CLI Befehlsreferenz.

get-bucket-cors

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-cors`.

AWS CLI

Der folgende Befehl ruft die Cross-Origin Resource Sharing-Konfiguration für einen Bucket mit dem Namen ab: `my-bucket`

```
aws s3api get-bucket-cors --bucket my-bucket
```

Ausgabe:

```
{
  "CORSRules": [
    {
      "AllowedHeaders": [
        "*"
      ],
      "ExposeHeaders": [
        "x-amz-server-side-encryption"
      ],
      "AllowedMethods": [
        "PUT",
        "POST",
        "DELETE"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedOrigins": [
        "http://www.example.com"
      ]
    },
    {
      "AllowedHeaders": [
        "Authorization"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedMethods": [
        "GET"
      ],
      "AllowedOrigins": [
        "*"
      ]
    }
  ]
}
```

```

    ]
}

```

- API-Einheiten finden Sie [GetBucketCors](#) in der AWS CLI Befehlsreferenz.

get-bucket-encryption

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-encryption`.

AWS CLI

Um die serverseitige Verschlüsselungskonfiguration für einen Bucket abzurufen

Im folgenden `get-bucket-encryption` Beispiel wird die serverseitige Verschlüsselungskonfiguration für den Bucket abgerufen. `my-bucket`

```

aws s3api get-bucket-encryption \
  --bucket my-bucket

```

Ausgabe:

```

{
  "ServerSideEncryptionConfiguration": {
    "Rules": [
      {
        "ApplyServerSideEncryptionByDefault": {
          "SSEAlgorithm": "AES256"
        }
      }
    ]
  }
}

```

- API-Einheiten finden Sie unter [GetBucketEncryption AWS CLI](#) Befehlsreferenz.

get-bucket-intelligent-tiering-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-intelligent-tiering-configuration`.

AWS CLI

Um eine S3 Intelligent-Tiering-Konfiguration für einen Bucket abzurufen

Im folgenden `get-bucket-intelligent-tiering-configuration` Beispiel wird eine S3 Intelligent-Tiering-Konfiguration mit dem Namen, für einen Bucket abgerufen. `ExampleConfig`

```
aws s3api get-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id ExampleConfig
```

Ausgabe:

```
{  
  "IntelligentTieringConfiguration": {  
    "Id": "ExampleConfig2",  
    "Filter": {  
      "Prefix": "images"  
    },  
    "Status": "Enabled",  
    "Tierings": [  
      {  
        "Days": 90,  
        "AccessTier": "ARCHIVE_ACCESS"  
      },  
      {  
        "Days": 180,  
        "AccessTier": "DEEP_ARCHIVE_ACCESS"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Verwenden von S3 Intelligent-Tiering](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [GetBucketIntelligentTieringConfiguration](#).AWS CLI

get-bucket-inventory-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-inventory-configuration`.

AWS CLI

Um die Inventarkonfiguration für einen Bucket abzurufen

Im folgenden `get-bucket-inventory-configuration` Beispiel wird die Inventarkonfiguration für den angegebenen Bucket mit der ID 1 abgerufen.

```
aws s3api get-bucket-inventory-configuration \
  --bucket my-bucket \
  --id 1
```

Ausgabe:

```
{
  "InventoryConfiguration": {
    "IsEnabled": true,
    "Destination": {
      "S3BucketDestination": {
        "Format": "ORC",
        "Bucket": "arn:aws:s3:::my-bucket",
        "AccountId": "123456789012"
      }
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
      "Frequency": "Weekly"
    }
  }
}
```

- API-Einheiten finden Sie unter [GetBucketInventoryConfiguration AWS CLI Befehlsreferenz](#).

`get-bucket-lifecycle-configuration`

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-lifecycle-configuration`.

AWS CLI

Der folgende Befehl ruft die Lebenszykluskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Ausgabe:

```
{
  "Rules": [
    {
      "ID": "Move rotated logs to Glacier",
      "Prefix": "rotated/",
      "Status": "Enabled",
      "Transitions": [
        {
          "Date": "2015-11-10T00:00:00.000Z",
          "StorageClass": "GLACIER"
        }
      ]
    },
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 0,
          "StorageClass": "GLACIER"
        }
      ],
      "ID": "Move old versions to Glacier"
    }
  ]
}
```

- API-Einheiten finden Sie [GetBucketLifecycleConfiguration](#) in der AWS CLI Befehlsreferenz.

get-bucket-lifecycle

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-lifecycle`.

AWS CLI

Der folgende Befehl ruft die Lebenszykluskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-lifecycle --bucket my-bucket
```

Ausgabe:

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
  ]
}
```

- API-Einheiten finden Sie [GetBucketLifecycle](#) in der AWS CLI Befehlsreferenz.

get-bucket-location

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-location`.

AWS CLI

Der folgende Befehl ruft die Standortbeschränkung für einen Bucket mit dem Namen `abmy-bucket`, falls eine Einschränkung existiert:

```
aws s3api get-bucket-location --bucket my-bucket
```

Ausgabe:

```
{
  "LocationConstraint": "us-west-2"
}
```

- API-Einheiten finden Sie unter [GetBucketLocation AWS CLI Befehlsreferenz](#).

get-bucket-logging

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-logging`.

AWS CLI

Um den Logging-Status für einen Bucket abzurufen

Im folgenden `get-bucket-logging` Beispiel wird der Logging-Status für den angegebenen Bucket abgerufen.

```
aws s3api get-bucket-logging \
  --bucket my-bucket
```

Ausgabe:

```
{
  "LoggingEnabled": {
    "TargetPrefix": "",
    "TargetBucket": "my-bucket-logs"
  }
}
```

- API-Einheiten finden Sie [GetBucketLogging](#) in der AWS CLI Befehlsreferenz.

get-bucket-metrics-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-metrics-configuration`.

AWS CLI

Um die Metrikkonfiguration für einen Bucket mit einer bestimmten ID abzurufen

Im folgenden `get-bucket-metrics-configuration` Beispiel wird die Metrikkonfiguration für den angegebenen Bucket und die angegebene ID angezeigt.


```
aws s3api get-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Ausgabe:

```
{  
  "MetricsConfiguration": {  
    "Filter": {  
      "Prefix": "logs"  
    },  
    "Id": "123"  
  }  
}
```

- API-Einheiten finden Sie [GetBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

get-bucket-notification-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-notification-configuration`.

AWS CLI

Der folgende Befehl ruft die Benachrichtigungskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-notification-configuration --bucket my-bucket
```

Ausgabe:

```
{  
  "TopicConfigurations": [  
    {  
      "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWw1",  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",  
      "Events": [  
        "s3:ObjectCreated:*"  
      ]  
    }  
  ]  
}
```

```
]
}
```

- API-Einheiten finden Sie [GetBucketNotificationConfiguration](#) in der AWS CLI Befehlsreferenz.

get-bucket-notification

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-notification`.

AWS CLI

Der folgende Befehl ruft die Benachrichtigungskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-notification --bucket my-bucket
```

Ausgabe:

```
{
  "TopicConfiguration": {
    "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
    "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNI",
    "Event": "s3:ObjectCreated:*",
    "Events": [
      "s3:ObjectCreated:*"
    ]
  }
}
```

- API-Einheiten finden Sie [GetBucketNotification](#) in der AWS CLI Befehlsreferenz.

get-bucket-ownership-controls

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-ownership-controls`.

AWS CLI

Um die Bucket-Besitzeinstellungen eines Buckets abzurufen

Im folgenden `get-bucket-ownership-controls` Beispiel werden die Einstellungen für den Bucket-Besitz eines Buckets abgerufen.

```
aws s3api get-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET
```

Ausgabe:

```
{  
  "OwnershipControls": {  
    "Rules": [  
      {  
        "ObjectOwnership": "BucketOwnerEnforced"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie [im Amazon S3-Benutzerhandbuch unter Objektbesitzeinstellungen für einen S3-Bucket anzeigen](#).

- API-Einheiten finden Sie [GetBucketOwnershipControls](#) unter AWS CLI Befehlsreferenz.

get-bucket-policy-status

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-policy-status`.

AWS CLI

Um den Richtlinienstatus für einen Bucket abzurufen, der angibt, ob der Bucket öffentlich ist

Im folgenden `get-bucket-policy-status` Beispiel wird der Richtlinienstatus für den Bucket `my-bucket` abgerufen.

```
aws s3api get-bucket-policy-status \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "PolicyStatus": {  
    "IsPublic": false  
  }  
}
```

```
}

```

- API-Einheiten finden Sie unter [GetBucketPolicyStatus AWS CLI-Befehlsreferenz](#).

get-bucket-policy

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-policy`.

AWS CLI

Der folgende Befehl ruft die Bucket-Richtlinie für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-policy --bucket my-bucket
```

Ausgabe:

```
{
  "Policy": "{ \"Version\": \"2008-10-17\", \"Statement\": [ { \"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/*\" }, { \"Sid\": \"\", \"Effect\": \"Deny\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/secret/*\" } ] }"
```

Get and Put a Bucket Policy Das folgende Beispiel zeigt, wie Sie eine Amazon S3 S3-Bucket-Richtlinie herunterladen, Änderungen an der Datei vornehmen und dann die geänderte Bucket-Richtlinie anwenden können. `put-bucket-policy` Um die Bucket-Richtlinie in eine Datei herunterzuladen, können Sie Folgendes ausführen:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text > policy.json
```

Anschließend können Sie die `policy.json` Datei nach Bedarf ändern. Schließlich können Sie diese geänderte Richtlinie wieder auf den S3-Bucket anwenden, indem Sie Folgendes ausführen:

`policy.json` Datei nach Bedarf. Schließlich können Sie diese geänderte Richtlinie wieder auf den S3-Bucket anwenden, indem Sie Folgendes ausführen:

Datei nach Bedarf. Schließlich können Sie diese geänderte Richtlinie wieder auf den S3-Bucket anwenden, indem Sie Folgendes ausführen:

```
aws s3api put-bucket-policy --bucket mybucket --policy file:///policy.json
```

- API-Einheiten finden Sie [GetBucketPolicy](#) in der AWS CLI Befehlsreferenz.

get-bucket-replication

Das folgende Codebeispiel zeigt, wie man es benutzt `get-bucket-replication`.

AWS CLI

Der folgende Befehl ruft die Replikationskonfiguration für einen Bucket mit dem Namen `my-bucket` ab:

```
aws s3api get-bucket-replication --bucket my-bucket
```

Ausgabe:

```
{
  "ReplicationConfiguration": {
    "Rules": [
      {
        "Status": "Enabled",
        "Prefix": "",
        "Destination": {
          "Bucket": "arn:aws:s3:::my-bucket-backup",
          "StorageClass": "STANDARD"
        },
        "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
      }
    ],
    "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
  }
}
```

- API-Einheiten finden Sie [GetBucketReplication](#) in der AWS CLI Befehlsreferenz.

get-bucket-request-payment

Das folgende Codebeispiel zeigt die Verwendung `get-bucket-request-payment`.

AWS CLI

Um die Konfiguration der Zahlungsanfrage für einen Bucket abzurufen

Im folgenden `get-bucket-request-payment` Beispiel wird die Konfiguration für Zahlungen durch den Antragsteller für den angegebenen Bucket abgerufen.

```
aws s3api get-bucket-request-payment \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "Payer": "BucketOwner"  
}
```

- API-Einheiten finden Sie [GetBucketRequestPayment](#) in der AWS CLI Befehlsreferenz.

get-bucket-tagging

Das folgende Codebeispiel zeigt die Verwendung `get-bucket-tagging`.

AWS CLI

Der folgende Befehl ruft die Tagging-Konfiguration für einen Bucket mit dem Namen `my-bucket`

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Ausgabe:

```
{  
  "TagSet": [  
    {  
      "Value": "marketing",  
      "Key": "organization"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetBucketTagging](#) in der AWS CLI Befehlsreferenz.

get-bucket-versioning

Das folgende Codebeispiel zeigt die Verwendung `get-bucket-versioning`.

AWS CLI

Mit dem folgenden Befehl wird die Versionierungskonfiguration für einen Bucket mit dem Namen `my-bucket` abgerufen:

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Ausgabe:

```
{
  "Status": "Enabled"
}
```

- API-Einheiten finden Sie [GetBucketVersioning](#) in der AWS CLI Befehlsreferenz.

get-bucket-website

Das folgende Codebeispiel zeigt die Verwendung `get-bucket-website`.

AWS CLI

Mit dem folgenden Befehl wird die statische Website-Konfiguration für einen Bucket mit dem Namen `my-bucket` abgerufen:

```
aws s3api get-bucket-website --bucket my-bucket
```

Ausgabe:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

```
}  
}
```

- API-Einheiten finden Sie [GetBucketWebsite](#) in der AWS CLI Befehlsreferenz.

get-object-acl

Das folgende Codebeispiel zeigt die Verwendung `get-object-acl`.

AWS CLI

Mit dem folgenden Befehl wird die Zugriffskontrollliste für ein Objekt in einem Bucket mit dem Namen `my-bucket` abgerufen:

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Ausgabe:

```
{  
  "Owner": {  
    "DisplayName": "my-username",  
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"  
  },  
  "Grants": [  
    {  
      "Grantee": {  
        "DisplayName": "my-username",  
        "ID":  
"7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"  
      },  
      "Permission": "FULL_CONTROL"  
    },  
    {  
      "Grantee": {  
        "URI": "http://acs.amazonaws.com/groups/global/AllUsers"  
      },  
      "Permission": "READ"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [GetObjectAcl](#) in der AWS CLI Befehlsreferenz.

get-object-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-object-attributes`.

AWS CLI

Um Metadaten von einem Objekt abzurufen, ohne das Objekt selbst zurückzugeben

Im folgenden `get-object-attributes` Beispiel werden Metadaten aus dem Objekt abgerufen.
`doc1.rtf`

```
aws s3api get-object-attributes \  
  --bucket my-bucket \  
  --key doc1.rtf \  
  --object-attributes "StorageClass" "ETag" "ObjectSize"
```

Ausgabe:

```
{  
  "LastModified": "2022-03-15T19:37:31+00:00",  
  "VersionId": "IuCPjXTDzHNf1dAuitVBIKJpF2p1fg4P",  
  "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",  
  "StorageClass": "STANDARD",  
  "ObjectSize": 405  
}
```

Weitere Informationen finden Sie [GetObjectAttributes](#) in der Amazon S3 API S3-Referenz.

- API-Einheiten finden Sie [GetObjectAttributes](#) unter AWS CLI Befehlsreferenz.

get-object-legal-hold

Das folgende Codebeispiel zeigt die Verwendung `get-object-legal-hold`.

AWS CLI

Ruft den Status „Legal Hold“ eines Objekts ab

Im folgenden `get-object-legal-hold` Beispiel wird der Status Legal Hold für das angegebene Objekt abgerufen.

```
aws s3api get-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

```
--key doc1.rtf
```

Ausgabe:

```
{
  "LegalHold": {
    "Status": "ON"
  }
}
```

- API-Einheiten finden Sie unter [GetObjectLegalHold AWS CLI](#) Befehlsreferenz.

get-object-lock-configuration

Das folgende Codebeispiel zeigt die Verwendung von `get-object-lock-configuration`.

AWS CLI

Um eine Objektsperrenkonfiguration für einen Bucket abzurufen

Im folgenden `get-object-lock-configuration` Beispiel wird die Objektsperrenkonfiguration für den angegebenen Bucket abgerufen.

```
aws s3api get-object-lock-configuration \
  --bucket my-bucket-with-object-lock
```

Ausgabe:

```
{
  "ObjectLockConfiguration": {
    "ObjectLockEnabled": "Enabled",
    "Rule": {
      "DefaultRetention": {
        "Mode": "COMPLIANCE",
        "Days": 50
      }
    }
  }
}
```

- API-Einheiten finden Sie [GetObjectLockConfiguration](#) in der AWS CLI Befehlsreferenz.

get-object-retention

Das folgende Codebeispiel zeigt die Verwendung `get-object-retention`.

AWS CLI

Um die Objektaufbewahrungskonfiguration für ein Objekt abzurufen

Im folgenden `get-object-retention` Beispiel wird die Objektaufbewahrungskonfiguration für das angegebene Objekt abgerufen.

```
aws s3api get-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

Ausgabe:

```
{  
  "Retention": {  
    "Mode": "GOVERNANCE",  
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"  
  }  
}
```

- API-Einheiten finden Sie [GetObjectRetention](#) in der AWS CLI Befehlsreferenz.

get-object-tagging

Das folgende Codebeispiel zeigt die Verwendung `get-object-tagging`.

AWS CLI

Um die an ein Objekt angehängten Tags abzurufen

Im folgenden `get-object-tagging` Beispiel werden die Werte für den angegebenen Schlüssel aus dem angegebenen Objekt abgerufen.

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Ausgabe:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    }
  ]
}
```

Im folgenden `get-object-tagging` Beispiel wird versucht, die Tagsätze des Objekts `doc2.rtf`, das keine Tags hat.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc2.rtf
```

Ausgabe:

```
{
  "TagSet": []
}
```

Im folgenden `get-object-tagging` Beispiel werden die Tagsätze des Objekts `doc3.rtf`, das über mehrere Tags verfügt.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc3.rtf
```

Ausgabe:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    },
    {
```

```
    "Value": "finance",
    "Key": "department"
  },
  {
    "Value": "payroll",
    "Key": "team"
  }
]
```

- API-Einheiten finden Sie unter [GetObjectTagging AWS CLI](#) Befehlsreferenz.

get-object-torrent

Das folgende Codebeispiel zeigt die Verwendung `get-object-torrent`.

AWS CLI

Der folgende Befehl erstellt einen Torrent für ein Objekt in einem Bucket mit dem Namen `my-bucket`:

```
aws s3api get-object-torrent --bucket my-bucket --key large-video-file.mp4 large-video-file.torrent
```

Die Torrent-Datei wird lokal im aktuellen Ordner gespeichert. Beachten Sie, dass der Ausgabedateiname (`large-video-file.torrent`) ohne Optionsnamen angegeben wird und das letzte Argument im Befehl sein muss.

- API-Einheiten finden Sie [GetObjectTorrent](#) in der AWS CLI Befehlsreferenz.

get-object

Das folgende Codebeispiel zeigt die Verwendung `get-object`.

AWS CLI

Im folgenden Beispiel wird der `get-object` Befehl verwendet, um ein Objekt von Amazon S3 herunterzuladen:

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2 my_images.tar.bz2
```

Beachten Sie, dass der Outfile-Parameter ohne einen Optionsnamen wie „--outfile“ angegeben wird. Der Name der Ausgabedatei muss der letzte Parameter im Befehl sein.

Das folgende Beispiel zeigt die Verwendung von `--range`, um einen bestimmten Bytebereich von einem Objekt herunterzuladen. Beachten Sie, dass den Bytebereichen das Präfix „bytes=“ vorangestellt werden muss:

```
aws s3api get-object --bucket text-content --key dir/my_data --  
range bytes=8888-9999 my_data_range
```

Weitere Informationen zum Abrufen von Objekten finden Sie unter Getting Objects im Amazon S3 Developer Guide.

- API-Einheiten finden Sie unter [GetObject AWS CLI Befehlsreferenz](#).

get-public-access-block

Das folgende Codebeispiel zeigt die Verwendung `get-public-access-block`.

AWS CLI

Um die Konfiguration für den öffentlichen Zugriff blockieren für einen Bucket festzulegen oder zu ändern

Das folgende `get-public-access-block` Beispiel zeigt die Konfiguration für den blockierten öffentlichen Zugriff für den angegebenen Bucket.

```
aws s3api get-public-access-block \  
--bucket my-bucket
```

Ausgabe:

```
{  
  "PublicAccessBlockConfiguration": {  
    "IgnorePublicAcls": true,  
    "BlockPublicPolicy": true,  
    "BlockPublicAcls": true,  
    "RestrictPublicBuckets": true  
  }  
}
```

- API-Einheiten finden Sie [GetPublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

head-bucket

Das folgende Codebeispiel zeigt die Verwendung `head-bucket`.

AWS CLI

Der folgende Befehl überprüft den Zugriff auf einen Bucket mit dem Namen `my-bucket`:

```
aws s3api head-bucket --bucket my-bucket
```

Wenn der Bucket existiert und Sie Zugriff darauf haben, wird keine Ausgabe zurückgegeben. Andernfalls wird eine Fehlermeldung angezeigt. Beispielsweise:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- API-Einheiten finden Sie [HeadBucket](#) in der AWS CLI Befehlsreferenz.

head-object

Das folgende Codebeispiel zeigt die Verwendung `head-object`.

AWS CLI

Der folgende Befehl ruft Metadaten für ein Objekt in einem Bucket mit dem Namen `my-bucket` ab:

```
aws s3api head-object --bucket my-bucket --key index.html
```

Ausgabe:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "text/html",
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
  "ContentLength": 77,
  "VersionId": "null",
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
  "Metadata": {}
}
```

- API-Einheiten finden Sie [HeadObject](#) in der AWS CLI Befehlsreferenz.

list-bucket-analytics-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-bucket-analytics-configurations`.

AWS CLI

Um eine Liste von Analytics-Konfigurationen für einen Bucket abzurufen

Im Folgenden wird eine Liste der Analytics-Konfigurationen für den angegebenen Bucket `list-bucket-analytics-configurations` abgerufen.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Ausgabe:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ],
  "IsTruncated": false
}
```

- API-Einheiten finden Sie [ListBucketAnalyticsConfigurations](#) in der AWS CLI Befehlsreferenz.

list-bucket-intelligent-tiering-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-bucket-intelligent-tiering-configurations`.

AWS CLI

Um alle S3 Intelligent-Tiering-Konfigurationen in einem Bucket abzurufen

Im folgenden `list-bucket-intelligent-tiering-configurations` Beispiel wird die gesamte S3 Intelligent-Tiering-Konfiguration in einem Bucket abgerufen.


```
aws s3api list-bucket-intelligent-tiering-configurations \  
--bucket DOC-EXAMPLE-BUCKET
```

Ausgabe:

```
{  
  "IsTruncated": false,  
  "IntelligentTieringConfigurationList": [  
    {  
      "Id": "ExampleConfig",  
      "Filter": {  
        "Prefix": "images"  
      },  
      "Status": "Enabled",  
      "Tierings": [  
        {  
          "Days": 90,  
          "AccessTier": "ARCHIVE_ACCESS"  
        },  
        {  
          "Days": 180,  
          "AccessTier": "DEEP_ARCHIVE_ACCESS"  
        }  
      ]  
    },  
    {  
      "Id": "ExampleConfig2",  
      "Status": "Disabled",  
      "Tierings": [  
        {  
          "Days": 730,  
          "AccessTier": "ARCHIVE_ACCESS"  
        }  
      ]  
    },  
    {  
      "Id": "ExampleConfig3",  
      "Filter": {  
        "Tag": {  
          "Key": "documents",  
          "Value": "taxes"  
        }  
      }  
    }  
  ],  
}
```

```

    "Status": "Enabled",
    "Tierings": [
      {
        "Days": 90,
        "AccessTier": "ARCHIVE_ACCESS"
      },
      {
        "Days": 365,
        "AccessTier": "DEEP_ARCHIVE_ACCESS"
      }
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Verwenden von S3 Intelligent-Tiering](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie unter Befehlsreferenz [ListBucketIntelligentTieringConfigurations](#).AWS CLI

list-bucket-inventory-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-bucket-inventory-configurations`.

AWS CLI

Um eine Liste von Inventarkonfigurationen für einen Bucket abzurufen

Das folgende `list-bucket-inventory-configurations` Beispiel listet die Inventarkonfigurationen für den angegebenen Bucket auf.

```

aws s3api list-bucket-inventory-configurations \
  --bucket my-bucket

```

Ausgabe:

```

{
  "InventoryConfigurationList": [
    {
      "IsEnabled": true,
      "Destination": {

```

```

        "S3BucketDestination": {
            "Format": "ORC",
            "Bucket": "arn:aws:s3::my-bucket",
            "AccountId": "123456789012"
        }
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
        "Frequency": "Weekly"
    }
},
{
    "IsEnabled": true,
    "Destination": {
        "S3BucketDestination": {
            "Format": "CSV",
            "Bucket": "arn:aws:s3::my-bucket",
            "AccountId": "123456789012"
        }
    },
    "IncludedObjectVersions": "Current",
    "Id": "2",
    "Schedule": {
        "Frequency": "Daily"
    }
}
],
    "IsTruncated": false
}

```

- API-Einheiten finden Sie [ListBucketInventoryConfigurations](#) in der AWS CLI Befehlsreferenz.

list-bucket-metrics-configurations

Das folgende Codebeispiel zeigt die Verwendung `list-bucket-metrics-configurations`.

AWS CLI

Um eine Liste von Metrikkonfigurationen für einen Bucket abzurufen

Im folgenden `list-bucket-metrics-configurations` Beispiel wird eine Liste von Metrikkonfigurationen für den angegebenen Bucket abgerufen.

```
aws s3api list-bucket-metrics-configurations \
  --bucket my-bucket
```

Ausgabe:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListBucketMetricsConfigurations AWS CLI Befehlsreferenz](#).

list-buckets

Das folgende Codebeispiel zeigt die Verwendung `list-buckets`.

AWS CLI

Der folgende Befehl verwendet den `list-buckets` Befehl, um die Namen aller Ihrer Amazon S3 S3-Buckets (in allen Regionen) anzuzeigen:

```
aws s3api list-buckets --query "Buckets[].Name"
```

Die Abfrageoption filtert die Ausgabe von `list-buckets` bis auf die Bucket-Namen.

Weitere Informationen zu Buckets finden Sie unter [Working with Amazon S3 Buckets](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [ListBuckets](#) in der AWS CLI Befehlsreferenz.

list-multipart-uploads

Das folgende Codebeispiel zeigt die Verwendung `list-multipart-uploads`.

AWS CLI

Der folgende Befehl listet alle aktiven mehrteiligen Uploads für einen Bucket mit dem Namen `my-bucket` auf:

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Ausgabe:

```
{
  "Uploads": [
    {
      "Initiator": {
        "DisplayName": "username",
        "ID": "arn:aws:iam::0123456789012:user/username"
      },
      "Initiated": "2015-06-02T18:01:30.000Z",
      "UploadId":
      "dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3URC
      "StorageClass": "STANDARD",
      "Key": "multipart/01",
      "Owner": {
        "DisplayName": "aws-account-name",
        "ID":
        "100719349fc3b6dcd7c820a124bf7aec408092c3d7b51b38494939801fc248b"
      }
    }
  ],
  "CommonPrefixes": []
}
```

Bei laufenden mehrteiligen Uploads fallen Speicherkosten in Amazon S3 an. Schließen Sie einen aktiven mehrteiligen Upload ab oder brechen Sie ihn ab, um seine Teile aus Ihrem Konto zu entfernen.

- API-Einheiten finden Sie [ListMultipartUploads](#) in der AWS CLI Befehlsreferenz.

list-object-versions

Das folgende Codebeispiel zeigt die Verwendung `list-object-versions`.

AWS CLI

Der folgende Befehl ruft Versionsinformationen für ein Objekt in einem Bucket mit dem Namen `my-bucket` ab:

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Ausgabe:

```
{
  "DeleteMarkers": [
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": true,
      "VersionId": "B2VsEK5saUNNHKc0AJj7hIE86RozToyq",
      "Key": "index.html",
      "LastModified": "2015-11-10T00:57:03.000Z"
    },
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": false,
      "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
      "Key": "index.html",
      "LastModified": "2015-11-09T23:32:20.000Z"
    }
  ],
  "Versions": [
    {
      "LastModified": "2015-11-10T00:20:11.000Z",
      "VersionId": "Rb_l2T8UHDkFEwCgJjhlGPOZC0qJ.vpD",
      "ETag": "\"0622528de826c0df5db1258a23b80be5\"",

```

```

    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 38
  },
  {
    "LastModified": "2015-11-09T23:26:41.000Z",
    "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAff",
    "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 38
  },
  {
    "LastModified": "2015-11-09T22:50:50.000Z",
    "VersionId": "null",
    "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
    "StorageClass": "STANDARD",
    "Key": "index.html",
    "Owner": {
      "DisplayName": "my-username",
      "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
    },
    "IsLatest": false,
    "Size": 533823
  }
]
}

```

- API-Einheiten finden Sie [ListObjectVersions](#) in der AWS CLI Befehlsreferenz.

list-objects-v2

Das folgende Codebeispiel zeigt die Verwendung `list-objects-v2`.

AWS CLI

Um eine Liste von Objekten in einem Bucket abzurufen

Das folgende `list-objects-v2` Beispiel listet die Objekte im angegebenen Bucket auf.

```
aws s3api list-objects-v2 \  
  --bucket my-bucket
```

Ausgabe:

```
{  
  "Contents": [  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"621503c373607d548b37cff8778d992c\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc1.rtf",  
      "Size": 391  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc2.rtf",  
      "Size": 373  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"08210852f65a2e9cb999972539a64d68\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc3.rtf",  
      "Size": 399  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"d1852dd683f404306569471af106988e\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc4.rtf",  
      "Size": 6225  
    }  
  ]  
}
```



```

    }
  ]
}

```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in der AWS CLI Befehlsreferenz.

list-objects

Das folgende Codebeispiel zeigt die Verwendung `list-objects`.

AWS CLI

Im folgenden Beispiel wird der `list-objects` Befehl verwendet, um die Namen aller Objekte im angegebenen Bucket anzuzeigen:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

Im Beispiel wird das `--query` Argument verwendet, um die Ausgabe von `list-objects` nach Schlüsselwert und Größe für jedes Objekt zu filtern.

Weitere Informationen zu Objekten finden Sie unter [Working with Amazon S3 Objects](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [ListObjects](#) unter AWS CLI Befehlsreferenz.

list-parts

Das folgende Codebeispiel zeigt die Verwendung `list-parts`.

AWS CLI

Der folgende Befehl listet alle Teile auf, die für einen mehrteiligen Upload mit Schlüssel `multipart/01` im Bucket `my-bucket` hochgeladen wurden:

```
aws s3api list-parts --bucket my-bucket --key 'multipart/01' --upload-id dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

Ausgabe:

```
{
```

```

"Owner": {
  "DisplayName": "aws-account-name",
  "ID": "100719349fc3b6dcd7c820a124bf7aec408092c3d7b51b38494939801fc248b"
},
"Initiator": {
  "DisplayName": "username",
  "ID": "arn:aws:iam::0123456789012:user/username"
},
"Parts": [
  {
    "LastModified": "2015-06-02T18:07:35.000Z",
    "PartNumber": 1,
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\"",
    "Size": 5242880
  },
  {
    "LastModified": "2015-06-02T18:07:42.000Z",
    "PartNumber": 2,
    "ETag": "\"6bb2b12753d66fe86da4998aa33fffb0\"",
    "Size": 5242880
  },
  {
    "LastModified": "2015-06-02T18:07:47.000Z",
    "PartNumber": 3,
    "ETag": "\"d0a0112e841abec9c9ec83406f0159c8\"",
    "Size": 5242880
  }
],
"StorageClass": "STANDARD"
}

```

- API-Einheiten finden Sie [ListParts](#) in der AWS CLI Befehlsreferenz.

ls

Das folgende Codebeispiel zeigt die Verwendung `ls`.

AWS CLI

Beispiel 1: Auflisten aller benutzereigenen Buckets

Der folgende `ls` Befehl listet alle Buckets auf, die dem Benutzer gehören. In diesem Beispiel besitzt der Benutzer die Buckets `mybucket` und `mybucket2`. Der Zeitstempel ist das Datum,

an dem der Bucket erstellt wurde. Er wird in der Zeitzone Ihres Computers angezeigt. Dieses Datum kann sich ändern, wenn Sie Änderungen an Ihrem Bucket vornehmen, z. B. wenn Sie dessen Bucket-Richtlinie bearbeiten. Beachten Sie, `s3://` dass, wenn es für das Pfadargument verwendet wird `<S3Uri>`, auch alle Buckets aufgelistet werden.

```
aws s3 ls
```

Ausgabe:

```
2013-07-11 17:08:50 mybucket
2013-07-24 14:55:44 mybucket2
```

Beispiel 2: Alle Präfixe und Objekte in einem Bucket auflisten

Der folgende `ls` Befehl listet Objekte und allgemeine Präfixe unter einem bestimmten Bucket und Präfix auf. In diesem Beispiel besitzt der Benutzer den Bucket `mybucket` mit den Objekten `test.txt` und `somePrefix/test.txt`. Die `LastWriteTime` und `Length` sind willkürlich. Beachten Sie, dass das `s3://` URI Schema nicht zur Auflösung von Mehrdeutigkeiten erforderlich ist und daher weggelassen werden kann, da der `ls` Befehl keine Interaktion mit dem lokalen Dateisystem hat.

```
aws s3 ls s3://mybucket
```

Ausgabe:

```
                PRE somePrefix/
2013-07-25 17:06:27      88 test.txt
```

Beispiel 3: Auflisten aller Präfixe und Objekte in einem bestimmten Bucket und Präfix

Der folgende `ls` Befehl listet Objekte und allgemeine Präfixe unter einem bestimmten Bucket und Präfix auf. Unter dem angegebenen Bucket und Präfix befinden sich jedoch weder Objekte noch allgemeine Präfixe.

```
aws s3 ls s3://mybucket/noExistPrefix
```

Ausgabe:

```
None
```

Beispiel 4: Rekursives Auflisten aller Präfixe und Objekte in einem Bucket

Der folgende `ls` Befehl listet Objekte in einem Bucket rekursiv auf. Anstatt `PRE dirname/` in der Ausgabe angezeigt zu werden, wird der gesamte Inhalt eines Buckets der Reihe nach aufgelistet.

```
aws s3 ls s3://mybucket \  
--recursive
```

Ausgabe:

```
2013-09-02 21:37:53      10 a.txt  
2013-09-02 21:37:53 2863288 foo.zip  
2013-09-02 21:32:57      23 foo/bar/.baz/a  
2013-09-02 21:32:58      41 foo/bar/.baz/b  
2013-09-02 21:32:57     281 foo/bar/.baz/c  
2013-09-02 21:32:57      73 foo/bar/.baz/d  
2013-09-02 21:32:57     452 foo/bar/.baz/e  
2013-09-02 21:32:57     896 foo/bar/.baz/hooks/bar  
2013-09-02 21:32:57     189 foo/bar/.baz/hooks/foo  
2013-09-02 21:32:57     398 z.txt
```

Beispiel 5: Zusammenfassung aller Präfixe und Objekte in einem Bucket

Der folgende `ls` Befehl demonstriert denselben Befehl mit den Optionen `--human-readable` und `--summarize`. `--human-readable` zeigt die Dateigröße in Bytes/MiB/KiB/GiB/TiB/PiB/EiB --`summarize` zeigt die Gesamtzahl der Objekte und die Gesamtgröße am Ende der Ergebnisliste an:

```
aws s3 ls s3://mybucket \  
--recursive \  
--human-readable \  
--summarize
```

Ausgabe:

```
2013-09-02 21:37:53  10 Bytes a.txt  
2013-09-02 21:37:53 2.9 MiB foo.zip  
2013-09-02 21:32:57  23 Bytes foo/bar/.baz/a  
2013-09-02 21:32:58  41 Bytes foo/bar/.baz/b  
2013-09-02 21:32:57 281 Bytes foo/bar/.baz/c  
2013-09-02 21:32:57  73 Bytes foo/bar/.baz/d
```

```
2013-09-02 21:32:57 452 Bytes foo/bar/.baz/e
2013-09-02 21:32:57 896 Bytes foo/bar/.baz/hooks/bar
2013-09-02 21:32:57 189 Bytes foo/bar/.baz/hooks/foo
2013-09-02 21:32:57 398 Bytes z.txt
```

```
Total Objects: 10
Total Size: 2.9 MiB
```

Beispiel 6: Auflistung von einem S3-Zugangspunkt aus

Der folgende `ls` Befehl listet Objekte von Access Point (myaccesspoint) auf:

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Ausgabe:

```
                PRE somePrefix/
2013-07-25 17:06:27      88 test.txt
```

- API-Einheiten finden Sie unter [Ls](#) in der AWS CLI Befehlsreferenz.

mb

Das folgende Codebeispiel zeigt, wie man es benutzt.

AWS CLI

Beispiel 1: Erstellen Sie einen Bucket

Der folgende `mb` Befehl erstellt einen Bucket. In diesem Beispiel erstellt der Benutzer den Bucket `mybucket`. Der Bucket wird in der Region erstellt, die in der Konfigurationsdatei des Benutzers angegeben ist:

```
aws s3 mb s3://mybucket
```

Ausgabe:

```
make_bucket: s3://mybucket
```

Beispiel 2: Erstellen Sie einen Bucket in der angegebenen Region

Der folgende `mb` Befehl erstellt einen Bucket in einer durch den `--region` Parameter angegebenen Region. In diesem Beispiel erstellt der Benutzer den Bucket `mybucket` in der Region `us-west-1`:

```
aws s3 mb s3://mybucket \  
  --region us-west-1
```

Ausgabe:

```
make_bucket: s3://mybucket
```

- API-Einheiten finden Sie unter [Mb](#) in der AWS CLI Befehlsreferenz.

mv

Das folgende Codebeispiel zeigt die Verwendung `mv`.

AWS CLI

Beispiel 1: Verschiebt eine lokale Datei in den angegebenen Bucket

Der folgende `mv` Befehl verschiebt eine einzelne Datei in einen angegebenen Bucket und Schlüssel.

```
aws s3 mv test.txt s3://mybucket/test2.txt
```

Ausgabe:

```
move: test.txt to s3://mybucket/test2.txt
```

Beispiel 2: Verschiebt ein Objekt in den angegebenen Bucket und Schlüssel

Der folgende `mv` Befehl verschiebt ein einzelnes S3-Objekt in einen angegebenen Bucket und Schlüssel.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt
```

Ausgabe:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Beispiel 3: Verschiebt ein S3-Objekt in das lokale Verzeichnis

Der folgende mv Befehl verschiebt ein einzelnes Objekt lokal in eine angegebene Datei.

```
aws s3 mv s3://mybucket/test.txt test2.txt
```

Ausgabe:

```
move: s3://mybucket/test.txt to test2.txt
```

Beispiel 4: Verschiebt ein Objekt mit seinem ursprünglichen Namen in den angegebenen Bucket

Der folgende mv Befehl verschiebt ein einzelnes Objekt in einen angegebenen Bucket, wobei der ursprüngliche Name beibehalten wird:

```
aws s3 mv s3://mybucket/test.txt s3://mybucket2/
```

Ausgabe:

```
move: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

Beispiel 5: Verschiebt alle Objekte und Präfixe in einem Bucket in das lokale Verzeichnis

Wenn der folgende mv Befehl zusammen mit dem Parameter übergeben wird `--recursive`, verschiebt er alle Objekte unter einem bestimmten Präfix und Bucket rekursiv in ein bestimmtes Verzeichnis. In diesem Beispiel mybucket enthält der Bucket die Objekte test1.txt und test2.txt.

```
aws s3 mv s3://mybucket . \
--recursive
```

Ausgabe:

```
move: s3://mybucket/test1.txt to test1.txt
move: s3://mybucket/test2.txt to test2.txt
```

Beispiel 6: Verschiebt alle Objekte und Präfixe in einem Bucket in das lokale Verzeichnis, mit Ausnahme von *.jpg-Dateien

Wenn der folgende `mv` Befehl zusammen mit dem Parameter übergeben wird `--recursive`, verschiebt er rekursiv alle Dateien in einem angegebenen Verzeichnis in einen bestimmten Bucket und ein bestimmtes Präfix, wobei einige Dateien mithilfe eines Parameters ausgeschlossen werden. `--exclude` In diesem Beispiel `myDir` enthält das Verzeichnis die Dateien `test1.txt` und `test2.jpg`.

```
aws s3 mv myDir s3://mybucket/ \  
  --recursive \  
  --exclude "*.jpg"
```

Ausgabe:

```
move: myDir/test1.txt to s3://mybucket2/test1.txt
```

Beispiel 7: Verschiebt alle Objekte und Präfixe in einem Bucket in das lokale Verzeichnis, mit Ausnahme des angegebenen Präfixes

Wenn der folgende `mv` Befehl zusammen mit dem Parameter übergeben wird `--recursive`, verschiebt er rekursiv alle Objekte unter einem angegebenen Bucket in einen anderen Bucket, wobei einige Objekte mithilfe eines `--exclude` Parameters ausgeschlossen werden. In diesem Beispiel `mybucket` enthält der Bucket die Objekte `test1.txt` und `another/test1.txt`.

```
aws s3 mv s3://mybucket/ s3://mybucket2/ \  
  --recursive \  
  --exclude "mybucket/another/*"
```

Ausgabe:

```
move: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

Beispiel 8: Verschiebt ein Objekt in den angegebenen Bucket und legt den Wert fest ACL

Mit dem folgenden `mv` Befehl wird ein einzelnes Objekt in einen bestimmten Bucket und Schlüssel verschoben, wobei der Wert ACL auf `public-read-write` gesetzt wird.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt \  
  --acl public-read-write
```

Ausgabe:


```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Beispiel 9: Verschiebt eine lokale Datei in den angegebenen Bucket und gewährt Berechtigungen

Der folgende mv Befehl veranschaulicht die Verwendung der `--grants` Option, um allen Benutzern Lesezugriff und einem bestimmten Benutzer, der anhand seiner E-Mail-Adresse identifiziert wird, Vollzugriff zu gewähren.

```
aws s3 mv file.txt s3://mybucket/ \  
  --grants read=uri=http://acs.amazonaws.com/groups/global/  
  AllUsers full=emailaddress=user@example.com
```

Ausgabe:

```
move: file.txt to s3://mybucket/file.txt
```

Beispiel 10: Verschieben Sie eine Datei auf einen S3-Zugriffspunkt

Der folgende mv Befehl verschiebt eine einzelne Datei mit dem Namen `mydoc.txt` zu dem Access Point, der mit `myaccesspoint` dem angegebenen Schlüssel benannt ist `mykey`.

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
  myaccesspoint/mykey
```

Ausgabe:

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/  
mykey
```

- API-Einheiten finden Sie unter [Mv](#) in der AWS CLI Befehlsreferenz.

presign

Das folgende Codebeispiel zeigt die Verwendung `presign`.

AWS CLI

Beispiel 1: Um eine vorsignierte Datei URL mit der Standardlebensdauer von einer Stunde zu erstellen, die auf ein Objekt in einem S3-Bucket verweist

Der folgende `presign` Befehl generiert einen vorsignierten Code URL für einen bestimmten Bucket und einen Schlüssel, der für eine Stunde gültig ist.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt
```

Ausgabe:

```
https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=EXAMPLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

Beispiel 2: Um ein vorsigniertes Objekt URL mit einer benutzerdefinierten Gültigkeitsdauer zu erstellen, das auf ein Objekt in einem S3-Bucket verweist

Der folgende `presign` Befehl generiert einen vorsignierten Code URL für einen bestimmten Bucket und einen Schlüssel, der für eine Woche gültig ist.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt \
--expires-in 604800
```

Ausgabe:

```
https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=604800&X-Amz-SignedHeaders=host&X-Amz-Signature=EXAMPLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

Weitere Informationen finden Sie unter [Ein Objekt mit anderen teilen](#) im S3 Developer Guide.

- API-Einheiten finden Sie unter [Presign](#) in der AWS CLI Befehlsreferenz.

put-bucket-accelerate-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-accelerate-configuration`.

AWS CLI

Um die Beschleunigungskonfiguration eines Buckets festzulegen

Das folgende `put-bucket-accelerate-configuration` Beispiel aktiviert die Accelerate-Konfiguration für den angegebenen Bucket.

```
aws s3api put-bucket-accelerate-configuration \  
  --bucket my-bucket \  
  --accelerate-configuration Status=Enabled
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketAccelerateConfiguration](#) in der AWS CLI Befehlsreferenz.

put-bucket-acl

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-acl`.

AWS CLI

In diesem Beispiel wird zwei AWS Benutzern (`user1@example.com` und `user2@example.com`) und allen Benutzern die `read` Erlaubnis erteilt: `full control`

```
aws s3api put-bucket-acl --bucket MyBucket --grant-full-  
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-  
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Weitere Informationen finden Sie unter <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> für Einzelheiten zu Benutzerdefiniert ACLs (die ACL `s3api`-Befehle, z. B. `put-bucket-acl`, verwenden dieselbe Kurzschreibweise für Argumente).

- API-Einheiten finden Sie in der Befehlsreferenz [PutBucketAcl](#).AWS CLI

put-bucket-analytics-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-analytics-configuration`.

AWS CLI

Um eine Analytics-Konfiguration für den Bucket festzulegen

Im folgenden `put-bucket-analytics-configuration` Beispiel werden Analysen für den angegebenen Bucket konfiguriert.

```
aws s3api put-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --analytics-configuration Configuration=Configuration
```

```
--bucket my-bucket --id 1 \  
--analytics-configuration '{"Id": "1", "StorageClassAnalysis": {}}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketAnalyticsConfiguration](#) in der AWS CLI Befehlsreferenz.

put-bucket-cors

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-cors`.

AWS CLI

Das folgende Beispiel aktiviert PUT/POST, und DELETE Anfragen von `www.example.com` und ermöglicht GET Anfragen von einer beliebigen Domain:

```
aws s3api put-bucket-cors --bucket MyBucket --cors-configuration file://cors.json
```

cors.json:

```
{  
  "CORSRules": [  
    {  
      "AllowedOrigins": ["http://www.example.com"],  
      "AllowedHeaders": ["*"],  
      "AllowedMethods": ["PUT", "POST", "DELETE"],  
      "MaxAgeSeconds": 3000,  
      "ExposeHeaders": ["x-amz-server-side-encryption"]  
    },  
    {  
      "AllowedOrigins": ["*"],  
      "AllowedHeaders": ["Authorization"],  
      "AllowedMethods": ["GET"],  
      "MaxAgeSeconds": 3000  
    }  
  ]  
}
```

- API-Einheiten finden Sie [PutBucketCors](#) in der AWS CLI Befehlsreferenz.

put-bucket-encryption

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-encryption`.

AWS CLI

Um die serverseitige Verschlüsselung für einen Bucket zu konfigurieren

Im folgenden `put-bucket-encryption` Beispiel wird AES256 Verschlüsselung als Standard für den angegebenen Bucket festgelegt.

```
aws s3api put-bucket-encryption \  
  --bucket my-bucket \  
  --server-side-encryption-configuration '{"Rules":  
  [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketEncryption](#) in der AWS CLI Befehlsreferenz.

`put-bucket-intelligent-tiering-configuration`

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-intelligent-tiering-configuration`.

AWS CLI

Um eine S3 Intelligent-Tiering-Konfiguration in einem Bucket zu aktualisieren

Das folgende `put-bucket-intelligent-tiering-configuration` Beispiel aktualisiert eine S3 Intelligent-Tiering-Konfiguration mit dem Namen, in einem Bucket. `ExampleConfig` Bei der Konfiguration werden Objekte, auf die nicht unter dem Präfix `images` zugegriffen wurde, nach 90 Tagen auf Archive Access und nach 180 Tagen auf Deep Archive Access umgestellt.

```
aws s3api put-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id "ExampleConfig" \  
  --intelligent-tiering-configuration file://intelligent-tiering-  
  configuration.json
```

Inhalt von `intelligent-tiering-configuration.json`:

```
{  
  "Id": "ExampleConfig",
```

```

    "Status": "Enabled",
    "Filter": {
      "Prefix": "images"
    },
    "Tierings": [
      {
        "Days": 90,
        "AccessTier": "ARCHIVE_ACCESS"
      },
      {
        "Days": 180,
        "AccessTier": "DEEP_ARCHIVE_ACCESS"
      }
    ]
  }
}

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Objektbesitz für einen vorhandenen Bucket festlegen](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie [PutBucketIntelligentTieringConfiguration](#) unter AWS CLI Befehlsreferenz.

put-bucket-inventory-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-inventory-configuration`.

AWS CLI

Beispiel 1: So legen Sie eine Inventarkonfiguration für einen Bucket fest

Das folgende `put-bucket-inventory-configuration` Beispiel legt einen wöchentlichen Inventarbericht im ORC -Format für den Bucket fest. `my-bucket`

```

aws s3api put-bucket-inventory-configuration \
  --bucket my-bucket \
  --id 1 \
  --inventory-configuration '{"Destination": { "S3BucketDestination":
  { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::my-bucket", "Format":
  "ORC" }}, "IsEnabled": true, "Id": "1", "IncludedObjectVersions": "Current",
  "Schedule": { "Frequency": "Weekly" } }'

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: So legen Sie eine Inventarkonfiguration für einen Bucket fest

Das folgende `put-bucket-inventory-configuration` Beispiel legt einen täglich CSV formatierten Inventarbericht für den Bucket fest. `my-bucket`

```
aws s3api put-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 2 \  
  --inventory-configuration '{"Destination": { "S3BucketDestination":  
  { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::my-bucket", "Format":  
  "CSV" }}, "IsEnabled": true, "Id": "2", "IncludedObjectVersions": "Current",  
  "Schedule": { "Frequency": "Daily" } }'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketInventoryConfiguration](#) in der AWS CLI Befehlsreferenz.

put-bucket-lifecycle-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-lifecycle-configuration`.

AWS CLI

Der folgende Befehl wendet eine Lebenszykluskonfiguration auf einen Bucket mit dem Namen `army-bucket`:

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-  
configuration file:///lifecycle.json
```

Die Datei `lifecycle.json` ist ein JSON-Dokument im aktuellen Ordner, das zwei Regeln festlegt:

```
{  
  "Rules": [  
    {  
      "ID": "Move rotated logs to Glacier",  
      "Prefix": "rotated/",  
      "Status": "Enabled",  
      "Transitions": [  

```

```

        {
            "Date": "2015-11-10T00:00:00.000Z",
            "StorageClass": "GLACIER"
        }
    ],
    },
    {
        "Status": "Enabled",
        "Prefix": "",
        "NoncurrentVersionTransitions": [
            {
                "NoncurrentDays": 2,
                "StorageClass": "GLACIER"
            }
        ],
        "ID": "Move old versions to Glacier"
    }
]
}

```

Die erste Regel verschiebt Dateien mit dem Präfix am angegebenen Datum `rotated` nach Glacier. Die zweite Regel verschiebt alte Objektversionen nach Glacier, wenn sie nicht mehr aktuell sind. Informationen zu akzeptablen Zeitstempelformaten finden Sie im AWS CLIBenutzerhandbuch unter Parameterwerte angeben.

- APIEinzelheiten finden Sie unter [PutBucketLifecycleConfiguration AWS CLI](#) Befehlsreferenz.

put-bucket-lifecycle

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-lifecycle`.

AWS CLI

Der folgende Befehl wendet eine Lebenszykluskonfiguration auf den Bucket `army-bucket`:

```
aws s3api put-bucket-lifecycle --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

Die Datei `lifecycle.json` ist ein JSON Dokument im aktuellen Ordner, das zwei Regeln festlegt:

```
{
```



```
"Rules": [  
  {  
    "ID": "Move to Glacier after sixty days (objects in logs/2015/)",  
    "Prefix": "logs/2015/",  
    "Status": "Enabled",  
    "Transition": {  
      "Days": 60,  
      "StorageClass": "GLACIER"  
    }  
  },  
  {  
    "Expiration": {  
      "Date": "2016-01-01T00:00:00.000Z"  
    },  
    "ID": "Delete 2014 logs in 2016.",  
    "Prefix": "logs/2014/",  
    "Status": "Enabled"  
  }  
]  
}
```

Die erste Regel verschiebt Dateien nach sechzig Tagen nach Amazon Glacier. Die zweite Regel löscht Dateien am angegebenen Datum aus Amazon S3. Informationen zu akzeptablen Zeitstempelformaten finden Sie unter Parameterwerte angeben im AWS CLIBenutzerhandbuch.

Jede Regel im obigen Beispiel gibt eine Richtlinie (Transition oder Expiration) und ein Dateipräfix (Ordnername) an, für die sie gilt. Sie können auch eine Regel erstellen, die für einen gesamten Bucket gilt, indem Sie ein leeres Präfix angeben:

```
{  
  "Rules": [  
    {  
      "ID": "Move to Glacier after sixty days (all objects in bucket)",  
      "Prefix": "",  
      "Status": "Enabled",  
      "Transition": {  
        "Days": 60,  
        "StorageClass": "GLACIER"  
      }  
    }  
  ]  
}
```

- API-Einheiten finden Sie [PutBucketLifecycle](#) in der AWS CLI Befehlsreferenz.

put-bucket-logging

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-logging`.

AWS CLI

Beispiel 1: So legen Sie die Bucket-Policy-Protokollierung fest

Im folgenden `put-bucket-logging` Beispiel wird die Protokollierungsrichtlinie für festgelegt `MyBucket`. Erteilen Sie zunächst mit dem `put-bucket-policy` Befehl dem Prinzipal des Logging-Dienstes die Berechtigung in Ihrer Bucket-Richtlinie.

```
aws s3api put-bucket-policy \  
  --bucket MyBucket \  
  --policy file://policy.json
```

Inhalt von `policy.json`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "S3ServerAccessLogsPolicy",  
      "Effect": "Allow",  
      "Principal": {"Service": "logging.s3.amazonaws.com"},  
      "Action": "s3:PutObject",  
      "Resource": "arn:aws:s3:::MyBucket/Logs/*",  
      "Condition": {  
        "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},  
        "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}  
      }  
    }  
  ]  
}
```

Um die Protokollierungsrichtlinie anzuwenden, verwenden Sie `put-bucket-logging`.

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --policy file://policy.json
```

```
--bucket-logging-status file://logging.json
```

Inhalt von `logging.json`:

```
{
  "LoggingEnabled": {
    "TargetBucket": "MyBucket",
    "TargetPrefix": "Logs/"
  }
}
```

Der `put-bucket-policy` Befehl ist erforderlich, um dem Prinzipal des Protokollierungsdienstes `s3:PutObject` Berechtigungen zu erteilen.

Weitere Informationen finden Sie unter [Amazon S3 Server Access Logging](#) im Amazon S3 S3-Benutzerhandbuch.

Beispiel 2: So legen Sie eine Bucket-Richtlinie für die Protokollierung des Zugriffs auf nur einen einzelnen Benutzer fest

Im folgenden `put-bucket-logging` Beispiel wird die Protokollierungsrichtlinie für festgelegt `MyBucket`. Der AWS Benutzer `bob@example.com` hat die volle Kontrolle über die Protokolldateien, und niemand sonst hat Zugriff darauf. Erteilen Sie zunächst die S3-Erlaubnis mit `put-bucket-acl`.

```
aws s3api put-bucket-acl \
  --bucket MyBucket \
  --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \
  --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

Wenden Sie dann die Protokollierungsrichtlinie mit `anput-bucket-logging`.

```
aws s3api put-bucket-logging \
  --bucket MyBucket \
  --bucket-logging-status file://logging.json
```

Inhalt von `logging.json`:

```
{
```

```

    "LoggingEnabled": {
      "TargetBucket": "MyBucket",
      "TargetPrefix": "MyBucketLogs/",
      "TargetGrants": [
        {
          "Grantee": {
            "Type": "AmazonCustomerByEmail",
            "EmailAddress": "bob@example.com"
          },
          "Permission": "FULL_CONTROL"
        }
      ]
    }
  }
}

```

Der `put-bucket-acl` Befehl ist erforderlich, um dem Protokollzustellungssystem von S3 die erforderlichen Berechtigungen (Schreib- und Lese-ACP-Berechtigungen) zu gewähren.

Weitere Informationen finden Sie unter [Amazon S3 Server Access Logging](#) im Amazon S3 Developer Guide.

- API-Einheiten finden Sie [PutBucketLogging](#) in der AWS CLI Befehlsreferenz.

put-bucket-metrics-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-metrics-configuration`.

AWS CLI

Um eine Metrikkonfiguration für einen Bucket festzulegen

Im folgenden `put-bucket-metrics-configuration` Beispiel wird eine Metrikkonfiguration mit der ID 123 für den angegebenen Bucket festgelegt.

```

aws s3api put-bucket-metrics-configuration \
  --bucket my-bucket \
  --id 123 \
  --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketMetricsConfiguration](#) unter AWS CLI Befehlsreferenz.

put-bucket-notification-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-notification-configuration`.

AWS CLI

Um die angegebenen Benachrichtigungen für einen Bucket zu aktivieren

Im folgenden `put-bucket-notification-configuration` Beispiel wird eine Benachrichtigungskonfiguration auf einen Bucket mit dem Namen `my-bucket` angewendet. Die Datei `notification.json` ist ein JSON Dokument im aktuellen Ordner, das ein SNS Thema und einen zu überwachenden Ereignistyp angibt.

```
aws s3api put-bucket-notification-configuration \  
  --bucket my-bucket \  
  --notification-configuration file://notification.json
```

Inhalt von `notification.json`:

```
{  
  "TopicConfigurations": [  
    {  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",  
      "Events": [  
        "s3:ObjectCreated:*"  
      ]  
    }  
  ]  
}
```

Dem SNS Thema muss eine IAM Richtlinie beigefügt sein, die es Amazon S3 ermöglicht, darin zu veröffentlichen.

```
{  
  "Version": "2008-10-17",  
  "Id": "example-ID",  
  "Statement": [  
    {  
      "Sid": "example-statement-ID",  
      "Effect": "Allow",  
      "Principal": {
```

```

        "Service": "s3.amazonaws.com"
    },
    "Action": [
        "SNS:Publish"
    ],
    "Resource": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
        }
    }
}
]
}

```

- API-Einheiten finden Sie [PutBucketNotificationConfiguration](#) in der AWS CLI Befehlsreferenz.

put-bucket-notification

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-notification`.

AWS CLI

Das wendet eine Benachrichtigungskonfiguration auf einen Bucket mit dem Namen `my-bucket` an:

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration file://notification.json
```

Bei der Datei `notification.json` handelt es sich um ein JSON-Dokument im aktuellen Ordner, das ein SNS-Thema und einen Ereignistyp angibt, die überwacht werden sollen:

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

Dem SNS-Thema muss eine IAM-Richtlinie beigefügt sein, die es Amazon S3 ermöglicht, darin zu veröffentlichen:

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-west-2:123456789012:my-bucket",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
        }
      }
    }
  ]
}
```

- API-Einheiten finden Sie [PutBucketNotification](#) in der AWS CLI Befehlsreferenz.

put-bucket-ownership-controls

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-ownership-controls`.

AWS CLI

Um die Bucket-Besitzeinstellungen eines Buckets zu aktualisieren

Im folgenden `put-bucket-ownership-controls` Beispiel werden die Einstellungen für den Bucket-Besitz eines Buckets aktualisiert.

```
aws s3api put-bucket-ownership-controls \
  --bucket DOC-EXAMPLE-BUCKET \
  --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Objektbesitz für einen vorhandenen Bucket festlegen](#) im Amazon S3 S3-Benutzerhandbuch.

- API-Einheiten finden Sie [PutBucketOwnershipControls](#) unter AWS CLI Befehlsreferenz.

put-bucket-policy

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-policy`.

AWS CLI

In diesem Beispiel können alle Benutzer alle Objekte in abrufen, `MyBucket` mit Ausnahme der Objekte in `MySecretFolder`. Es gewährt `put` dem Root-Benutzer des AWS Kontos auch die folgenden `delete` Berechtigungen `1234-5678-9012`:

```
aws s3api put-bucket-policy --bucket MyBucket --policy file://policy.json
```

```
policy.json:
```

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::MyBucket/*"
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::MyBucket/MySecretFolder/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": [
        "s3:DeleteObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::MyBucket/*"
    }
  ]
}
```



```
]
}
```

- API-Einheiten finden Sie [PutBucketPolicy](#) in der AWS CLI Befehlsreferenz.

put-bucket-replication

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-replication`.

AWS CLI

Um die Replikation für einen S3-Bucket zu konfigurieren

Im folgenden `put-bucket-replication` Beispiel wird eine Replikationskonfiguration auf den angegebenen S3-Bucket angewendet.

```
aws s3api put-bucket-replication \
  --bucket AWSDOC-EXAMPLE-BUCKET1 \
  --replication-configuration file://replication.json
```

Inhalt von `replication.json`:

```
{
  "Role": "arn:aws:iam::123456789012:role/s3-replication-role",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": ""},
      "Destination": {
        "Bucket": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2"
      }
    }
  ]
}
```

Für den Ziel-Bucket muss die Versionierung aktiviert sein. Die angegebene Rolle muss über Schreibberechtigungen in den Ziel-Bucket verfügen und über eine Vertrauensbeziehung verfügen, die es Amazon S3 ermöglicht, die Rolle zu übernehmen.

Beispiel für eine Rollenberechtigungsrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetReplicationConfiguration",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
      ],
      "Resource": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2/*"
    }
  ]
}
```

Beispiel für eine Vertrauensbeziehungsrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Dies ist der Thementitel](#) im Amazon Simple Storage Service Console-Benutzerhandbuch.

- API-Einheiten finden Sie [PutBucketReplication](#) unter AWS CLI Befehlsreferenz.

put-bucket-request-payment

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-request-payment`.

AWS CLI

Beispiel 1: Um die Konfiguration `requester pays` für einen Bucket zu aktivieren

Das folgende `put-bucket-request-payment` Beispiel aktiviert für den angegebenen Bucket. `requester pays`

```
aws s3api put-bucket-request-payment \
  --bucket my-bucket \
  --request-payment-configuration '{"Payer":"Requester"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um die Konfiguration `requester pays` für einen Bucket zu deaktivieren

Das folgende `put-bucket-request-payment` Beispiel deaktiviert die Option für den angegebenen Bucket. `requester pays`

```
aws s3api put-bucket-request-payment \
  --bucket my-bucket \
  --request-payment-configuration '{"Payer":"BucketOwner"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutBucketRequestPayment](#) in der AWS CLI Befehlsreferenz.

put-bucket-tagging

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-tagging`.

AWS CLI

Der folgende Befehl wendet eine Tagging-Konfiguration auf einen Bucket mit dem Namen `my-bucket` an:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

Die Datei `tagging.json` ist ein JSON-Dokument im aktuellen Ordner, das Tags angibt:

```
{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

Oder wenden Sie eine Tagging-Konfiguration `my-bucket` direkt von der Befehlszeile aus an:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
'TagSet=[{Key=organization,Value=marketing}]'
```

- API-Einheiten finden Sie [PutBucketTagging](#) in der AWS CLI Befehlsreferenz.

put-bucket-versioning

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-versioning`.

AWS CLI

Der folgende Befehl aktiviert die Versionierung für einen Bucket mit dem Namen `my-bucket`:

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled
```

Der folgende Befehl aktiviert die Versionierung und verwendet einen MFA-Code

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled --mfa "SERIAL 123456"
```

- API-Einheiten finden Sie [PutBucketVersioning](#) in der AWS CLI Befehlsreferenz.

put-bucket-website

Das folgende Codebeispiel zeigt die Verwendung `put-bucket-website`.

AWS CLI

Das wendet eine statische Website-Konfiguration auf einen Bucket mit dem Namen `army-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://website.json
```

Die Datei `website.json` ist ein JSON-Dokument im aktuellen Ordner, das Index- und Fehlerseiten für die Website angibt:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

- API-Einheiten finden Sie [PutBucketWebsite](#) unter AWS CLI Befehlsreferenz.

put-object-acl

Das folgende Codebeispiel zeigt die Verwendung `put-object-acl`.

AWS CLI

Mit dem folgenden Befehl erhalten `full control` zwei AWS Benutzer (`user1@example.com` und `user2@example.com`) Zugriff `read` auf alle Benutzer:

```
aws s3api put-object-acl --bucket MyBucket --key file.txt --grant-full-control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Weitere Informationen finden Sie unter <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> für Einzelheiten zu Benutzerdefiniert ACLs (die ACL `s3api`-Befehle, z. B. `put-object-acl`, verwenden dieselbe Kurzschreibweise für Argumente).

- API-Einheiten finden Sie in der Befehlsreferenz [PutObjectAcl](#). AWS CLI

put-object-legal-hold

Das folgende Codebeispiel zeigt die Verwendung `put-object-legal-hold`.

AWS CLI

Um einem Objekt eine gesetzliche Sperre zuzuweisen

Im folgenden `put-object-legal-hold` Beispiel wird ein Legal Hold für das Objekt `festgelegtdoc1.rtf`.

```
aws s3api put-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --legal-hold Status=ON
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutObjectLegalHold](#) unter AWS CLI Befehlsreferenz.

put-object-lock-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-object-lock-configuration`.

AWS CLI

Um eine Objektsperrenkonfiguration für einen Bucket festzulegen

Im folgenden `put-object-lock-configuration` Beispiel wird eine 50-tägige Objektsperre für den angegebenen Bucket eingerichtet.

```
aws s3api put-object-lock-configuration \  
  --bucket my-bucket-with-object-lock \  
  --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":  
  { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutObjectLockConfiguration](#) in der AWS CLI Befehlsreferenz.

put-object-retention

Das folgende Codebeispiel zeigt die Verwendung `put-object-retention`.

AWS CLI

Um eine Objektaufbewahrungskonfiguration für ein Objekt festzulegen

Im folgenden `put-object-retention` Beispiel wird eine Objektaufbewahrungskonfiguration für das angegebene Objekt bis zum 01.01.2025 festgelegt.

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter Befehlsreferenz [PutObjectRetention](#).AWS CLI

put-object-tagging

Das folgende Codebeispiel zeigt die Verwendung `put-object-tagging`.

AWS CLI

Um ein Tag für ein Objekt festzulegen

Im folgenden `put-object-tagging` Beispiel wird ein Tag mit dem Schlüssel `designation` und `confidential` dem Wert für das angegebene Objekt festgelegt.

```
aws s3api put-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf \  
  --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" } ]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Im folgenden `put-object-tagging` Beispiel werden dem angegebenen Objekt mehrere Tagsätze zugewiesen.

```
aws s3api put-object-tagging \  
  --bucket my-bucket-example \  
  --key doc3.rtf \  
  --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" },  
  { "Key": "department", "Value": "finance" }, { "Key": "team", "Value":  
  "payroll" } ]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutObjectTagging](#) unter AWS CLI Befehlsreferenz.

put-object

Das folgende Codebeispiel zeigt die Verwendung `put-object`.

AWS CLI

Im folgenden Beispiel wird der `put-object` Befehl verwendet, um ein Objekt auf Amazon S3 hochzuladen:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --  
body my_images.tar.bz2
```

Das folgende Beispiel zeigt den Upload einer Videodatei (Die Videodatei wird mithilfe der Windows-Dateisystemsyntax spezifiziert.):

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body e:  
\media\videos\f-sharp-3-data-services.mp4
```

Weitere Informationen zum Hochladen von Objekten finden Sie unter Hochladen von Objekten im Amazon S3 Developer Guide.

- API-Einheiten finden Sie unter [PutObject AWS CLI](#) Befehlsreferenz.

put-public-access-block

Das folgende Codebeispiel zeigt die Verwendung `put-public-access-block`.

AWS CLI

So legen Sie die Konfiguration für den blockierten öffentlichen Zugriff für einen Bucket fest

Im folgenden `put-public-access-block` Beispiel wird eine restriktive Konfiguration für den öffentlichen Blockzugriff für den angegebenen Bucket festgelegt.

```
aws s3api put-public-access-block \  
  --bucket my-bucket \  
  --public-access-block-  
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutPublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

rb

Das folgende Codebeispiel zeigt die Verwendung `rb`.

AWS CLI

Beispiel 1: Löschen Sie einen Bucket

Mit dem folgenden `rb` Befehl wird ein Bucket entfernt. In diesem Beispiel ist der Bucket des Benutzers `mybucket`. Beachten Sie, dass der Bucket leer sein muss, um Folgendes zu entfernen:

```
aws s3 rb s3://mybucket
```

Ausgabe:

```
remove_bucket: mybucket
```

Beispiel 2: Erzwingen Sie das Löschen eines Buckets

Der folgende `rb` Befehl verwendet den `--force` Parameter, um zuerst alle Objekte im Bucket und dann den Bucket selbst zu entfernen. In diesem Beispiel ist der Bucket des Benutzers `mybucket` und die Objekte darin `mybucket` sind `test1.txt` und `test2.txt`:

```
aws s3 rb s3://mybucket \  
  --force
```

Ausgabe:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt  
remove_bucket: mybucket
```

- API-Einheiten finden Sie unter [Rb](#) in der AWS CLI Befehlsreferenz.

restore-object

Das folgende Codebeispiel zeigt die Verwendung `restore-object`.

AWS CLI

Um eine Wiederherstellungsanforderung für ein Objekt zu erstellen

Im folgenden `restore-object` Beispiel wird das angegebene Amazon S3 Glacier-Objekt für den Bucket `my-glacier-bucket` für 10 Tage wiederhergestellt.

```
aws s3api restore-object \  
  --bucket my-glacier-bucket \  
  --key doc1.rtf \  
  --restore-request Days=10
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [RestoreObject](#) unter AWS CLI Befehlsreferenz.

rm

Das folgende Codebeispiel zeigt die Verwendung `rm`.

AWS CLI

Beispiel 1: Löschen Sie ein S3-Objekt

Der folgende `rm` Befehl löscht ein einzelnes S3-Objekt:

```
aws s3 rm s3://mybucket/test2.txt
```

Ausgabe:

```
delete: s3://mybucket/test2.txt
```

Beispiel 2: Löscht den gesamten Inhalt eines Buckets

Der folgende `rm` Befehl löscht rekursiv alle Objekte unter einem angegebenen Bucket und Präfix, wenn er mit dem Parameter übergeben wird. `--recursive` In diesem Beispiel `mybucket` enthält der Bucket die Objekte `test1.txt` und: `test2.txt`

```
aws s3 rm s3://mybucket \  
--recursive
```

Ausgabe:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt
```

Beispiel 3: Löscht alle Inhalte in einem Bucket, außer ``.jpg``-Dateien

Der folgende `rm` Befehl löscht rekursiv alle Objekte unter einem angegebenen Bucket und Präfix, wenn er mit dem Parameter übergeben wird, wobei einige Objekte mithilfe eines Parameters ausgeschlossen `--recursive` werden. `--exclude` In diesem Beispiel `mybucket` enthält der Bucket die Objekte `test1.txt` und: `test2.jpg`

```
aws s3 rm s3://mybucket/ \  
--recursive \  
--exclude "*.jpg"
```

Ausgabe:

```
delete: s3://mybucket/test1.txt
```

Beispiel 4: Löscht den gesamten Inhalt eines Buckets, mit Ausnahme von Objekten unter dem angegebenen Präfix

Der folgende `rm` Befehl löscht rekursiv alle Objekte unter einem bestimmten Bucket und Präfix, wenn er mit dem Parameter übergeben wird, `--recursive` während alle Objekte unter einem bestimmten Präfix mithilfe eines `--exclude` Parameters ausgeschlossen werden. In diesem Beispiel `mybucket` enthält der Bucket die Objekte `test1.txt` und: `another/test.txt`

```
aws s3 rm s3://mybucket/ \  
--recursive \  
--exclude "another/*"
```

Ausgabe:

```
delete: s3://mybucket/test1.txt
```

Beispiel 5: Löschen Sie ein Objekt von einem S3-Zugangspunkt

Der folgende `rm` Befehl löscht ein einzelnes Objekt (`mykey`) vom Access Point (`myaccesspoint`). :: Der folgende `rm` Befehl löscht ein einzelnes Objekt (`mykey`) vom Access Point (`myaccesspoint`).

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

Ausgabe:

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

- API-Einheiten finden Sie unter [Rm](#) in der AWS CLI Befehlsreferenz.

select-object-content

Das folgende Codebeispiel zeigt die Verwendung `select-object-content`.

AWS CLI

Um den Inhalt eines Amazon S3 S3-Objekts auf der Grundlage einer SQL-Anweisung zu filtern

Das folgende `select-object-content` Beispiel filtert das Objekt `my-data-file.csv` mit der angegebenen SQL Anweisung und sendet die Ausgabe in eine Datei.

```
aws s3api select-object-content \  
  --bucket my-bucket \  
  --key my-data-file.csv \  
  --expression "select * from s3object limit 100" \  
  --expression-type 'SQL' \  
  --input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \  
  --output-serialization '{"CSV": {}}' "output.csv"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SelectObjectContent](#) unter AWS CLI Befehlsreferenz.

sync

Das folgende Codebeispiel zeigt die Verwendung von `sync`.

AWS CLI

Beispiel 1: Synchronisieren Sie alle lokalen Objekte mit dem angegebenen Bucket

Mit dem folgenden `sync` Befehl werden Objekte aus einem lokalen Verzeichnis mit dem angegebenen Präfix und Bucket synchronisiert, indem die lokalen Dateien auf S3 hochgeladen werden. Eine lokale Datei muss hochgeladen werden, wenn sich die Größe der lokalen Datei von der Größe des S3-Objekts unterscheidet, der Zeitpunkt der letzten Änderung der lokalen Datei neuer ist als der Zeitpunkt der letzten Änderung des S3-Objekts oder die lokale Datei nicht unter dem angegebenen Bucket und Präfix existiert. In diesem Beispiel synchronisiert der Benutzer den Bucket `mybucket` mit dem aktuellen lokalen Verzeichnis. Das lokale aktuelle Verzeichnis enthält die Dateien `test.txt` und `test2.txt`. Der Bucket `mybucket` enthält keine Objekte.

```
aws s3 sync . s3://mybucket
```

Ausgabe:

```
upload: test.txt to s3://mybucket/test.txt  
upload: test2.txt to s3://mybucket/test2.txt
```

Beispiel 2: Synchronisieren Sie alle S3-Objekte aus dem angegebenen S3-Bucket mit einem anderen Bucket

Der folgende `sync` Befehl synchronisiert Objekte unter einem bestimmten Präfix und Bucket mit Objekten unter einem anderen angegebenen Präfix und Bucket, indem S3-Objekte kopiert werden. Ein S3-Objekt muss kopiert werden, wenn sich die Größen der beiden S3-Objekte unterscheiden, die Uhrzeit der letzten Änderung der Quelle neuer ist als die Uhrzeit der letzten Änderung des Ziels oder wenn das S3-Objekt unter dem angegebenen Bucket und dem Präfixziel nicht existiert.

In diesem Beispiel synchronisiert der Benutzer den Bucket mit `mybucket` dem Bucket `mybucket2`. Der Bucket `mybucket` enthält die Objekte `test.txt` und `test2.txt`. Der Bucket `mybucket2` enthält keine Objekte:

```
aws s3 sync s3://mybucket s3://mybucket2
```

Ausgabe:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
copy: s3://mybucket/test2.txt to s3://mybucket2/test2.txt
```

Beispiel 3: Synchronisieren Sie alle S3-Objekte aus dem angegebenen S3-Bucket mit dem lokalen Verzeichnis

Der folgende `sync` Befehl synchronisiert Dateien aus dem angegebenen S3-Bucket mit dem lokalen Verzeichnis, indem S3-Objekte heruntergeladen werden. Ein S3-Objekt muss heruntergeladen werden, wenn die Größe des S3-Objekts von der Größe der lokalen Datei abweicht, der Zeitpunkt der letzten Änderung des S3-Objekts neuer ist als der Zeitpunkt der letzten Änderung der lokalen Datei oder wenn das S3-Objekt nicht im lokalen Verzeichnis existiert. Beachten Sie, dass beim Herunterladen von Objekten aus S3 der Zeitpunkt der letzten Änderung der lokalen Datei auf den Zeitpunkt der letzten Änderung des S3-Objekts geändert wird. In diesem Beispiel synchronisiert der Benutzer den Bucket `mybucket` mit dem aktuellen lokalen Verzeichnis. Der Bucket `mybucket` enthält die Objekte `test.txt` und `test2.txt`. Das aktuelle lokale Verzeichnis enthält keine Dateien:

```
aws s3 sync s3://mybucket .
```

Ausgabe:

```
download: s3://mybucket/test.txt to test.txt
```

```
download: s3://mybucket/test2.txt to test2.txt
```

Beispiel 4: Synchronisieren Sie alle lokalen Objekte mit dem angegebenen Bucket und löschen Sie alle Dateien, die nicht übereinstimmen

Der folgende sync Befehl synchronisiert Objekte unter einem bestimmten Präfix und Bucket mit Dateien in einem lokalen Verzeichnis, indem die lokalen Dateien auf S3 hochgeladen werden. Aufgrund des `--delete` Parameters werden alle Dateien gelöscht, die unter dem angegebenen Präfix und Bucket, aber nicht im lokalen Verzeichnis vorhanden sind. In diesem Beispiel synchronisiert der Benutzer den Bucket mybucket mit dem lokalen aktuellen Verzeichnis. Das lokale aktuelle Verzeichnis enthält die Dateien `test.txt` und `test2.txt`. Der Bucket mybucket enthält das Objekt `test3.txt`:

```
aws s3 sync . s3://mybucket \  
--delete
```

Ausgabe:

```
upload: test.txt to s3://mybucket/test.txt  
upload: test2.txt to s3://mybucket/test2.txt  
delete: s3://mybucket/test3.txt
```

Beispiel 5: Synchronisiert alle lokalen Objekte mit Ausnahme von ``.jpg``-Dateien mit dem angegebenen Bucket

Der folgende sync Befehl synchronisiert Objekte unter einem bestimmten Präfix und Bucket mit Dateien in einem lokalen Verzeichnis, indem die lokalen Dateien auf S3 hochgeladen werden. Aufgrund des `--exclude` Parameters werden alle Dateien, die dem Muster entsprechen, das sowohl in S3 als auch lokal existiert, von der Synchronisierung ausgeschlossen. In diesem Beispiel synchronisiert der Benutzer den Bucket mybucket mit dem lokalen aktuellen Verzeichnis. Das lokale aktuelle Verzeichnis enthält die Dateien `test.jpg` und `test2.txt`. Der Bucket mybucket enthält das Objekt mit `test.jpg` einer anderen Größe als das lokale `test.jpg`:

```
aws s3 sync . s3://mybucket \  
--exclude "*.jpg"
```

Ausgabe:

```
upload: test2.txt to s3://mybucket/test2.txt
```

Beispiel 6: Synchronisiert alle lokalen Objekte mit Ausnahme von ``.jpg``-Dateien mit dem angegebenen Bucket

Der folgende `sync` Befehl synchronisiert Dateien in einem lokalen Verzeichnis mit Objekten unter einem bestimmten Präfix und Bucket, indem S3-Objekte heruntergeladen werden. In diesem Beispiel werden das `--exclude` Parameter-Flag verwendet, um ein bestimmtes Verzeichnis und ein S3-Präfix aus dem `sync` Befehl auszuschließen. In diesem Beispiel synchronisiert der Benutzer das aktuelle lokale Verzeichnis mit dem Bucket `mybucket`. Das lokale aktuelle Verzeichnis enthält die Dateien `test.txt` und `another/test2.txt`. Der Bucket `mybucket` enthält die Objekte `another/test5.txt` und `test1.txt`:

```
aws s3 sync s3://mybucket/ . \  
--exclude "*another/*"
```

Ausgabe:

```
download: s3://mybucket/test1.txt to test1.txt
```

Beispiel 7: Synchronisieren Sie alle Objekte zwischen Buckets in verschiedenen Regionen

Der folgende `sync` Befehl synchronisiert Dateien zwischen zwei Buckets in verschiedenen Regionen:

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \  
--source-region us-west-2 \  
--region us-east-1
```

Ausgabe:

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

Beispiel 8: Synchronisieren Sie mit einem S3-Zugriffspunkt

Der folgende `sync` Befehl synchronisiert das aktuelle Verzeichnis mit dem Access Point (`myaccesspoint`):


```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Ausgabe:

```
upload: test.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test.txt
upload: test2.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test2.txt
```

- API-Einheiten finden Sie unter [Sync](#) in der AWS CLI Befehlsreferenz.

upload-part-copy

Das folgende Codebeispiel zeigt die Verwendung `upload-part-copy`.

AWS CLI

Um einen Teil eines Objekts hochzuladen, indem Daten aus einem vorhandenen Objekt als Datenquelle kopiert werden

Im folgenden `upload-part-copy` Beispiel wird ein Teil hochgeladen, indem Daten aus einem vorhandenen Objekt als Datenquelle kopiert werden.

```
aws s3api upload-part-copy \
  --bucket my-bucket \
  --key "Map_Data_June.mp4" \
  --copy-source "my-bucket/copy_of_Map_Data_June.mp4" \
  --part-number 1 \
  --upload-
id "bq0tdE1CDpWQYRPLHuNG50xAT6pA5D.m_RiBy0gg0H6b13pVRY7QjvLlf75iFdJqp_2wztk5hvpUM2SesXgrzbeh"
```

Ausgabe:

```
{
  "CopyPartResult": {
    "LastModified": "2019-12-13T23:16:03.000Z",
    "ETag": "\"711470fc377698c393d94aed6305e245\""
  }
}
```

- API-Einheiten finden Sie unter [UploadPartCopy AWS CLI](#) Befehlsreferenz.

upload-part

Das folgende Codebeispiel zeigt die Verwendung `upload-part`.

AWS CLI

Der folgende Befehl lädt den ersten Teil eines mehrteiligen Uploads hoch, der `create-multipart-upload` mit dem Befehl initiiert wurde:

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --  
body part01 --upload-id  
"dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR"
```

Die `body` Option verwendet den Namen oder Pfad einer lokalen Datei für den Upload (verwenden Sie nicht das Präfix `file://`). Die Mindestteilgröße beträgt 5 MB. Die Upload-ID wird von zurückgegeben `create-multipart-upload` und kann auch mit abgerufen werden `list-multipart-uploads`. Bucket und Schlüssel werden angegeben, wenn Sie den mehrteiligen Upload erstellen.

Ausgabe:

```
{  
  "ETag": "\"e868e0f4719e394144ef36531ee6824c\""  
}
```

Speichern Sie den ETag Wert jedes Teils für später. Sie sind erforderlich, um den mehrteiligen Upload abzuschließen.

- API-Einheiten finden Sie [UploadPart](#) in der AWS CLI Befehlsreferenz.

website

Das folgende Codebeispiel zeigt die Verwendung `website`.

AWS CLI

Konfigurieren Sie einen S3-Bucket als statische Website

Der folgende Befehl konfiguriert einen Bucket, der `my-bucket` als statische Website benannt ist. Die Option `Dokument indexieren` gibt die Datei `index.html` an, zu der Besucher weitergeleitet werden, wenn sie die Website URL aufrufen. In diesem Fall befindet sich der Bucket in der Region `US-West-2`, sodass die Site unter `http://my-bucket.s3-website-us-west-2.amazonaws.com` angezeigt würde.

Alle Dateien im Bucket, die auf der statischen Site angezeigt werden, müssen so konfiguriert sein, dass Besucher sie öffnen können. Dateiberechtigungen werden getrennt von der Konfiguration der Bucket-Website konfiguriert.

```
aws s3 website s3://my-bucket/ \  
  --index-document index.html \  
  --error-document error.html
```

Informationen zum Hosten einer statischen Website in Amazon S3 finden Sie unter [Hosten einer statischen Website](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie unter [Website](#) in der AWS CLI Befehlsreferenz.

Beispiele für Amazon S3 Control mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon S3 Control Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-access-point

Das folgende Codebeispiel zeigt die Verwendung `create-access-point`.

AWS CLI

Um einen Access Point zu erstellen

Im folgenden `create-access-point` Beispiel wird ein Access Point erstellt, der `finance-ap` nach dem Bucket `business-records` im Konto `123456789012` benannt ist. Bevor Sie dieses Beispiel ausführen, ersetzen Sie den Namen des Zugriffspunkts, den Bucket-Namen und die Kontonummer durch die entsprechenden Werte für Ihren Anwendungsfall.

```
aws s3control create-access-point \  
  --account-id 123456789012 \  
  --bucket business-records \  
  --name finance-ap
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Creating Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [CreateAccessPoint](#) in der AWS CLI Befehlsreferenz.

create-job

Das folgende Codebeispiel zeigt die Verwendung `create-job`.

AWS CLI

So erstellen Sie einen Amazon S3 S3-Auftrag für Batch-Operationen

Das folgende `create-job` Beispiel erstellt einen Amazon S3 S3-Auftrag für Batch-Operationen, um Objekte als `confidential` in the bucket `employee-records`.

```
aws s3control create-job \  
  --account-id 123456789012 \  
  --operation '{"S3PutObjectTagging": { "TagSet": [{"Key": "confidential",  
  "Value": "true"}] }}' \  
  --report '{"Bucket": "arn:aws:s3:::employee-records-logs", "Prefix": "batch-op-  
create-job",  
  "Format": "Report_CSV_20180820", "Enabled": true, "ReportScope": "AllTasks"}' \  
  --manifest '{"Spec": {"Format": "S3BatchOperations_CSV_20180820", "Fields":  
  ["Bucket", "Key"]}, "Location": {"ObjectArn": "arn:aws:s3:::employee-records-logs/inv-
```

```
report/7a6a9be4-072c-407e-85a2-ec3e982f773e.csv", "ETag": "69f52a4e9f797e987155d9c8f5880897"}}' \  
--priority 42 \  
--role-arn arn:aws:iam::123456789012:role/S3BatchJobRole
```

Ausgabe:

```
{  
  "JobId": "93735294-df46-44d5-8638-6356f335324e"  
}
```

- API-Einheiten finden Sie [CreateJob](#) in der AWS CLI Befehlsreferenz.

delete-access-point-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-access-point-policy`.

AWS CLI

Um eine Zugriffspunktrichtlinie zu löschen

Im folgenden `delete-access-point-policy` Beispiel wird die Zugriffspunktrichtlinie von dem `finance-ap` im Konto `123456789012` genannten Zugriffspunkt gelöscht. Bevor Sie dieses Beispiel ausführen, ersetzen Sie den Namen und die Kontonummer des Zugriffspunkts durch die entsprechenden Werte für Ihren Anwendungsfall.

```
aws s3control delete-access-point-policy \  
--account-id 123456789012 \  
--name finance-ap
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung des Datenzugriffs mit Amazon S3 Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [DeleteAccessPointPolicy](#) in der AWS CLI Befehlsreferenz.

delete-access-point

Das folgende Codebeispiel zeigt die Verwendung `delete-access-point`.

AWS CLI

Um einen Access Point zu löschen

Im folgenden `delete-access-point` Beispiel wird ein Access Point mit dem Namen `finance-ap 123456789012` gelöscht. Bevor Sie dieses Beispiel ausführen, ersetzen Sie den Namen und die Kontonummer des Zugriffspunkts durch die entsprechenden Werte für Ihren Anwendungsfall.

```
aws s3control delete-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung des Datenzugriffs mit Amazon S3 Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [DeleteAccessPoint](#) in der AWS CLI Befehlsreferenz.

`delete-public-access-block`

Das folgende Codebeispiel zeigt die Verwendung `delete-public-access-block`.

AWS CLI

Um die Einstellungen für den öffentlichen Zugriff zu sperren, für ein Konto zu löschen

Im folgenden `delete-public-access-block` Beispiel werden die Einstellungen zum Blockieren des öffentlichen Zugriffs für das angegebene Konto gelöscht.

```
aws s3control delete-public-access-block \  
  --account-id 123456789012
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

`describe-job`

Das folgende Codebeispiel zeigt die Verwendung `describe-job`.

AWS CLI

Um einen Amazon S3 S3-Auftrag für Batch-Operationen zu beschreiben

Im Folgenden finden `describe-job` Sie die Konfigurationsparameter und den Status für den angegebenen Batch-Operationsauftrag.

```
aws s3control describe-job \  
  --account-id 123456789012 \  
  --job-id 93735294-df46-44d5-8638-6356f335324e
```

Ausgabe:

```
{  
  "Job": {  
    "TerminationDate": "2019-10-03T21:49:53.944Z",  
    "JobId": "93735294-df46-44d5-8638-6356f335324e",  
    "FailureReasons": [],  
    "Manifest": {  
      "Spec": {  
        "Fields": [  
          "Bucket",  
          "Key"  
        ],  
        "Format": "S3BatchOperations_CSV_20180820"  
      },  
      "Location": {  
        "ETag": "69f52a4e9f797e987155d9c8f5880897",  
        "ObjectArn": "arn:aws:s3:::employee-records-logs/inv-report/7a6a9be4-072c-407e-85a2-ec3e982f773e.csv"  
      }  
    },  
    "Operation": {  
      "S3PutObjectTagging": {  
        "TagSet": [  
          {  
            "Value": "true",  
            "Key": "confidential"  
          }  
        ]  
      }  
    },  
    "RoleArn": "arn:aws:iam::123456789012:role/S3BatchJobRole",
```

```

    "ProgressSummary": {
      "TotalNumberOfTasks": 8,
      "NumberOfTasksFailed": 0,
      "NumberOfTasksSucceeded": 8
    },
    "Priority": 42,
    "Report": {
      "ReportScope": "AllTasks",
      "Format": "Report_CSV_20180820",
      "Enabled": true,
      "Prefix": "batch-op-create-job",
      "Bucket": "arn:aws:s3:::employee-records-logs"
    },
    "JobArn": "arn:aws:s3:us-west-2:123456789012:job/93735294-
df46-44d5-8638-6356f335324e",
    "CreationTime": "2019-10-03T21:48:48.048Z",
    "Status": "Complete"
  }
}

```

- API-Einheiten finden Sie [DescribeJob](#) in der AWS CLI Befehlsreferenz.

get-access-point-policy-status

Das folgende Codebeispiel zeigt die Verwendung `get-access-point-policy-status`.

AWS CLI

Um den Status der Access Point-Richtlinie abzurufen

Im folgenden `get-access-point-policy-status` Beispiel wird der Status der Zugriffspunktrichtlinie für den Access Point abgerufen, der `finance-ap` im Konto `123456789012` benannt ist. Der Status der Zugriffspunkt-Richtlinie gibt an, ob die Richtlinie des Zugriffspunkts öffentlichen Zugriff zulässt. Bevor Sie dieses Beispiel ausführen, ersetzen Sie den Namen und die Kontonummer des Access Points durch die entsprechenden Werte für Ihren Anwendungsfall.

```

aws s3control get-access-point-policy-status \
  --account-id 123456789012 \
  --name finance-ap

```

Ausgabe:


```
{
  "PolicyStatus": {
    "IsPublic": false
  }
}
```

Weitere Informationen darüber, wann eine Zugriffspunktrichtlinie als öffentlich betrachtet wird, finden Sie unter [Die Bedeutung von „öffentlich“](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [GetAccessPointPolicyStatus](#) in der AWS CLI Befehlsreferenz.

get-access-point-policy

Das folgende Codebeispiel zeigt die Verwendung `get-access-point-policy`.

AWS CLI

Um eine Zugriffspunktrichtlinie abzurufen

Im folgenden `get-access-point-policy` Beispiel wird die Zugriffspunktrichtlinie von dem `finance-ap` im Konto `123456789012` genannten Access Point abgerufen. Bevor Sie dieses Beispiel ausführen, ersetzen Sie den Namen und die Kontonummer des Zugriffspunkts durch die entsprechenden Werte für Ihren Anwendungsfall.

```
aws s3control get-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap
```

Ausgabe:

```
{
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\", \"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/Admin\"}, \"Action\": \"s3:GetObject\", \"Resource\":\"arn:aws:s3:us-west-2:123456789012:accesspoint/finance-ap/object/records/*\"}]}"
}
```

Weitere Informationen finden Sie unter [Verwaltung des Datenzugriffs mit Amazon S3 Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [GetAccessPointPolicy](#) in der AWS CLI Befehlsreferenz.

get-access-point

Das folgende Codebeispiel zeigt die Verwendung `get-access-point`.

AWS CLI

Um die Konfigurationsdetails des Access Points abzurufen

Im folgenden `get-access-point` Beispiel werden die Konfigurationsdetails für den Access Point abgerufen, der `finance-ap` im Konto `123456789012` benannt ist. Bevor Sie dieses Beispiel ausführen, ersetzen Sie den Namen und die Kontonummer des Access Points durch die entsprechenden Werte für Ihren Anwendungsfall.

```
aws s3control get-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

Ausgabe:

```
{  
  "Name": "finance-ap",  
  "Bucket": "business-records",  
  "NetworkOrigin": "Internet",  
  "PublicAccessBlockConfiguration": {  
    "BlockPublicAcls": false,  
    "IgnorePublicAcls": false,  
    "BlockPublicPolicy": false,  
    "RestrictPublicBuckets": false  
  },  
  "CreationDate": "2020-01-01T00:00:00Z"  
}
```

Weitere Informationen finden Sie unter [Verwaltung des Datenzugriffs mit Amazon S3 Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [GetAccessPoint](#) in der AWS CLI Befehlsreferenz.

get-multi-region-access-point-routes

Das folgende Codebeispiel zeigt die Verwendung `get-multi-region-access-point-routes`.

AWS CLI

Um die aktuelle Routenkonfiguration für Access Points mit mehreren Regionen abzufragen

Das folgende `get-multi-region-access-point-routes` Beispiel gibt die aktuelle Routingkonfiguration für den angegebenen Multiregion Access Point zurück.

```
aws s3control get-multi-region-access-point-routes \  
  --region Region \  
  --account-id 111122223333 \  
  --mrap MultiRegionAccessPoint_ARN
```

Ausgabe:

```
{  
  "Mrap": "arn:aws:s3::111122223333:accesspoint/0000000000000000.mrap",  
  "Routes": [  
    {  
      "Bucket": "DOC-EXAMPLE-BUCKET-1",  
      "Region": "ap-southeast-2",  
      "TrafficDialPercentage": 100  
    },  
    {  
      "Bucket": "DOC-EXAMPLE-BUCKET-2",  
      "Region": "us-west-1",  
      "TrafficDialPercentage": 0  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [GetMultiRegionAccessPointRoutes AWS CLI Befehlsreferenz](#).

get-public-access-block

Das folgende Codebeispiel zeigt die Verwendung `get-public-access-block`.

AWS CLI

Um öffentliche Blockzugriffseinstellungen für ein Konto aufzulisten

Im folgenden `get-public-access-block` Beispiel werden die Einstellungen zum Sperren des öffentlichen Zugriffs für das angegebene Konto angezeigt.

```
aws s3control get-public-access-block \  
  --account-id 123456789012
```

Ausgabe:

```
{  
  "PublicAccessBlockConfiguration": {  
    "BlockPublicPolicy": true,  
    "RestrictPublicBuckets": true,  
    "IgnorePublicAcls": true,  
    "BlockPublicAcls": true  
  }  
}
```

- API-Einheiten finden Sie [GetPublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

list-access-points

Das folgende Codebeispiel zeigt die Verwendung `list-access-points`.

AWS CLI

Beispiel 1: Um eine Liste aller Access Points für ein Konto abzurufen

Im folgenden `list-access-points` Beispiel wird eine Liste aller Access Points angezeigt, die an Buckets angehängt sind, die dem Konto 123456789012 gehören.

```
aws s3control list-access-points \  
  --account-id 123456789012
```

Ausgabe:

```
{  
  "AccessPointList": [  
    {  
      "Name": "finance-ap",  
      "NetworkOrigin": "Internet",  
      "Bucket": "business-records"  
    },  
    {
```

```

        "Name": "managers-ap",
        "NetworkOrigin": "Internet",
        "Bucket": "business-records"
    },
    {
        "Name": "private-network-ap",
        "NetworkOrigin": "VPC",
        "VpcConfiguration": {
            "VpcId": "1a2b3c"
        },
        "Bucket": "business-records"
    },
    {
        "Name": "customer-ap",
        "NetworkOrigin": "Internet",
        "Bucket": "external-docs"
    },
    {
        "Name": "public-ap",
        "NetworkOrigin": "Internet",
        "Bucket": "external-docs"
    }
]
}

```

Beispiel 2: Um eine Liste aller Access Points für einen Bucket abzurufen

Im folgenden `list-access-points` Beispiel wird eine Liste aller Access Points abgerufen, die an den Bucket angehängt sind, der dem Konto 123456789012 `external-docs` gehört.

```

aws s3control list-access-points \
  --account-id 123456789012 \
  --bucket external-docs

```

Ausgabe:

```

{
  "AccessPointList": [
    {
      "Name": "customer-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    }
  ]
}

```

```
    },
    {
      "Name": "public-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwaltung des Datenzugriffs mit Amazon S3 Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [ListAccessPoints](#) in der AWS CLI Befehlsreferenz.

list-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-jobs`.

AWS CLI

Um ein Konto aufzulisten, Amazon S3 Batch Operations Jobs

Das folgende `list-jobs` Beispiel listet alle aktuellen Batch-Operationsaufträge für das angegebene Konto auf.

```
aws s3control list-jobs \
  --account-id 123456789012
```

Ausgabe:

```
{
  "Jobs": [
    {
      "Operation": "S3PutObjectTagging",
      "ProgressSummary": {
        "NumberOfTasksFailed": 0,
        "NumberOfTasksSucceeded": 8,
        "TotalNumberOfTasks": 8
      },
      "CreationTime": "2019-10-03T21:48:48.048Z",
      "Status": "Complete",
    }
  ]
}
```

```

    "JobId": "93735294-df46-44d5-8638-6356f335324e",
    "Priority": 42
  },
  {
    "Operation": "S3PutObjectTagging",
    "ProgressSummary": {
      "NumberOfTasksFailed": 0,
      "NumberOfTasksSucceeded": 0,
      "TotalNumberOfTasks": 0
    },
    "CreationTime": "2019-10-03T21:46:07.084Z",
    "Status": "Failed",
    "JobId": "3f3c7619-02d3-4779-97f6-1d98dd313108",
    "Priority": 42
  },
]
}

```

- API-Einheiten finden Sie [ListJobs](#) in der AWS CLI Befehlsreferenz.

put-access-point-policy

Das folgende Codebeispiel zeigt die Verwendung `put-access-point-policy`.

AWS CLI

Um eine Zugriffspunktrichtlinie festzulegen

Im folgenden `put-access-point-policy` Beispiel wird die angegebene Zugriffspunktrichtlinie für den Zugriffspunkt dem Konto `finance-ap 123456789012` zugewiesen. Wenn der Access Point `finance-ap` bereits über eine Richtlinie verfügt, ersetzt dieser Befehl die vorhandene Richtlinie durch die in diesem Befehl angegebene Richtlinie. Bevor Sie dieses Beispiel ausführen, ersetzen Sie die Kontonummer, den Namen des Zugriffspunkts und die Richtlinienanweisungen durch entsprechende Werte für Ihren Anwendungsfall.

```

aws s3control put-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap \
  --policy file://ap-policy.json

```

Inhalt von `ap-policy.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Alice"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/finance-ap/
object/Alice/*"
    }
  ]
}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung des Datenzugriffs mit Amazon S3 Access Points](#) im Amazon Simple Storage Service Developer Guide.

- API-Einheiten finden Sie [PutAccessPointPolicy](#) in der AWS CLI Befehlsreferenz.

put-public-access-block

Das folgende Codebeispiel zeigt die Verwendung `put-public-access-block`.

AWS CLI

So bearbeiten Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs für ein Konto

Im folgenden `put-public-access-block` Beispiel werden alle Einstellungen zum Sperren des öffentlichen Zugriffs `true` für das angegebene Konto auf „Sperren des öffentlichen Zugriffs“ umgeschaltet.

```
aws s3control put-public-access-block \
  --account-id 123456789012 \
  --public-access-block-configuration '{"BlockPublicAcls": true,
  "IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
  true}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutPublicAccessBlock](#) in der AWS CLI Befehlsreferenz.

submit-multi-region-access-point-routes

Das folgende Codebeispiel zeigt die Verwendung `submit-multi-region-access-point-routes`.

AWS CLI

So aktualisieren Sie die Routing-Konfiguration Ihres Access Points für mehrere Regionen

Im folgenden `submit-multi-region-access-point-routes` Beispiel werden die Routing-Status von `DOC-EXAMPLE-BUCKET-1` und `DOC-EXAMPLE-BUCKET-2` in der `ap-southeast-2` Region für Ihren Multi-Region Access Point aktualisiert.

```
aws s3control submit-multi-region-access-point-routes \  
  --region ap-southeast-2 \  
  --account-id 111122223333 \  
  --mrp MultiRegionAccessPoint_ARN \  
  --route-updates Bucket=DOC-EXAMPLE-  
BUCKET-1,TrafficDialPercentage=100 Bucket=DOC-EXAMPLE-  
BUCKET-2,TrafficDialPercentage=0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [SubmitMultiRegionAccessPointRoutes AWS CLI](#) Befehlsreferenz.

update-job-priority

Das folgende Codebeispiel zeigt die Verwendung `update-job-priority`.

AWS CLI

So aktualisieren Sie die Auftragspriorität eines Amazon S3 S3-Auftrags für Batch-Operationen

Im folgenden `update-job-priority` Beispiel wird der angegebene Job auf eine neue Priorität aktualisiert.

```
aws s3control update-job-priority \  
  --job-id JobID \  
  --priority Priority
```

```
--account-id 123456789012 \  
--job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
--priority 52
```

Ausgabe:

```
{  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",  
  "Priority": 52  
}
```

- API-Einheiten finden Sie [UpdateJobPriority](#) unter AWS CLI Befehlsreferenz.

update-job-status

Das folgende Codebeispiel zeigt die Verwendung `update-job-status`.

AWS CLI

So aktualisieren Sie den Status eines Amazon S3 S3-Auftrags für Batch-Operationen

Im folgenden `update-job-status` Beispiel wird der angegebene Job storniert, der auf seine Genehmigung wartet.

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
  --requested-job-status Cancelled
```

Ausgabe:

```
{  
  "Status": "Cancelled",  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"  
}
```

Im folgenden `update-job-status` Beispiel wird der angegebene, auf Genehmigung wartende Vorgang bestätigt und ausgeführt.

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
  --requested-job-status Completed
```

```
--account-id 123456789012 \  
--job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \  
--requested-job-status Ready
```

Output::

```
{  
  "Status": "Ready",  
  "JobId": "5782949f-3301-4fb3-  
be34-8d5bab54dbca"  
}
```

Im folgenden `update-job-status` Beispiel wird der angegebene Job, der gerade ausgeführt wird, abgebrochen.

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \  
  --requested-job-status Cancelled  
  
Output::  
{  
  "Status": "Cancelling",  
  "JobId": "5782949f-3301-4fb3-be34-8d5bab54dbca"  
}
```

- API-Einheiten finden Sie unter [UpdateJobStatus AWS CLI Befehlsreferenz](#).

S3 Glacier-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von S3 Glacier Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

abort-multipart-upload

Das folgende Codebeispiel zeigt die Verwendung `abort-multipart-upload`.

AWS CLI

Der folgende Befehl löscht einen laufenden mehrteiligen Upload in einen Tresor mit dem Namen: `my-vault`

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
--upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ
```

Dieser Befehl erzeugt keine Ausgabe. Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben. Die Upload-ID wird vom Befehl `aws glacier initiate-multipart-upload` zurückgegeben und kann auch mithilfe von `aws glacier list-multipart-uploads` abgerufen werden.

Weitere Informationen zu mehrteiligen Uploads auf Amazon Glacier mithilfe von finden Sie unter [Verwenden von Amazon Glacier im AWS CLIBenutzerhandbuch](#). AWS CLI

- API-Einheiten finden Sie [AbortMultipartUpload](#) in der AWS CLI Befehlsreferenz.

abort-vault-lock

Das folgende Codebeispiel zeigt die Verwendung `abort-vault-lock`.

AWS CLI

Um einen laufenden Vorgang zum Sperren von Tresoren abubrechen

Im folgenden `abort-vault-lock` Beispiel wird eine Tresorsperrrichtlinie aus dem angegebenen Tresor gelöscht und der Sperrstatus der Tresorsperre auf „Entsperrt“ zurückgesetzt.

```
aws glacier abort-vault-lock \
```

```
--account-id - \  
--vault-name MyVaultName
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Abort Vault Lock \(DELETEDlock-policy\)](#) im Amazon Glacier API Developer Guide.

- API-Einheiten finden Sie unter Befehlsreferenz [AbortVaultLock](#).AWS CLI

add-tags-to-vault

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-vault`.

AWS CLI

Der folgende Befehl fügt zwei Tags zu einem Tresor mit dem Namen `my-vault` hinzu:

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --  
tags id=1234,date=july2015
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [AddTagsToVault](#) in der AWS CLI Befehlsreferenz.

complete-multipart-upload

Das folgende Codebeispiel zeigt die Verwendung `complete-multipart-upload`.

AWS CLI

Der folgende Befehl schließt den mehrteiligen Upload für ein 3-MiB-Archiv ab:

```
aws glacier complete-multipart-upload --archive-size 3145728 --  
checksum 9628195fcdcbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67  
--upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-  
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Die Upload-ID wird vom Befehl `aws glacier initiate-multipart-upload` zurückgegeben und kann auch mithilfe von `aws glacier list-multipart-uploads` abgerufen werden. Der Checksum-Parameter verwendet einen Hash des Archivs mit einer Baumstruktur von SHA -256 im Hexadezimalformat.

Weitere Informationen zu mehrteiligen Uploads auf Amazon Glacier mithilfe von AWS CLI, einschließlich Anweisungen zur Berechnung eines Tree-Hashs, finden Sie unter Using Amazon Glacier im AWS CLIBenutzerhandbuch.

- API-Einheiten finden Sie [CompleteMultipartUpload](#) in der AWS CLI Befehlsreferenz.

complete-vault-lock

Das folgende Codebeispiel zeigt die Verwendung `complete-vault-lock`.

AWS CLI

Um einen laufenden Vorgang zum Sperren von Tresoren abzuschließen

Das folgende `complete-vault-lock` Beispiel schließt den Vorgang der Sperrung für den angegebenen Tresor ab und setzt den Sperrstatus der Tresorsperre auf. Locked Sie erhalten den Wert für den `lock-id` Parameter, wenn Sie ihn ausführen `initiate-lock-process`.

```
aws glacier complete-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName \  
  --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Complete Vault Lock \(POSTlockId\)](#) im Amazon Glacier API Developer Guide.

- API-Einheiten finden Sie [CompleteVaultLock](#) in der AWS CLI Befehlsreferenz.

create-vault

Das folgende Codebeispiel zeigt die Verwendung `create-vault`.

AWS CLI

Der folgende Befehl erstellt einen neuen Tresor mit dem Namen `my-vault`:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [CreateVault](#) in der AWS CLI Befehlsreferenz.

delete-archive

Das folgende Codebeispiel zeigt die Verwendung `delete-archive`.

AWS CLI

So löschen Sie ein Archiv aus einem Tresor

Im folgenden Beispiel für `delete-archive` wird das angegebene Archiv aus `example_vault` entfernt.

```
aws glacier delete-archive \  
  --account-id 111122223333 \  
  --vault-name example_vault \  
  --archive-id Sc0u9ZP8yaWkmh-XGLIvAVprtLhaLCGnNwN15I5x9HqPIkX5mjc0DrId3Ln-Gi_k2HzmLIDZUz117KSdVMdMXLuFWi9PJUitxW073edQ43eTLMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteArchive](#) in der AWS CLI Befehlsreferenz.

delete-vault-access-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-vault-access-policy`.

AWS CLI

Um die Zugriffsrichtlinie eines Tresors zu entfernen

Im folgenden `delete-vault-access-policy` Beispiel wird die Zugriffsrichtlinie für den angegebenen Tresor entfernt.

```
aws glacier delete-vault-access-policy \  
  --vault-name example_vault
```

```
--account-id 111122223333 \  
--vault-name example_vault
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteVaultAccessPolicy](#) in der AWS CLI Befehlsreferenz.

delete-vault-notifications

Das folgende Codebeispiel zeigt die Verwendung `delete-vault-notifications`.

AWS CLI

Um die SNS Benachrichtigungen für einen Tresor zu entfernen

Im folgenden `delete-vault-notifications` Beispiel werden Benachrichtigungen entfernt, die von Amazon Simple Notification Service (Amazon SNS) für den angegebenen Tresor gesendet wurden.

```
aws glacier delete-vault-notifications \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteVaultNotifications](#) unter AWS CLI Befehlsreferenz.

delete-vault

Das folgende Codebeispiel zeigt die Verwendung `delete-vault`.

AWS CLI

Der folgende Befehl löscht einen Tresor mit dem Namen `my-vault`:

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

Dieser Befehl erzeugt keine Ausgabe. Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [DeleteVault](#) in der AWS CLI Befehlsreferenz.

describe-job

Das folgende Codebeispiel zeigt die Verwendung `describe-job`.

AWS CLI

Der folgende Befehl ruft Informationen über einen Auftrag zum Abrufen von Inventar in einem Tresor mit dem Namen `my-vault` ab:

```
aws glacier describe-job --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3R1oGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW
```

Ausgabe:

```
{
  "InventoryRetrievalParameters": {
    "Format": "JSON"
  },
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "Completed": false,
  "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3R1oGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
  "Action": "InventoryRetrieval",
  "CreationDate": "2015-07-17T20:23:41.616Z",
  "StatusCode": "InProgress"
}
```

Die Auftrags-ID finden Sie in der Ausgabe von `aws glacier initiate-job` und `aws glacier list-jobs`. Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [DescribeJob](#) in der AWS CLI Befehlsreferenz.

describe-vault

Das folgende Codebeispiel zeigt die Verwendung `describe-vault`.

AWS CLI

Der folgende Befehl ruft Daten über einen Tresor mit dem Namen `my-vault` ab:

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [DescribeVault](#) in der AWS CLI Befehlsreferenz.

get-data-retrieval-policy

Das folgende Codebeispiel zeigt die Verwendung `get-data-retrieval-policy`.

AWS CLI

Mit dem folgenden Befehl wird die Datenabrufrichtlinie für das verwendete Konto abgerufen:

```
aws glacier get-data-retrieval-policy --account-id -
```

Ausgabe:

```
{
  "Policy": {
    "Rules": [
      {
        "BytesPerHour": 10737418240,
        "Strategy": "BytesPerHour"
      }
    ]
  }
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [GetDataRetrievalPolicy](#) in der AWS CLI Befehlsreferenz.

get-job-output

Das folgende Codebeispiel zeigt die Verwendung `get-job-output`.

AWS CLI

Der folgende Befehl speichert die Ausgabe eines Tresor-Inventarauftrags in einer Datei im aktuellen Verzeichnis mit dem Namen `output.json`:

```
aws glacier get-job-output --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3R1oGduS7Eg-R047Yc6FxsDGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW output.json
```

Die `job-id` ist in der Ausgabe von `aws glacier list-jobs` verfügbar. Beachten Sie, dass der Name der Ausgabedatei ein Positionsargument ist, dem kein Optionsname vorangestellt ist. Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Ausgabe:

```
{
  "status": 200,
  "acceptRanges": "bytes",
  "contentType": "application/json"
}
```

`output.json`:

```
{"VaultARN":"arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault", "InventoryDate":"2015-04-07T00:26:18Z", "ArchiveList": [{"ArchiveId":"kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtFKmQrj0IRQLSGsNuDp-AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw", "ArchiveDescription":"multipart upload test", "CreationDate":"2015-04-06T22:24:34Z", "Size":3145728, "SHA256TreeHash":"9628195fcdbcbb
```

- API-Einheiten finden Sie [GetJobOutput](#) in der AWS CLI Befehlsreferenz.

get-vault-access-policy

Das folgende Codebeispiel zeigt die Verwendung `get-vault-access-policy`.

AWS CLI

Um die Zugriffsrichtlinie eines Tresors abzurufen

Im folgenden `get-vault-access-policy` Beispiel wird die Zugriffsrichtlinie für den angegebenen Tresor abgerufen.

```
aws glacier get-vault-access-policy \
  --account-id 111122223333 \
  --vault-name example_vault
```

Ausgabe:

```
{
  "policy": {
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"arn:aws:iam:444455556666:root\"}, \"Action\": \"glacier:ListJobs\", \"Resource\": \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}, {\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"arn:aws:iam:444455556666:root\"}, \"Action\": \"glacier:UploadArchive\", \"Resource\": \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}]}"
  }
}
```

- API-Einheiten finden Sie unter [GetVaultAccessPolicy AWS CLI Befehlsreferenz](#).

get-vault-lock

Das folgende Codebeispiel zeigt die Verwendung `get-vault-lock`.

AWS CLI

Um die Details eines Tresorschlosses abzurufen

Im folgenden `get-vault-lock` Beispiel wurden die Details zur Sperre für den angegebenen Tresor abgerufen.

```
aws glacier get-vault-lock \
  --account-id - \
  --vault-name MyVaultName
```

Ausgabe:

```
{
```

```

    "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\": \"Define-vault-lock\", \"Effect\": \"Deny\", \"Principal\": { \"AWS\": \"arn:aws:iam:999999999999:root\" }, \"Action\": \"glacier:DeleteArchive\", \"Resource\": \"arn:aws:glacier:us-west-2:999999999999:vaults/MyVaultName\", \"Condition\": { \"NumericLessThanEquals\": { \"glacier:ArchiveAgeinDays\": \"365\" } } } ] }\",
    \"State\": \"Locked\",
    \"CreationDate\": \"2019-07-29T22:25:28.640Z\"
  }

```

Weitere Informationen finden [Sie unter Get Vault Lock \(GETlock-policy\)](#) im Amazon Glacier API Developer Guide.

- API-Einheiten finden Sie unter [GetVaultLock AWS CLI Befehlsreferenz](#).

get-vault-notifications

Das folgende Codebeispiel zeigt die Verwendung `get-vault-notifications`.

AWS CLI

Der folgende Befehl ruft eine Beschreibung der Benachrichtigungskonfiguration für einen Tresor mit dem Namen `my-vault` ab:

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

Ausgabe:

```

{
  "vaultNotificationConfig": {
    "Events": [
      "InventoryRetrievalCompleted",
      "ArchiveRetrievalCompleted"
    ],
    "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
  }
}

```

Wenn keine Benachrichtigungen für den Tresor konfiguriert wurden, wird ein Fehler zurückgegeben. Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [GetVaultNotifications](#) in der AWS CLI Befehlsreferenz.

initiate-job

Das folgende Codebeispiel zeigt die Verwendung `initiate-job`.

AWS CLI

Der folgende Befehl initiiert einen Job zum Abrufen einer Bestandsaufnahme des Tresors `my-vault`:

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
'{"Type": "inventory-retrieval"}'
```

Ausgabe:

```
{
  "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
R047Yc6FxsdBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
  "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
R047Yc6FxsdBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW"
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Der folgende Befehl initiiert einen Job zum Abrufen eines Archivs aus dem Tresor: `my-vault`

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file:///job-archive-retrieval.json
```

`job-archive-retrieval.json` ist eine JSON-Datei im lokalen Ordner, die den Auftragstyp, die Archiv-ID und einige optionale Parameter angibt:

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkpSDtFkmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
```

```
"Description": "Retrieve archive on 2015-07-17",
"SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

Archive IDs sind in der Ausgabe von `aws glacier upload-archive` und verfügbar aus `aws glacier get-job-output`.

Ausgabe:

```
{
  "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
  "jobId": "l7IL5-EkXy205uLYaFdAY0iEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav"
}
```

Einzelheiten zum Format der Auftragsparameter finden Sie unter [Job initiieren](#) in der Amazon API Glacier-Referenz.

- API Einzelheiten finden Sie [InitiateJob](#) unter AWS CLI Befehlsreferenz.

initiate-multipart-upload

Das folgende Codebeispiel zeigt die Verwendung `initiate-multipart-upload`.

AWS CLI

Der folgende Befehl initiiert einen mehrteiligen Upload in einen Tresor `my-vault` mit einer Teilgröße von 1 MiB (1024 x 1024 Byte) pro Datei:

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-
name my-vault --archive-description "multipart upload test"
```

Der Parameter für die Archivbeschreibung ist optional. Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Dieser Befehl gibt bei Erfolg eine Upload-ID aus. Verwenden Sie die Upload-ID, wenn Sie jeden Teil Ihres Archivs mit `aws glacier upload-multipart-part` hochladen. Weitere

Informationen zu mehrteiligen Uploads auf Amazon Glacier mithilfe von finden Sie unter [Verwenden von Amazon Glacier im AWS CLIBenutzerhandbuch](#). AWS CLI

- APIEinzelheiten finden Sie [InitiateMultipartUpload](#) in der AWS CLI Befehlsreferenz.

initiate-vault-lock

Das folgende Codebeispiel zeigt die Verwendung `initiate-vault-lock`.

AWS CLI

Um den Vorgang zum Sperren des Tresors einzuleiten

Im folgenden `initiate-vault-lock` Beispiel wird eine Tresorsperrrichtlinie für den angegebenen Tresor installiert und der Sperrstatus der Tresorsperre auf festgelegt `InProgress`. Sie müssen den Vorgang abschließen, indem Sie `complete-vault-lock` innerhalb von 24 Stunden anrufen, um den Status der Tresorsperre auf zu setzen `Locked`.

```
aws glacier initiate-vault-lock \
  --account-id - \
  --vault-name MyVaultName \
  --policy file://vault_lock_policy.json
```

Inhalt von `vault_lock_policy.json`:

```
{"Policy": "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Sid\": \"Define-vault-lock\", \"Effect\": \"Deny\", \"Principal\": {\"AWS\": \"arn:aws:iam::999999999999:root\"}, \"Action\": \"glacier:DeleteArchive\", \"Resource\": \"arn:aws:glacier:us-west-2:999999999999:vaults/examplevault\", \"Condition\": {\"NumericLessThanEquals\": {\"glacier:ArchiveAgeinDays\": \"365\"}}}]\"}"}
```

Die Ausgabe ist die Tresorsperren-ID, mit der Sie den Tresorsperrvorgang abschließen können.

```
{
  "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"
}
```

Weitere Informationen finden Sie unter [Initiate Vault Lock \(POSTlock-policy\)](#) im Amazon Glacier API Developer Guide.

- APIEinzelheiten finden Sie unter [InitiateVaultLock AWS CLIBefehlsreferenz](#).

list-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-jobs`.

AWS CLI

Der folgende Befehl listet laufende und kürzlich abgeschlossene Aufträge für einen Tresor mit dem Namen `my-vault` auf:

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

Ausgabe:

```
{
  "JobList": [
    {
      "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
      "RetrievalByteRange": "0-3145727",
      "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
      "Completed": false,
      "SHA256TreeHash":
"9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
      "JobId": "17IL5-EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
      "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkpSDtFkmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
      "JobDescription": "Retrieve archive on 2015-07-17",
      "ArchiveSizeInBytes": 3145728,
      "Action": "ArchiveRetrieval",
      "ArchiveSHA256TreeHash":
"9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
      "CreationDate": "2015-07-17T21:16:13.840Z",
      "StatusCode": "InProgress"
    },
    {
      "InventoryRetrievalParameters": {
        "Format": "JSON"
      },
      "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
      "Completed": false,
      "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
R047Yc6FxsdBGgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
```

```
    "Action": "InventoryRetrieval",
    "CreationDate": "2015-07-17T20:23:41.616Z",
    "StatusCode": ""InProgress""
  }
]
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [ListJobs](#) in der AWS CLI Befehlsreferenz.

list-multipart-uploads

Das folgende Codebeispiel zeigt die Verwendung `list-multipart-uploads`.

AWS CLI

Der folgende Befehl zeigt alle laufenden mehrteiligen Uploads für einen Tresor mit dem Namen: `my-vault`

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Weitere Informationen zu mehrteiligen Uploads auf Amazon Glacier mithilfe von finden Sie unter [Verwenden von Amazon Glacier im AWS CLIBenutzerhandbuch](#). AWS CLI

- API-Einheiten finden Sie [ListMultipartUploads](#) in der AWS CLI Befehlsreferenz.

list-parts

Das folgende Codebeispiel zeigt die Verwendung `list-parts`.

AWS CLI

Der folgende Befehl listet die hochgeladenen Teile für einen mehrteiligen Upload in einen Tresor mit dem Namen `my-vault` auf:

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdxCs8ut-D"
```

Ausgabe:

```
{
  "MultipartUploadId": "SYZi7qnL-
YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdιCs8ut-
D",
  "Parts": [
    {
      "RangeInBytes": "0-1048575",
      "SHA256TreeHash":
"e1f2a7cd6e047350f69b9f8cfa60fa606fe2f02802097a9a026360a7edc1f553"
    },
    {
      "RangeInBytes": "1048576-2097151",
      "SHA256TreeHash":
"43cf3061fb95796aed99a11a6aa3cd8f839eed15e655ab0a597126210636aee6"
    }
  ],
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "CreationDate": "2015-07-18T00:05:23.830Z",
  "PartSizeInBytes": 1048576
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Weitere Informationen zu mehrteiligen Uploads auf Amazon Glacier mithilfe von finden Sie unter [Verwenden von Amazon Glacier im AWS CLIBenutzerhandbuch](#). AWS CLI

- APIEinzelheiten finden Sie [ListParts](#) in der AWS CLI Befehlsreferenz.

list-provisioned-capacity

Das folgende Codebeispiel zeigt die Verwendung `list-provisioned-capacity`.

AWS CLI

Um die bereitgestellten Kapazitätseinheiten abzurufen

Im folgenden `list-provisioned-capacity` Beispiel werden Details zu allen bereitgestellten Kapazitätseinheiten für das angegebene Konto abgerufen.

```
aws glacier list-provisioned-capacity \
```

```
--account-id 111122223333
```

Ausgabe:

```
{
  "ProvisionedCapacityList": [
    {
      "CapacityId": "HpASAUvfRFiVDb0jMfEIcr8K",
      "ExpirationDate": "2020-03-18T19:59:24.000Z",
      "StartDate": "2020-02-18T19:59:24.912Z"
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListProvisionedCapacity AWS CLI Befehlsreferenz](#).

list-tags-for-vault

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-vault`.

AWS CLI

Der folgende Befehl listet die Tags auf, die auf einen Tresor mit dem Namen `my-vault` angewendet wurden:

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

Ausgabe:

```
{
  "Tags": {
    "date": "july2015",
    "id": "1234"
  }
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [ListTagsForVault](#) in der AWS CLI Befehlsreferenz.

list-vaults

Das folgende Codebeispiel zeigt die Verwendung `list-vaults`.

AWS CLI

Der folgende Befehl listet die Tresore im Standardkonto und der Standardregion auf:

```
aws glacier list-vaults --account-id -
```

Ausgabe:

```
{
  "VaultList": [
    {
      "SizeInBytes": 3178496,
      "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
      "LastInventoryDate": "2015-04-07T00:26:19.028Z",
      "VaultName": "my-vault",
      "NumberOfArchives": 1,
      "CreationDate": "2015-04-06T21:23:45.708Z"
    }
  ]
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [ListVaults](#) in der AWS CLI Befehlsreferenz.

purchase-provisioned-capacity

Das folgende Codebeispiel zeigt die Verwendung `purchase-provisioned-capacity`.

AWS CLI

Um eine bereitgestellte Kapazitätseinheit zu erwerben

Im folgenden `purchase-provisioned-capacity` Beispiel wird eine bereitgestellte Kapazitätseinheit gekauft.

```
aws glacier purchase-provisioned-capacity \
```

```
--account-id 111122223333
```

Ausgabe:

```
{  
  "capacityId": "HpASAvfRFiVDb0jMfEIcr8K"  
}
```

- API-Einheiten finden Sie [PurchaseProvisionedCapacity](#) in der AWS CLI Befehlsreferenz.

remove-tags-from-vault

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-vault`.

AWS CLI

Der folgende Befehl entfernt ein Tag mit dem Schlüssel `date` aus einem Tresor mit dem Namen `my-vault`:

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-  
keys date
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [RemoveTagsFromVault](#) in der AWS CLI Befehlsreferenz.

set-data-retrieval-policy

Das folgende Codebeispiel zeigt die Verwendung `set-data-retrieval-policy`.

AWS CLI

Mit dem folgenden Befehl wird eine Datenabrufrichtlinie für das verwendete Konto konfiguriert:

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-  
policy.json
```

`data-retrieval-policy.json` ist eine JSON-Datei im aktuellen Ordner, die eine Datenabrufrichtlinie angibt:

```
{
  "Rules":[
    {
      "Strategy":"BytesPerHour",
      "BytesPerHour":10737418240
    }
  ]
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Mit dem folgenden Befehl wird die Richtlinie für den Datenabruf auf die FreeTier Verwendung von Inline festgelegt: JSON

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules": [{"Strategy":"FreeTier"}]}'
```

Einzelheiten zum Richtlinienformat finden Sie unter Richtlinie für den Datenabruf festlegen in der Amazon API Glacier-Referenz.

- API Einzelheiten finden Sie unter [SetDataRetrievalPolicy AWS CLI](#) Befehlsreferenz.

set-vault-access-policy

Das folgende Codebeispiel zeigt die Verwendung `set-vault-access-policy`.

AWS CLI

Um die Zugriffsrichtlinie für einen Tresor festzulegen

Im folgenden `set-vault-access-policy` Beispiel wird eine Berechtigungsrichtlinie an den angegebenen Tresor angehängt.

```
aws glacier set-vault-access-policy \
  --account-id 111122223333 \
  --vault-name example_vault
  --policy '{"Policy": "{\n\"Version\":\n\"2012-10-17\", \n\"Statement\": [\n{\n\"Effect\":\n\"Allow\", \n\"Principal\": {\n\"AWS\":\n\"arn:aws:iam::444455556666:root\n\"}, \n\"Action\":\n\"glacier:ListJobs\", \n\"Resource\":\n\"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\n\"}, {\n\"Effect\":\n\"Allow\", \n\"Principal\":
```

```
{\"AWS\": {\"arn:aws:iam::444455556666:root\"}, \"Action\": \"glacier:UploadArchive\", \"Resource\": \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"}}}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [SetVaultAccessPolicy AWS CLI](#) Befehlsreferenz.

set-vault-notifications

Das folgende Codebeispiel zeigt die Verwendung `set-vault-notifications`.

AWS CLI

Der folgende Befehl konfiguriert SNS Benachrichtigungen für einen Tresor mit dem Namen `my-vault`:

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-notification-config file://notificationconfig.json
```

`notificationconfig.json` ist eine JSON-Datei im aktuellen Ordner, die ein SNS-Thema und die zu veröffentlichen Ereignisse angibt:

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

- API-Einheiten finden Sie [SetVaultNotifications](#) in der AWS CLI Befehlsreferenz.

upload-archive

Das folgende Codebeispiel zeigt die Verwendung `upload-archive`.

AWS CLI

Der folgende Befehl lädt ein Archiv im aktuellen Ordner mit dem Namen `archive.zip` in einen Tresor mit dem Namen `my-vault` hoch:


```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

Ausgabe:

```
{
  "archiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-
ybtRDvc2VkJPSDtfKmqRj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
  "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
  "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-ybtRDvc2VkJPSDtfKmqRj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Um ein hochgeladenes Archiv abzurufen, initiieren Sie einen Abrufauftrag mit dem Befehl `aws glacier initiate-job`.

- API-Einheiten finden Sie [UploadArchive](#) in der AWS CLI Befehlsreferenz.

upload-multipart-part

Das folgende Codebeispiel zeigt die Verwendung `upload-multipart-part`.

AWS CLI

Der folgende Befehl lädt den ersten, 1 MiB (1024 x 1024 Byte) umfassenden Teil eines Archivs hoch:

```
aws glacier upload-multipart-part --body part1 --range 'bytes
0-1048575/*' --account-id - --vault-name my-vault --upload-
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
0ssZtLqyFu7sY1_LR7vgFuJV6NtcV5zpsJ
```

Amazon Glacier benötigt bei der Durchführung von Operationen ein Konto-ID-Argument, Sie können jedoch einen Bindestrich verwenden, um das verwendete Konto anzugeben.

Der Parameter „body“ nimmt einen Pfad zu einer Teildatei im lokalen Dateisystem an. Der Bereichsparameter verwendet einen HTTP Inhaltsbereich, der die Byte angibt, die der Teil im

fertigen Archiv belegt. Die Upload-ID wird vom Befehl `aws glacier initiate-multipart-upload` zurückgegeben und kann auch mithilfe von `aws glacier list-multipart-uploads` abgerufen werden.

Weitere Informationen zu mehrteiligen Uploads auf Amazon Glacier mithilfe von finden Sie unter [Verwenden von Amazon Glacier im AWS CLIBenutzerhandbuch](#). AWS CLI

- API-Einheiten finden Sie [UploadMultipartPart](#) in der AWS CLI Befehlsreferenz.

Secrets Manager Manager-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Secrets Manager Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-get-secret-value

Das folgende Codebeispiel zeigt die Verwendung `batch-get-secret-value`.

AWS CLI

Beispiel 1: Um den geheimen Wert für eine Gruppe von Geheimnissen abzurufen, die nach Namen aufgelistet sind

Im folgenden `batch-get-secret-value` Beispiel wird der geheime Wert Secrets für drei Geheimnisse abgerufen.

```
aws secretsmanager batch-get-secret-value \  
  --secret-id-list MySecret1 MySecret2 MySecret3
```

Ausgabe:

```
{
  "SecretValues": [
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
      "Name": "MySecret1",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
      "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1523477145.729"
    },
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
      "Name": "MySecret2",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEbbbbbb",
      "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\""}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1673477781.275"
    },
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
      "Name": "MySecret3",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEcccccc",
      "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\""}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1373477721.124"
    }
  ],
  "Errors": []
}
```

```
}
```

Weitere Informationen finden Sie unter [Abrufen einer Gruppe von Geheimnissen in einem Batch](#) im AWS Secrets Manager Manager-Benutzerhandbuch.

Beispiel 2: So rufen Sie den geheimen Wert für eine Gruppe von Geheimnissen ab, die durch einen Filter ausgewählt wurden

Im folgenden `batch-get-secret-value` Beispiel wird der geheime Wert in Ihrem Konto abgerufen, der `MySecret` im Namen enthalten ist. Bei dem Filtern nach Namen muss die Groß- und Kleinschreibung beachtet werden.

```
aws secretsmanager batch-get-secret-value \
  --filters Key="name",Values="MySecret"
```

Ausgabe:

```
{
  "SecretValues": [
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
      "Name": "MySecret1",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
      "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1523477145.729"
    },
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
      "Name": "MySecret2",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb",
      "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
      "VersionStages": [
        "AWSCURRENT"
      ]
    }
  ]
}
```

```

    ],
    "CreateDate": "1673477781.275"
  },
  {
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
    "Name": "MySecret3",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEcccc",
    "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\""}",
    "VersionStages": [
      "AWSCURRENT"
    ],
    "CreateDate": "1373477721.124"
  }
],
"Errors": []
}

```

Weitere Informationen finden Sie unter [Abrufen einer Gruppe von Geheimnissen in einem Batch](#) im AWS Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetSecretValue](#) in der AWS CLI Befehlsreferenz.

cancel-rotate-secret

Das folgende Codebeispiel zeigt die Verwendung `cancel-rotate-secret`.

AWS CLI

Um die automatische Rotation für ein Geheimnis auszuschalten

Im folgenden `cancel-rotate-secret` Beispiel wird die automatische Rotation für ein Geheimnis deaktiviert. Rufen Sie an, um die Rotation fortzusetzen `rotate-secret`.

```
aws secretsmanager cancel-rotate-secret \
  --secret-id MyTestSecret
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
```

```
"Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Rotation eines Secrets](#) im Secrets Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CancelRotateSecret](#) in der AWS CLI Befehlsreferenz.

create-secret

Das folgende Codebeispiel zeigt die Verwendung `create-secret`.

AWS CLI

Beispiel 1: Um ein Geheimnis aus Anmeldeinformationen in einer JSON-Datei zu erstellen

Das folgende `create-secret`-Beispiel erstellt ein Secret anhand von Anmeldeinformationen in einer Datei. Weitere Informationen finden Sie im AWS CLI-Benutzerhandbuch unter [Laden von AWS CLI Parametern aus einer Datei](#).

```
aws secretsmanager create-secret \
  --name MyTestSecret \
  --secret-string file://mycreds.json
```

Inhalt von `mycreds.json`:

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
}
```

```
"VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Create a Secret im Secrets Manager](#) Manager-Benutzerhandbuch.

Beispiel 2: So erstellen Sie ein Geheimnis

Das folgende `create-secret`-Beispiel erstellt ein Secret mit zwei Schlüssel-/Wert-Paaren. Wenn Sie Befehle in eine Befehls-Shell eingeben, besteht die Gefahr, dass auf den Befehlsverlauf zugegriffen wird oder Serviceprogramme Zugriff auf Ihre Befehlsparameter haben. Dies ist ein Problem, wenn der Befehl den Wert eines Geheimnisses enthält. Weitere Informationen finden Sie unter [Mindern der Risiken bei der Verwendung von Befehlszeilentools zum Speichern von Geheimnissen im Secrets Manager](#) Manager-Benutzerhandbuch.

```
aws secretsmanager create-secret \  
  --name MyTestSecret \  
  --description "My test secret created with the CLI." \  
  --secret-string '{"user":"diegor","password":"EXAMPLE-PASSWORD"}'
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Create a Secret im Secrets Manager](#) Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSecret](#) in der AWS CLI Befehlsreferenz.

delete-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-policy`.

AWS CLI

Um die ressourcenbasierte Richtlinie zu löschen, die einem Geheimnis zugeordnet ist

Im folgenden `delete-resource-policy`-Beispiel wird die an ein Secret angefügte ressourcenbasierte Richtlinie gelöscht.

```
aws secretsmanager delete-resource-policy \  
  --secret-id MyTestSecret
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Weitere Informationen finden Sie unter [Authentifizierung und Zugriffskontrolle](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteResourcePolicy](#) in der AWS CLI Befehlsreferenz.

delete-secret

Das folgende Codebeispiel zeigt die Verwendung `delete-secret`.

AWS CLI

Beispiel 1: Um ein Geheimnis zu löschen

Im folgenden `delete-secret`-Beispiel wird ein Secret gelöscht. Sie können das Geheimnis `restore-secret` bis zu dem Datum und der Uhrzeit im `DeletionDate` Antwortfeld wiederherstellen. Um ein Secret zu löschen, das in andere Regionen repliziert wird, entfernen Sie zuerst die zugehörigen Replikat mit `remove-regions-from-replication` und rufen Sie dann `delete-secret` auf.

```
aws secretsmanager delete-secret \  
  --secret-id MyTestSecret \  
  --recovery-window-in-days 7
```

Ausgabe:

```
{
```



```
"ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "DeletionDate": 1524085349.095
}
```

Weitere Informationen finden Sie unter [Löschen eines Geheimnisses im Secrets Manager](#)-Benutzerhandbuch.

Beispiel 2: Um ein Geheimnis sofort zu löschen

Das folgende `delete-secret`-Beispiel löscht ein Secret sofort und ohne ein Wiederherstellungsfenster. Sie können dieses Secret nicht wiederherstellen.

```
aws secretsmanager delete-secret \
  --secret-id MyTestSecret \
  --force-delete-without-recovery
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "DeletionDate": 1508750180.309
}
```

Weitere Informationen finden Sie unter [Löschen eines Geheimnisses im Secrets Manager](#)-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSecret](#) in der AWS CLI Befehlsreferenz.

describe-secret

Das folgende Codebeispiel zeigt die Verwendung `describe-secret`.

AWS CLI

Um die Details eines Geheimnisses abzurufen

Das folgende `describe-secret` Beispiel zeigt die Details eines Geheimnisses.

```
aws secretsmanager describe-secret \  
--secret-id MyTestSecret
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
Ca8JGt",  
  "Name": "MyTestSecret",  
  "Description": "My test secret",  
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-  
ba987EXAMPLE",  
  "RotationEnabled": true,  
  "RotationLambdaARN": "arn:aws:lambda:us-  
west-2:123456789012:function:MyTestRotationLambda",  
  "RotationRules": {  
    "AutomaticallyAfterDays": 2,  
    "Duration": "2h",  
    "ScheduleExpression": "cron(0 16 1,15 * ? *)"  
  },  
  "LastRotatedDate": 1525747253.72,  
  "LastChangedDate": 1523477145.729,  
  "LastAccessedDate": 1524572133.25,  
  "Tags": [  
    {  
      "Key": "SecondTag",  
      "Value": "AnotherValue"  
    },  
    {  
      "Key": "FirstTag",  
      "Value": "SomeValue"  
    }  
  ],  
  "VersionIdsToStages": {  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [  
      "AWSPREVIOUS"  
    ],  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [  
      "AWSCURRENT"  
    ],  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333": [  
      "AWSPENDING"  
    ]  
  }  
}
```

```
  },
  "CreateDate": 1521534252.66,
  "PrimaryRegion": "us-west-2",
  "ReplicationStatus": [
    {
      "Region": "eu-west-3",
      "KmsKeyId": "alias/aws/secretsmanager",
      "Status": "InSync",
      "StatusMessage": "Replication succeeded"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Secret im Secrets Manager](#)-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSecret](#) in der AWS CLI Befehlsreferenz.

get-random-password

Das folgende Codebeispiel zeigt die Verwendung von `get-random-password`.

AWS CLI

Um ein zufälliges Passwort zu generieren

Im folgenden `get-random-password` Beispiel wird ein zufälliges Passwort mit einer Länge von 20 Zeichen generiert, das mindestens einen Großbuchstaben, einen Kleinbuchstaben, eine Zahl und ein Satzzeichen enthält.

```
aws secretsmanager get-random-password \
  --require-each-included-type \
  --password-length 20
```

Ausgabe:

```
{
  "RandomPassword": "EXAMPLE-PASSWORD"
}
```

Weitere Informationen finden Sie unter [Create and manage Secrets im Secrets Manager](#)-Benutzerhandbuch.

- API-Einheiten finden Sie [GetRandomPassword](#) in der AWS CLI Befehlsreferenz.

get-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `get-resource-policy`.

AWS CLI

Um die ressourcenbasierte Richtlinie abzurufen, die einem Geheimnis zugeordnet ist

Im folgenden `get-resource-policy`-Beispiel wird die an ein Secret angefügte ressourcenbasierte Richtlinie abgerufen.

```
aws secretsmanager get-resource-policy \
  --secret-id MyTestSecret
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "ResourcePolicy": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n      \"Effect\":\n        \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::123456789012:root\"\n      },\n      \"Action\":\n        \"secretsmanager:GetSecretValue\",\n      \"Resource\": \"*\"\n    }\n  ]\n}"
```

Weitere Informationen finden Sie unter [Authentifizierung und Zugriffskontrolle](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetResourcePolicy](#) in der AWS CLI Befehlsreferenz.

get-secret-value

Das folgende Codebeispiel zeigt die Verwendung `get-secret-value`.

AWS CLI

Beispiel 1: Um den verschlüsselten Geheimwert eines Geheimnisses abzurufen

Im folgenden `get-secret-value`-Beispiel wird der aktuelle Secret-Wert abgerufen.

```
aws secretsmanager get-secret-value \  
  --secret-id MyTestSecret
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "SecretString": "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}",  
  "VersionStages": [  
    "AWSCURRENT"  
  ],  
  "CreateDate": 1523477145.713  
}
```

Weitere Informationen finden Sie unter [Retrieve a Secret im Secrets Manager](#) Benutzerhandbuch.

Beispiel 2: Um den vorherigen geheimen Wert abzurufen

Im folgenden `get-secret-value` Beispiel wird der vorherige geheime Wert abgerufen. :

```
aws secretsmanager get-secret-value \  
  --secret-id MyTestSecret  
  --version-stage AWSPREVIOUS
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "SecretString": "{\"user\":\"diegor\",\"password\":\"PREVIOUS-EXAMPLE-PASSWORD  
\"}",  
  "VersionStages": [  
    "AWSPREVIOUS"  
  ]  
}
```

```
  ],  
  "CreateDate": 1523477145.713  
}
```

Weitere Informationen finden Sie unter [Retrieve a Secret im Secrets Manager](#) Benutzerhandbuch.

- API-Einheiten finden Sie [GetSecretValue](#) in der AWS CLI Befehlsreferenz.

list-secret-version-ids

Das folgende Codebeispiel zeigt die Verwendung `list-secret-version-ids`.

AWS CLI

Um alle geheimen Versionen aufzulisten, die einem Geheimnis zugeordnet sind

Im folgenden `list-secret-version-ids` Beispiel wird eine Liste aller Versionen eines Secrets abgerufen.

```
aws secretsmanager list-secret-version-ids \  
  --secret-id MyTestSecret
```

Ausgabe:

```
{  
  "Versions": [  
    {  
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "VersionStages": [  
        "AWSPREVIOUS"  
      ],  
      "LastAccessedDate": 1523477145.713,  
      "CreateDate": 1523477145.713  
    },  
    {  
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "VersionStages": [  
        "AWSCURRENT"  
      ],  
      "LastAccessedDate": 1523477145.713,  
      "CreateDate": 1523486221.391  
    }  
  ]  
}
```

```
    },
    {
      "CreateDate": 1.51197446236E9,
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333;"
    }
  ],
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Version](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListSecretVersionIds](#) in der AWS CLI Befehlsreferenz.

list-secrets

Das folgende Codebeispiel zeigt die Verwendung `list-secrets`.

AWS CLI

Beispiel 1: Um die Geheimnisse in Ihrem Konto aufzulisten

Das folgende `list-secrets`-Beispiel erhält eine Liste der Secrets in Ihrem Konto.

```
aws secretsmanager list-secrets
```

Ausgabe:

```
{
  "SecretList": [
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
      "Name": "MyTestSecret",
      "LastChangedDate": 1523477145.729,
      "SecretVersionsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
          "AWSCURRENT"
        ]
      }
    },
    {
```

```

    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:AnotherSecret-d4e5f6",
    "Name": "AnotherSecret",
    "LastChangedDate": 1523482025.685,
    "SecretVersionsToStages": {
      "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
        "AWSCURRENT"
      ]
    }
  ]
}

```

Weitere Informationen [finden Sie unter Find a Secret im Secrets Manager](#) Benutzerhandbuch.

Beispiel 2: So filtern Sie die Liste der Geheimnisse in Ihrem Konto

Im folgenden `list-secrets` Beispiel wird eine Liste der Geheimnisse in Ihrem Konto abgerufen, die `Test` im Namen enthalten sind. Bei dem Filtern nach Namen muss die Groß- und Kleinschreibung beachtet werden.

```

aws secretsmanager list-secrets \
  --filter Key="name",Values="Test"

```

Ausgabe:

```

{
  "SecretList": [
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
      "Name": "MyTestSecret",
      "LastChangedDate": 1523477145.729,
      "SecretVersionsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
          "AWSCURRENT"
        ]
      }
    }
  ]
}

```


Weitere Informationen [finden Sie unter Find a Secret im Secrets Manager Manager-Benutzerhandbuch](#).

Beispiel 3: Um die Geheimnisse in Ihrem Konto aufzulisten, die von einem anderen Dienst verwaltet werden

Das folgende `list-secrets` Beispiel gibt die Geheimnisse in Ihrem Konto zurück, die von Amazon verwaltet werden RDS.

```
aws secretsmanager list-secrets \  
  --filter Key="owning-service",Values="rds"
```

Ausgabe:

```
{  
  "SecretList": [  
    {  
      "Name": "rds!cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Tags": [  
        {  
          "Value": "arn:aws:rds:us-  
west-2:123456789012:cluster:database-1",  
          "Key": "aws:rds:primaryDBClusterArn"  
        },  
        {  
          "Value": "rds",  
          "Key": "aws:secretsmanager:owningService"  
        }  
      ],  
      "RotationRules": {  
        "AutomaticallyAfterDays": 1  
      },  
      "LastChangedDate": 1673477781.275,  
      "LastRotatedDate": 1673477781.26,  
      "SecretVersionsToStages": {  
        "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa": [  
          "AWSPREVIOUS"  
        ],  
        "a1b2c3d4-5678-90ab-cdef-EXAMPLEEbbbbbb": [  
          "AWSCURRENT",  
          "AWSPENDING"  
        ]  
      }  
    },  
  ],  
}
```

```

        "OwningService": "rds",
        "RotationEnabled": true,
        "CreatedDate": 1673467300.7,
        "LastAccessedDate": 1673395200.0,
        "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:rds!
cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-a1b2c3",
        "Description": "Secret associated with primary RDS DB cluster:
arn:aws:rds:us-west-2:123456789012:cluster:database-1"
    }
]
}

```

Weitere Informationen finden Sie unter [Von anderen Diensten verwaltete Geheimnisse](#) im Secrets Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListSecrets](#) in der AWS CLI Befehlsreferenz.

put-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `put-resource-policy`.

AWS CLI

Um einem Geheimnis eine ressourcenbasierte Richtlinie hinzuzufügen

Im folgenden `put-resource-policy`-Beispiel wird einem Secret eine Berechtigungsrichtlinie hinzugefügt, wobei zunächst geprüft wird, ob die Richtlinie keinen umfassenden Zugriff auf das Secret gewährt. Die Richtlinie wird aus einer Datei gelesen. Weitere Informationen finden Sie im AWS CLIBenutzerhandbuch unter [Laden von AWS CLI Parametern aus einer Datei](#).

```

aws secretsmanager put-resource-policy \
  --secret-id MyTestSecret \
  --resource-policy file://mypolicy.json \
  --block-public-policy

```

Inhalt von `mypolicy.json`:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",

```

```

        "Principal": {
            "AWS": "arn:aws:iam::123456789012:role/MyRole"
        },
        "Action": "secretsmanager:GetSecretValue",
        "Resource": "*"
    }
]
}

```

Ausgabe:

```

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}

```

Weitere Informationen finden Sie unter [Anhängen einer Berechtigungsrichtlinie an ein Geheimnis im Secrets Manager](#)-Benutzerhandbuch.

- API-Einheiten finden Sie [PutResourcePolicy](#) in der AWS CLI Befehlsreferenz.

put-secret-value

Das folgende Codebeispiel zeigt die Verwendung `put-secret-value`.

AWS CLI

Beispiel 1: Um einen neuen geheimen Wert in einem Geheimnis zu speichern

Das folgende `put-secret-value` Beispiel erstellt eine neue Version eines Geheimnisses mit zwei Schlüssel-Wert-Paaren.

```

aws secretsmanager put-secret-value \
  --secret-id MyTestSecret \
  --secret-string '{"user":"diegor","password":"EXAMPLE-PASSWORD"}'

```

Ausgabe:

```

{
  "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",

```

```

    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "VersionStages": [
        "AWSCURRENT"
    ]
}

```

Weitere Informationen finden Sie unter [Ändern eines Geheimnisses im Secrets Manager](#) Benutzerhandbuch.

Beispiel 2: Um einen neuen geheimen Wert aus Anmeldeinformationen in einer JSON Datei zu speichern

Das folgende `put-secret-value`-Beispiel erstellt eine neue Version eines Secrets anhand von Anmeldeinformationen in einer Datei. Weitere Informationen finden Sie im AWS CLI Benutzerhandbuch unter [Laden von AWS CLI Parametern aus einer Datei](#).

```

aws secretsmanager put-secret-value \
  --secret-id MyTestSecret \
  --secret-string file://mycreds.json

```

Inhalt von `mycreds.json`:

```

{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}

```

Ausgabe:

```

{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "VersionStages": [
    "AWSCURRENT"
  ]
}

```

```
]
}
```

Weitere Informationen finden Sie unter [Ändern eines Geheimnisses im Secrets Manager](#) Benutzerhandbuch.

- API-Einheiten finden Sie [PutSecretValue](#) in der AWS CLI Befehlsreferenz.

remove-regions-from-replication

Das folgende Codebeispiel zeigt die Verwendung `remove-regions-from-replication`.

AWS CLI

Um ein geheimes Replikat zu löschen

Im folgenden `remove-regions-from-replication`-Beispiel wird ein Replikat-Secret in `eu-west-3` gelöscht. Um ein primäres Secret zu löschen, das in andere Regionen repliziert wird, entfernen Sie zuerst die Replikate und rufen Sie dann `delete-secret` auf.

```
aws secretsmanager remove-regions-from-replication \
  --secret-id MyTestSecret \
  --remove-replica-regions eu-west-3
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
  "ReplicationStatus": []
}
```

Weitere Informationen finden Sie unter [Löschen eines geheimen Replikats im Secrets Manager](#) Benutzerhandbuch.

- API-Einheiten finden Sie [RemoveRegionsFromReplication](#) in der AWS CLI Befehlsreferenz.

replicate-secret-to-regions

Das folgende Codebeispiel zeigt die Verwendung `replicate-secret-to-regions`.

AWS CLI

Um ein Geheimnis in eine andere Region zu replizieren

Im folgenden `replicate-secret-to-regions`-Beispiel wird ein Secret in eu-west-3 repliziert. Das Replikat ist mit dem AWS verwalteten Schlüssel verschlüsselt. `aws/secretsmanager`

```
aws secretsmanager replicate-secret-to-regions \  
  --secret-id MyTestSecret \  
  --add-replica-regions Region=eu-west-3
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "ReplicationStatus": [  
    {  
      "Region": "eu-west-3",  
      "KmsKeyId": "alias/aws/secretsmanager",  
      "Status": "InProgress"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Ein Geheimnis in eine andere Region replizieren](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ReplicateSecretToRegions](#) in der AWS CLI Befehlsreferenz.

restore-secret

Das folgende Codebeispiel zeigt die Verwendung `restore-secret`.

AWS CLI

Um ein zuvor gelöscht Geheimnis wiederherzustellen

Das folgende `restore-secret`-Beispiel stellt ein Secret wieder her, dessen Löschung zuvor geplant war.

```
aws secretsmanager restore-secret \  
  --secret-id MyTestSecret
```

```
--secret-id MyTestSecret
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Löschen eines Geheimnisses im Secrets Manager](#)-Benutzerhandbuch.

- API-Einheiten finden Sie [RestoreSecret](#) in der AWS CLI Befehlsreferenz.

rotate-secret

Das folgende Codebeispiel zeigt die Verwendung `rotate-secret`.

AWS CLI

Beispiel 1: Um die automatische Rotation für ein Geheimnis zu konfigurieren und zu starten

Im folgenden `rotate-secret` Beispiel wird die automatische Rotation für ein Geheimnis konfiguriert und gestartet. Secrets Manager wechselt das Geheimnis einmal sofort und dann alle acht Stunden in einem zweistündigen Fenster. Die Ausgabe zeigt die `VersionId` neue geheime Version, die durch Rotation erstellt wurde.

```
aws secretsmanager rotate-secret \
  --secret-id MyTestDatabaseSecret \
  --rotation-lambda-arn arn:aws:lambda:us-
west-2:1234566789012:function:SecretsManagerTestRotationLambda \
  --rotation-rules '{"ScheduleExpression": "cron(0 8/8 * * ? *)"', "Duration
": "2h"}'
```

Ausgabe:

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
```

```
"VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Rotation von Geheimnissen](#) im Secrets Manager Manager-Benutzerhandbuch.

Beispiel 2: Um die automatische Rotation in einem Rotationsintervall zu konfigurieren und zu starten

Im folgenden `rotate-secret` Beispiel wird die automatische Rotation für ein Geheimnis konfiguriert und gestartet. Secrets Manager wechselt das Secret einmal sofort und dann alle 10 Tage. Die Ausgabe zeigt die `VersionId` neue geheime Version, die durch Rotation erstellt wurde.

```
aws secretsmanager rotate-secret \  
  --secret-id MyTestDatabaseSecret \  
  --rotation-lambda-arn arn:aws:lambda:us-west-2:1234566789012:function:SecretsManagerTestRotationLambda \  
  --rotation-rules '{"ScheduleExpression": "rate(10 days)\"}'
```

Ausgabe:

```
{  
  "ARN": "aws:arn:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",  
  "Name": "MyTestDatabaseSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Rotation von Geheimnissen](#) im Secrets Manager Manager-Benutzerhandbuch.

Beispiel 3: Um ein Geheimnis sofort zu rotieren

Im folgenden `rotate-secret`-Beispiel wird eine sofortige Rotation gestartet. Die Ausgabe zeigt die `VersionId` der neuen geheimen Version, die durch Rotation erstellt wurde. Für das Secret muss die Rotation bereits konfiguriert sein.

```
aws secretsmanager rotate-secret \  
  --secret-id MyTestDatabaseSecret
```


Ausgabe:

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Weitere Informationen finden Sie unter [Rotation von Geheimnissen](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [RotateSecret](#) in der AWS CLI Befehlsreferenz.

stop-replication-to-replica

Das folgende Codebeispiel zeigt die Verwendung `stop-replication-to-replica`.

AWS CLI

Um ein geheimes Replikat zu einem Primärgeheimnis heraufzustufen

Im folgenden `stop-replication-to-replica`-Beispiel wird die Verknüpfung zwischen einem Replikat-Secret und dem primären entfernt. Das Replikat-Secret wird in der Replikat-Region zum primären Secret heraufgestuft. Sie müssen `stop-replication-to-replica` innerhalb der Replikatregion aufrufen.

```
aws secretsmanager stop-replication-to-replica \
  --secret-id MyTestSecret
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3"
}
```

Weitere Informationen finden Sie unter [Promote a Replica Secret im Secrets Manager](#)-Benutzerhandbuch.

- API-Einheiten finden Sie [StopReplicationToReplica](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Beispiel 1: Um einem Geheimnis ein Tag hinzuzufügen

Im folgenden -Beispiel wird gezeigt, wie Sie ein Tag mit Abkürzungssyntax anfügen.

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags Key=FirstTag,Value=FirstValue
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre Geheimnisse im Secrets Manager](#) Manager-Benutzerhandbuch.

Beispiel 2: So fügen Sie einem Geheimnis mehrere Tags hinzu

Das folgende `tag-resource`-Beispiel fügt zwei Schlüssel-/Wert-Tags an ein Secret an.

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",  
  "Value": "SecondValue"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kennzeichnen von Geheimnissen](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einem Geheimnis zu entfernen

Im folgenden `untag-resource`-Beispiel werden zwei Tags aus einem Secret entfernt. Für jedes Tag werden sowohl der Schlüssel als auch der Wert entfernt.

```
aws secretsmanager untag-resource \  
  --secret-id MyTestSecret \  
  --tag-keys '[ "FirstTag", "SecondTag" ]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Kennzeichen von Geheimnissen](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-secret-version-stage

Das folgende Codebeispiel zeigt die Verwendung `update-secret-version-stage`.

AWS CLI

Beispiel 1: Um ein Geheimnis auf die vorherige Version zurückzusetzen

Im folgenden `update-secret-version-stage` Beispiel wird das AWS CURRENT Staging-Label auf die vorherige Version eines Secrets verschoben, wodurch das Secret auf die vorherige Version zurückgesetzt wird. Um die ID für die vorherige Version zu finden, verwenden Sie `list-secret-version-ids`. In diesem Beispiel ist die Version mit der AWS CURRENT Bezeichnung `a1b2c3d4-5678-90ab-cdef-` und die Version mit der Bezeichnung `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`. In diesem Beispiel verschieben Sie AWS CURRENT das Label von Version 11111 auf 22222. Da das AWS CURRENT Label aus einer Version entfernt wurde, wird das AWS PREVIOUS Label `update-secret-version-stage` automatisch in diese Version (11111) verschoben. Dies hat zur Folge, dass die AWS PREVIOUS Versionen AWS CURRENT und die Versionen vertauscht werden.

```
aws secretsmanager update-secret-version-stage \  
  --secret-id MyTestSecret \  
  --version-stage AWSCURRENT \  
  --move-to-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \  
  --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Version](#) im Secrets Manager Manager-Benutzerhandbuch.

Beispiel 2: So fügen Sie ein Staging-Label hinzu, das an eine Version eines Secrets angehängt ist

Im folgenden `update-secret-version-stage` Beispiel wird einer Version eines Secrets ein Staging-Label hinzugefügt. Sie können die Ergebnisse überprüfen, indem Sie das `VersionStages` Antwortfeld für die betroffene Version ausführen `list-secret-version-ids` und anzeigen.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage STAGINGLABEL1 \
  --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Version](#) im Secrets Manager Manager-Benutzerhandbuch.

Beispiel 3: Um ein Staging-Label zu löschen, das an eine Version eines Secrets angehängt ist

Im folgenden `update-secret-version-stage` Beispiel wird ein Staging-Label gelöscht, das an eine Version eines Secrets angehängt ist. Sie können die Ergebnisse überprüfen, indem Sie das `VersionStages` Antwortfeld für die betroffene Version ausführen `list-secret-version-ids` und anzeigen.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage STAGINGLABEL1 \
```

```
--remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Version](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSecretVersionStage](#) in der AWS CLI Befehlsreferenz.

update-secret

Das folgende Codebeispiel zeigt die Verwendung `update-secret`.

AWS CLI

Beispiel 1: Um die Beschreibung eines Geheimnisses zu aktualisieren

Im folgenden `update-secret`-Beispiel wird die Beschreibung eines Secrets aktualisiert.

```
aws secretsmanager update-secret \
  --secret-id MyTestSecret \
  --description "This is a new description for the secret."
```

Ausgabe:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Weitere Informationen finden Sie unter [Ändern eines Geheimnisses im Secrets Manager](#) Benutzerhandbuch.

Beispiel 2: So aktualisieren Sie den Verschlüsselungsschlüssel, der einem Geheimnis zugeordnet ist

Im folgenden `update-secret` Beispiel wird der KMS Schlüssel aktualisiert, mit dem der geheime Wert verschlüsselt wurde. Der KMS Schlüssel muss sich in derselben Region wie der geheime Schlüssel befinden.

```
aws secretsmanager update-secret \  
  --secret-id MyTestSecret \  
  --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

Ausgabe:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Weitere Informationen finden Sie unter [Ändern eines Geheimnisses im Secrets Manager](#) Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSecret](#) in der AWS CLI Befehlsreferenz.

validate-resource-policy

Das folgende Codebeispiel zeigt die Verwendung `validate-resource-policy`.

AWS CLI

Um eine Ressourcenrichtlinie zu validieren

Im folgenden `validate-resource-policy` Beispiel wird überprüft, ob eine Ressourcenrichtlinie keinen umfassenden Zugriff auf ein Geheimnis gewährt. Die Richtlinie wird aus einer Datei auf der Festplatte gelesen. Weitere Informationen finden Sie im AWS CLIBenutzerhandbuch unter [Laden von AWS CLI Parametern aus einer Datei](#).

```
aws secretsmanager validate-resource-policy \  
  --resource-policy file://mypolicy.json
```

Inhalt von `mypolicy.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/MyRole"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

Ausgabe:

```
{
  "PolicyValidationPassed": true,
  "ValidationErrors": []
}
```

Weitere Informationen finden Sie unter [Berechtigungsreferenz für Secrets Manager](#) im Secrets Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ValidateResourcePolicy](#) unter AWS CLI Befehlsreferenz.

Security Hub Hub-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Security Hub Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-administrator-invitation

Das folgende Codebeispiel zeigt die Verwendung `accept-administrator-invitation`.

AWS CLI

Um eine Einladung von einem Administratorkonto anzunehmen

Im folgenden `accept-administrator-invitation` Beispiel wird die angegebene Einladung vom angegebenen Administratorkonto akzeptiert.

```
aws securityhub accept-invitation \  
  --administrator-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [AcceptAdministratorInvitation](#) in der AWS CLI Befehlsreferenz.

accept-invitation

Das folgende Codebeispiel zeigt die Verwendung `accept-invitation`.

AWS CLI

Um eine Einladung von einem Administratorkonto anzunehmen

Im folgenden `accept-invitation` Beispiel wird die angegebene Einladung vom angegebenen Administratorkonto akzeptiert.

```
aws securityhub accept-invitation \  
  --master-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [AcceptInvitation](#) in der AWS CLI Befehlsreferenz.

batch-delete-automation-rules

Das folgende Codebeispiel zeigt die Verwendung `batch-delete-automation-rules`.

AWS CLI

Um Automatisierungsregeln zu löschen

Im folgenden `batch-delete-automation-rules` Beispiel wird die angegebene Automatisierungsregel gelöscht. Sie können eine oder mehrere Regeln mit einem einzigen Befehl löschen. Nur das Security Hub-Administratorkonto kann diesen Befehl ausführen.

```
aws securityhub batch-delete-automation-rules \  
  --automation-rules-arns '["arn:aws:securityhub:us-  
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

Ausgabe:

```
{  
  "ProcessedAutomationRules": [  
    "arn:aws:securityhub:us-east-1:123456789012:automation-rule/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  ],  
  "UnprocessedAutomationRules": []  
}
```

Weitere Informationen finden Sie unter [Löschen von Automatisierungsregeln](#) im AWS Security Hub Benutzerhandbuch.

- API-Einheiten finden Sie [BatchDeleteAutomationRules](#) in der AWS CLI Befehlsreferenz.

batch-disable-standards

Das folgende Codebeispiel zeigt die Verwendung `batch-disable-standards`.

AWS CLI

Um einen Standard zu deaktivieren

Im folgenden `batch-disable-standards` Beispiel wird der Standard deaktiviert, der dem angegebenen Abonnement ARN zugeordnet ist.

```
aws securityhub batch-disable-standards \  
  --standards-subscription-arns "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

Ausgabe:

```
{  
  "StandardsSubscriptions": [  
    {  
      "StandardsArn": "arn:aws:securityhub:eu-central-1::standards/pci-dss/  
v/3.2.1",  
      "StandardsInput": { },  
      "StandardsStatus": "DELETING",  
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Deaktivieren oder Aktivieren eines Sicherheitsstandards im AWS Security Hub](#) Hub Benutzerhandbuch.

- API-Einheiten finden Sie [BatchDisableStandards](#) in der AWS CLI Befehlsreferenz.

batch-enable-standards

Das folgende Codebeispiel zeigt die Verwendung `batch-enable-standards`.

AWS CLI

Um einen Standard zu aktivieren

Das folgende `batch-enable-standards` Beispiel aktiviert den PCI DSS Standard für das anfordernde Konto.

```
aws securityhub batch-enable-standards \  
  --standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-  
west-1::standards/pci-dss/v/3.2.1"}'
```

Ausgabe:

```
{
  "StandardsSubscriptions": [
    {
      "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/v/3.2.1",
      "StandardsInput": { },
      "StandardsStatus": "PENDING",
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/v/3.2.1"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Deaktivieren oder Aktivieren eines Sicherheitsstandards im AWS Security Hub](#) Benutzerhandbuch.

- API-Einheiten finden Sie [BatchEnableStandards](#) in der AWS CLI Befehlsreferenz.

batch-get-automation-rules

Das folgende Codebeispiel zeigt die Verwendung `batch-get-automation-rules`.

AWS CLI

Um Details zu Automatisierungsregeln abzurufen

Im folgenden `batch-get-automation-rules` Beispiel werden Details für die angegebene Automatisierungsregel abgerufen. Sie können Details für eine oder mehrere Automatisierungsregeln mit einem einzigen Befehl abrufen.

```
aws securityhub batch-get-automation-rules \
  --automation-rules-arns '["arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

Ausgabe:

```
{
  "Rules": [
    {
```

```
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "RuleStatus": "ENABLED",
    "RuleOrder": 1,
    "RuleName": "Suppress informational findings",
    "Description": "Suppress GuardDuty findings with Informational
severity",
    "IsTerminal": false,
    "Criteria": {
      "ProductName": [
        {
          "Value": "GuardDuty",
          "Comparison": "EQUALS"
        }
      ],
      "SeverityLabel": [
        {
          "Value": "INFORMATIONAL",
          "Comparison": "EQUALS"
        }
      ],
      "WorkflowStatus": [
        {
          "Value": "NEW",
          "Comparison": "EQUALS"
        }
      ],
      "RecordState": [
        {
          "Value": "ACTIVE",
          "Comparison": "EQUALS"
        }
      ]
    },
    "Actions": [
      {
        "Type": "FINDING_FIELDS_UPDATE",
        "FindingFieldsUpdate": {
          "Note": {
            "Text": "Automatically suppress GuardDuty findings with
Informational severity",
            "UpdatedBy": "sechub-automation"
          },
          "Workflow": {
```

```

        "Status": "SUPPRESSED"
      }
    }
  ],
  "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
  "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
  "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
}
],
"UnprocessedAutomationRules": []
}

```

Weitere Informationen finden Sie unter [Automatisierungsregeln anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetAutomationRules](#) in der AWS CLI Befehlsreferenz.

batch-get-configuration-policy-associations

Das folgende Codebeispiel zeigt die Verwendung `batch-get-configuration-policy-associations`.

AWS CLI

Um Details zur Konfigurationszuweisung für einen Stapel von Zielen abzurufen

Im folgenden `batch-get-configuration-policy-associations` Beispiel werden Zuordnungsdetails für die angegebenen Ziele abgerufen. Sie können das KontoIDs, die Organisationseinheit IDs oder die Stamm-ID für das Ziel angeben.

```

aws securityhub batch-get-configuration-policy-associations \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'

```

Ausgabe:

```

{
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "TargetId": "ou-6hi7-8j91kl2m",
  "TargetType": "ORGANIZATIONAL_UNIT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
}

```

```

    "AssociationStatus": "SUCCESS",
    "AssociationStatusMessage": "Association applied successfully on this target."
  }

```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetConfigurationPolicyAssociations](#) in der AWS CLI Befehlsreferenz.

batch-get-security-controls

Das folgende Codebeispiel zeigt die Verwendung `batch-get-security-controls`.

AWS CLI

Um Details zur Sicherheitskontrolle abzurufen

Im folgenden `batch-get-security-controls` Beispiel werden Details zu den Sicherheitskontrollen ACM .1 und IAM .1 im AWS Girokonto und in der AWS Region abgerufen.

```

aws securityhub batch-get-security-controls \
  --security-control-ids '["ACM.1", "IAM.1"]'

```

Ausgabe:

```

{
  "SecurityControls": [
    {
      "SecurityControlId": "ACM.1",
      "SecurityControlArn": "arn:aws:securityhub:us-east-2:123456789012:security-control/ACM.1",
      "Title": "Imported and ACM-issued certificates should be renewed after a specified time period",
      "Description": "This control checks whether an AWS Certificate Manager (ACM) certificate is renewed within the specified time period. It checks both imported certificates and certificates provided by ACM. The control fails if the certificate isn't renewed within the specified time period. Unless you provide a custom parameter value for the renewal period, Security Hub uses a default value of 30 days.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/ACM.1/remediation",
      "SeverityRating": "MEDIUM",
    }
  ]
}

```

```

    "SecurityControlStatus": "ENABLED"
    "UpdateStatus": "READY",
    "Parameters": {
      "daysToExpiration": {
        "ValueType": CUSTOM,
        "Value": {
          "Integer": 15
        }
      }
    },
    "LastUpdateReason": "Updated control parameter"
  },
  {
    "SecurityControlId": "IAM.1",
    "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/IAM.1",
    "Title": "IAM policies should not allow full \"*\
privileges",
    "Description": "This AWS control checks whether the default version of
AWS Identity and Access Management (IAM) policies (also known as customer managed
policies) do not have administrator access with a statement that has \"Effect\
\": \"Allow\" with \"Action\": \"*\
over \"Resource\": \"*\
". It only checks for
the Customer Managed Policies that you created, but not inline and AWS Managed
Policies.",
    "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
IAM.1/remediation",
    "SeverityRating": "HIGH",
    "SecurityControlStatus": "ENABLED"
    "UpdateStatus": "READY",
    "Parameters": {}
  }
]
}

```

Weitere Informationen finden Sie unter [Details für ein Steuerelement anzeigen](#) im AWS Security Hub Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetSecurityControls](#) in der AWS CLI Befehlsreferenz.

batch-get-standards-control-associations

Das folgende Codebeispiel zeigt die Verwendung `batch-get-standards-control-associations`.

AWS CLI

Um den Aktivierungsstatus eines Steuerelements abzurufen

Das folgende `batch-get-standards-control-associations` Beispiel identifiziert, ob die angegebenen Steuerelemente in den angegebenen Standards aktiviert sind.

```
aws securityhub batch-get-standards-control-associations \
  --standards-control-association-ids '["SecurityControlId":
  "Config.1", "StandardsArn": "arn:aws:securityhub:us-east-1:123456789012:ruleset/cis-
  aws-foundations-benchmark/v/1.2.0"}, {"SecurityControlId": "IAM.6", "StandardsArn":
  "arn:aws:securityhub:us-east-1:123456789012:standards/aws-foundational-security-
  best-practices/v/1.0.0"}]'
```

Ausgabe:

```
{
  "StandardsControlAssociationDetails": [
    {
      "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
      benchmark/v/1.2.0",
      "SecurityControlId": "Config.1",
      "SecurityControlArn": "arn:aws:securityhub:us-
      east-1:068873283051:security-control/Config.1",
      "AssociationStatus": "ENABLED",
      "RelatedRequirements": [
        "CIS AWS Foundations 2.5"
      ],
      "UpdatedAt": "2022-10-27T16:07:12.960000+00:00",
      "StandardsControlTitle": "Ensure AWS Config is enabled",
      "StandardsControlDescription": "AWS Config is a web service that
      performs configuration management of supported AWS resources within your account
      and delivers log files to you. The recorded information includes the configuration
      item (AWS resource), relationships between configuration items (AWS resources), and
      any configuration changes between resources. It is recommended to enable AWS Config
      in all regions.",
      "StandardsControlArns": [
        "arn:aws:securityhub:us-east-1:068873283051:control/cis-aws-
        foundations-benchmark/v/1.2.0/2.5"
      ]
    },
    {
```



```

    "StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-
foundational-security-best-practices/v/1.0.0",
    "SecurityControlId": "IAM.6",
    "SecurityControlArn": "arn:aws:securityhub:us-
east-1:068873283051:security-control/IAM.6",
    "AssociationStatus": "DISABLED",
    "RelatedRequirements": [],
    "UpdatedAt": "2022-11-22T21:30:35.080000+00:00",
    "UpdatedReason": "test",
    "StandardsControlTitle": "Hardware MFA should be enabled for the root
user",
    "StandardsControlDescription": "This AWS control checks whether your AWS
account is enabled to use a hardware multi-factor authentication (MFA) device to
sign in with root user credentials.",
    "StandardsControlArns": [
        "arn:aws:securityhub:us-east-1:068873283051:control/aws-
foundational-security-best-practices/v/1.0.0/IAM.6"
    ]
}
]
}
}

```

Weitere Informationen finden Sie unter [Aktivieren und Deaktivieren von Steuerungen in bestimmten Standards](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchGetStandardsControlAssociations](#) in der AWS CLI Befehlsreferenz.

batch-import-findings

Das folgende Codebeispiel zeigt die Verwendung `batch-import-findings`.

AWS CLI

Um einen Befund zu aktualisieren

Im folgenden `batch-import-findings` Beispiel wird ein Befund aktualisiert.

```

aws securityhub batch-import-findings \
  --findings '
  [{
    "AwsAccountId": "123456789012",

```

```

    "CreatedAt": "2020-05-27T17:05:54.832Z",
    "Description": "Vulnerability in a CloudTrail trail",
    "FindingProviderFields": {
      "Severity": {
        "Label": "LOW",
        "Original": "10"
      },
      "Types": [
        "Software and Configuration Checks/Vulnerabilities/CVE"
      ]
    },
    "GeneratorId": "TestGeneratorId",
    "Id": "Id1",
    "ProductArn": "arn:aws:securityhub:us-
west-1:123456789012:product/123456789012/default",
    "Resources": [
      {
        "Id": "arn:aws:cloudtrail:us-west-1:123456789012:trail/
TrailName",
        "Partition": "aws",
        "Region": "us-west-1",
        "Type": "AwsCloudTrailTrail"
      }
    ],
    "SchemaVersion": "2018-10-08",
    "Title": "CloudTrail trail vulnerability",
    "UpdatedAt": "2020-06-02T16:05:54.832Z"
  }]

```

Ausgabe:

```

{
  "FailedCount": 0,
  "SuccessCount": 1,
  "FailedFindings": []
}

```

Weitere Informationen finden Sie unter [Verwendung BatchImportFindings zur Erstellung und Aktualisierung von Ergebnissen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchImportFindings](#) in der AWS CLI Befehlsreferenz.

batch-update-automation-rules

Das folgende Codebeispiel zeigt die Verwendung `batch-update-automation-rules`.

AWS CLI

Um Automatisierungsregeln zu aktualisieren

Im folgenden `batch-update-automation-rules` Beispiel wird die angegebene Automatisierungsregel aktualisiert. Sie können eine oder mehrere Regeln mit einem einzigen Befehl aktualisieren. Nur das Security Hub-Administratorkonto kann diesen Befehl ausführen.

```
aws securityhub batch-update-automation-rules \
  --update-automation-rules-request-items '[ \
    { \
      "Actions": [{ \
        "Type": "FINDING_FIELDS_UPDATE", \
        "FindingFieldsUpdate": { \
          "Note": { \
            "Text": "Known issue that is a risk", \
            "UpdatedBy": "sechub-automation" \
          }, \
          "Workflow": { \
            "Status": "NEW" \
          } \
        } \
      }], \
      "Criteria": { \
        "SeverityLabel": [{ \
          "Value": "LOW", \
          "Comparison": "EQUALS" \
        }] \
      }, \
      "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", \
      "RuleOrder": 1, \
      "RuleStatus": "DISABLED" \
    } \
  ]'
```

Ausgabe:

```
{
```

```

    "ProcessedAutomationRules": [
      "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
  }

```

Weitere Informationen finden Sie unter [Automatisierungsregeln bearbeiten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchUpdateAutomationRules](#) in der AWS CLI Befehlsreferenz.

batch-update-findings

Das folgende Codebeispiel zeigt die Verwendung `batch-update-findings`.

AWS CLI

Beispiel 1: Um einen Befund zu aktualisieren

Im folgenden `batch-update-findings` Beispiel werden zwei Ergebnisse aktualisiert, um eine Notiz hinzuzufügen, die Bezeichnung für den Schweregrad zu ändern und das Problem zu beheben.

```

aws securityhub batch-update-findings \
  --finding-identifiers '[{"Id": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", "ProductArn": "arn:aws:securityhub:us-
west-1::product/aws/securityhub"}, {"Id": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222", "ProductArn": "arn:aws:securityhub:us-
west-1::product/aws/securityhub"}]' \
  --note '{"Text": "Known issue that is not a risk.", "UpdatedBy": "user1"}' \
  --severity '{"Label": "LOW"}' \
  --workflow '{"Status": "RESOLVED"}'

```

Ausgabe:

```

{
  "ProcessedFindings": [
    {
      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",

```

```

        "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
    },
    {
        "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
    }
],
"UnprocessedFindings": []
}

```

Weitere Informationen finden Sie unter [Verwendung BatchUpdateFindings zur Aktualisierung eines Befundes](#) im AWS Security Hub Hub-Benutzerhandbuch.

Beispiel 2: Um ein Ergebnis mithilfe einer Kurzsyntax zu aktualisieren

Im folgenden batch-update-findings Beispiel werden zwei Ergebnisse aktualisiert, um eine Notiz hinzuzufügen, die Bezeichnung für den Schweregrad zu ändern und das Problem mithilfe einer Kurzsyntax aufzulösen.

```

aws securityhub batch-update-findings \
  --finding-identifiers Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" \
  --note Text="Known issue that is not a risk.",UpdatedBy="user1" \
  --severity Label="LOW" \
  --workflow Status="RESOLVED"

```

Ausgabe:

```

{
  "ProcessedFindings": [
    {
      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
    },
    {

```

```

      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
    }
  ],
  "UnprocessedFindings": []
}

```

Weitere Informationen finden Sie unter [Verwendung BatchUpdateFindings zur Aktualisierung eines Befundes](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [BatchUpdateFindings](#) in der AWS CLI Befehlsreferenz.

batch-update-standards-control-associations

Das folgende Codebeispiel zeigt die Verwendung `batch-update-standards-control-associations`.

AWS CLI

Um den Aktivierungsstatus eines Steuerelements in aktivierten Standards zu aktualisieren

Im folgenden `batch-update-standards-control-associations` Beispiel wird `CloudTrail .1` in den angegebenen Standards deaktiviert.

```

aws securityhub batch-update-standards-control-associations \
  --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",
"StandardsArn": "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable
to environment"}, {"SecurityControlId": "CloudTrail.1", "StandardsArn":
"arn:aws:securityhub::standards/cis-aws-foundations-benchmark/v/1.4.0",
"AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable to
environment"}]'

```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Weitere Informationen finden Sie unter [Steuerungen in bestimmten Standards aktivieren und deaktivieren und Kontrollen in allen Standards aktivieren und deaktivieren im AWS Security Hub Hub-Benutzerhandbuch](#).

- API-Einheiten finden Sie [BatchUpdateStandardsControlAssociations](#) in der AWS CLI Befehlsreferenz.

create-action-target

Das folgende Codebeispiel zeigt die Verwendung `create-action-target`.

AWS CLI

Um eine benutzerdefinierte Aktion zu erstellen

Im folgenden `create-action-target` Beispiel wird eine benutzerdefinierte Aktion erstellt. Es enthält den Namen, die Beschreibung und die Kennung für die Aktion.

```
aws securityhub create-action-target \  
  --name "Send to remediation" \  
  --description "Action to send the finding for remediation tracking" \  
  --id "Remediation"
```

Ausgabe:

```
{  
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation"  
}
```

Weitere Informationen finden Sie unter [Eine benutzerdefinierte Aktion erstellen und sie einer CloudWatch Ereignisregel zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateActionTarget](#) in der AWS CLI Befehlsreferenz.

create-automation-rule

Das folgende Codebeispiel zeigt die Verwendung `create-automation-rule`.

AWS CLI

Um eine Automatisierungsregel zu erstellen

Im folgenden `create-automation-rule` Beispiel wird eine Automatisierungsregel für das AWS Girokonto und die AWS Region erstellt. Security Hub filtert Ihre Ergebnisse anhand der angegebenen Kriterien und wendet die Aktionen auf übereinstimmende Ergebnisse an. Nur das Security Hub-Administratorkonto kann diesen Befehl ausführen.

```
aws securityhub create-automation-rule \  

```

```

--actions '[{ \
  "Type": "FINDING_FIELDS_UPDATE", \
  "FindingFieldsUpdate": { \
    "Severity": { \
      "Label": "HIGH" \
    }, \
    "Note": { \
      "Text": "Known issue that is a risk. Updated by automation rules", \
      "UpdatedBy": "sechub-automation" \
    } \
  } \
}]' \
--criteria '{ \
  "SeverityLabel": [{ \
    "Value": "INFORMATIONAL", \
    "Comparison": "EQUALS" \
  }] \
}' \
--description "A sample rule" \
--no-is-terminal \
--rule-name "sample rule" \
--rule-order 1 \
--rule-status "ENABLED"

```

Ausgabe:

```

{
  "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}

```

Weitere Informationen finden Sie unter [Automatisierungsregeln erstellen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateAutomationRule](#) unter AWS CLI Befehlsreferenz.

create-configuration-policy

Das folgende Codebeispiel zeigt die Verwendung `create-configuration-policy`.

AWS CLI

Um eine Konfigurationsrichtlinie zu erstellen

Im folgenden create-configuration-policy Beispiel wird eine Konfigurationsrichtlinie mit den angegebenen Einstellungen erstellt.

```
aws securityhub create-configuration-policy \
  --name "SampleConfigurationPolicy" \
  --description "SampleDescription" \
  --configuration-policy '{"SecurityHub": {"ServiceEnabled":
true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-
central-1::standards/aws-foundational-security-best-practices/
v/1.0.0", "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"], "SecurityControlsConfiguration": {"DisabledSecurityControlIdentifiers":
["CloudTrail.2"], "SecurityControlCustomParameters": [{"SecurityControlId":
"ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value":
{"Integer": 15}}}}}}}' \
  --tags '{"Environment": "Prod"}'
```

Ausgabe:

```
{
  "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Name": "SampleConfigurationPolicy",
  "Description": "SampleDescription",
  "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
  "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
  "ConfigurationPolicy": {
    "SecurityHub": {
      "ServiceEnabled": true,
      "EnabledStandardIdentifiers": [
        "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
        "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
      ],
      "SecurityControlsConfiguration": {
        "DisabledSecurityControlIdentifiers": [
          "CloudTrail.2"
        ],
        "SecurityControlCustomParameters": [
          {
            "SecurityControlId": "ACM.1",
            "Parameters": {
```

```

    "daysToExpiration": {
      "ValueType": "CUSTOM",
      "Value": {
        "Integer": 15
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien erstellen und zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateConfigurationPolicy AWS CLI Befehlsreferenz](#).

create-finding-aggregator

Das folgende Codebeispiel zeigt die Verwendung `create-finding-aggregator`.

AWS CLI

Um die Suchaggregation zu aktivieren

Im folgenden `create-finding-aggregator` Beispiel wird die Suchaggregation konfiguriert. Es wird von US East (Virginia) aus ausgeführt, was US East (Virginia) als Aggregationsregion ausweist. Es gibt an, nur bestimmte Regionen zu verknüpfen und neue Regionen nicht automatisch zu verknüpfen. Es wählt USA West (Nordkalifornien) und USA West (Oregon) als verknüpfte Regionen aus.

```

aws securityhub create-finding-aggregator \
  --region us-east-1 \
  --region-linking-mode SPECIFIED_REGIONS \
  --regions us-west-1,us-west-2

```

Ausgabe:

```
{
```

```

    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
    aggregator/123e4567-e89b-12d3-a456-426652340000",
    "FindingAggregationRegion": "us-east-1",
    "RegionLinkingMode": "SPECIFIED_REGIONS",
    "Regions": "us-west-1,us-west-2"
  }

```

Weitere Informationen finden Sie unter [Enabling Finding Aggregation](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateFindingAggregator](#) in der AWS CLI Befehlsreferenz.

create-insight

Das folgende Codebeispiel zeigt die Verwendung `create-insight`.

AWS CLI

Um einen benutzerdefinierten Einblick zu erstellen

Im folgenden `create-insight` Beispiel wird ein benutzerdefinierter Einblick mit dem Namen `Critical role findings` erstellt, der kritische Ergebnisse zurückgibt, die sich auf AWS Rollen beziehen.

```

aws securityhub create-insight \
  --filters '{"ResourceType": [{"Comparison": "EQUALS", "Value": "AwsIamRole"}],
  "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \
  --group-by-attribute "ResourceId" \
  --name "Critical role findings"

```

Ausgabe:

```

{
  "InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}

```

Weitere Informationen finden Sie unter [Managing Custom Insights](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateInsight](#) in der AWS CLI Befehlsreferenz.

create-members

Das folgende Codebeispiel zeigt die Verwendung `create-members`.

AWS CLI

Um Konten als Mitgliedskonten hinzuzufügen

Im folgenden `create-members` Beispiel werden dem anfragenden Administratorkonto zwei Konten als Mitgliedskonten hinzugefügt.

```
aws securityhub create-members \  
  --account-details '[{"AccountId": "123456789111"}, {"AccountId":  
  "123456789222"}]'
```

Ausgabe:

```
{  
  "UnprocessedAccounts": []  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateMembers](#) in der AWS CLI Befehlsreferenz.

decline-invitations

Das folgende Codebeispiel zeigt die Verwendung `decline-invitations`.

AWS CLI

Um eine Einladung als Mitgliedskonto abzulehnen

Im folgenden `decline-invitations` Beispiel wird eine Einladung, Mitglied des angegebenen Administratorkontos zu werden, abgelehnt. Das Mitgliedskonto ist das anfragende Konto.

```
aws securityhub decline-invitations \  
  --account-ids "123456789012"
```

Ausgabe:

```
{
  "UnprocessedAccounts": []
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeclineInvitations](#) in der AWS CLI Befehlsreferenz.

delete-action-target

Das folgende Codebeispiel zeigt die Verwendung `delete-action-target`.

AWS CLI

Um eine benutzerdefinierte Aktion zu löschen

Im folgenden `delete-action-target` Beispiel wird die benutzerdefinierte Aktion gelöscht, die durch die angegebene ARN Aktion identifiziert wurde.

```
aws securityhub delete-action-target \
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
  Remediation"
```

Ausgabe:

```
{
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
  Remediation"
}
```

Weitere Informationen finden Sie unter [Eine benutzerdefinierte Aktion erstellen und sie einer CloudWatch Ereignisregel zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteActionTarget](#) in der AWS CLI Befehlsreferenz.

delete-configuration-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-configuration-policy`.

AWS CLI

Um eine Konfigurationsrichtlinie zu löschen

Im folgenden `delete-configuration-policy` Beispiel wird die angegebene Konfigurationsrichtlinie gelöscht.

```
aws securityhub delete-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien löschen und deren Zuordnung aufheben](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteConfigurationPolicy](#) in der AWS CLI Befehlsreferenz.

`delete-finding-aggregator`

Das folgende Codebeispiel zeigt die Verwendung `delete-finding-aggregator`.

AWS CLI

Um die Suche nach Aggregation zu beenden

Im folgenden `delete-finding-aggregator` Beispiel wird die Suche nach einer Aggregation beendet. Es wird von US East (Virginia) aus ausgeführt, der Aggregationsregion.

```
aws securityhub delete-finding-aggregator \  
  --region us-east-1 \  
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-  
aggregator/123e4567-e89b-12d3-a456-426652340000
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Stoppen der Suche nach Aggregation](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteFindingAggregator](#) in der AWS CLI Befehlsreferenz.

delete-insight

Das folgende Codebeispiel zeigt die Verwendung `delete-insight`.

AWS CLI

Um einen benutzerdefinierten Einblick zu löschen

Im folgenden `delete-insight` Beispiel wird der benutzerdefinierte Einblick mit dem angegebenen ARN Wert gelöscht.

```
aws securityhub delete-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Ausgabe:

```
{  
  "InsightArn": "arn:aws:securityhub:eu-  
central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111"  
}
```

Weitere Informationen finden Sie unter [Managing Custom Insights](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteInsight](#) in der AWS CLI Befehlsreferenz.

delete-invitations

Das folgende Codebeispiel zeigt die Verwendung `delete-invitations`.

AWS CLI

Um eine Einladung zu einem Mitgliedskonto zu löschen

Im folgenden `delete-invitations` Beispiel wird eine Einladung gelöscht, ein Mitgliedskonto für das angegebene Administratorkonto zu werden. Das Mitgliedskonto ist das anfragende Konto.

```
aws securityhub delete-invitations \  
  --invitation-id "arn:aws:securityhub:eu-central-1:123456789012:invitation/123456789012/custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
--account-ids "123456789012"
```

Ausgabe:

```
{
  "UnprocessedAccounts": []
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteInvitations](#) in der AWS CLI Befehlsreferenz.

delete-members

Das folgende Codebeispiel zeigt die Verwendung `delete-members`.

AWS CLI

Um Mitgliedskonten zu löschen

Im folgenden `delete-members` Beispiel werden die angegebenen Mitgliedskonten aus dem anfragenden Administratorkonto gelöscht.

```
aws securityhub delete-members \
  --account-ids "123456789111" "123456789222"
```

Ausgabe:

```
{
  "UnprocessedAccounts": []
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteMembers](#) in der AWS CLI Befehlsreferenz.

describe-action-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-action-targets`.

AWS CLI

Um Details zu benutzerdefinierten Aktionen abzurufen

Im folgenden `describe-action-targets` Beispiel werden Informationen über die benutzerdefinierte Aktion abgerufen, die durch die angegebene ARN Aktion identifiziert wurde.

```
aws securityhub describe-action-targets \  
  --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
  Remediation"
```

Ausgabe:

```
{  
  "ActionTargets": [  
    {  
      "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/  
      custom/Remediation",  
      "Description": "Action to send the finding for remediation tracking",  
      "Name": "Send to remediation"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Eine benutzerdefinierte Aktion erstellen und sie einer CloudWatch Ereignisregel zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeActionTargets](#) in der AWS CLI Befehlsreferenz.

describe-hub

Das folgende Codebeispiel zeigt die Verwendung `describe-hub`.

AWS CLI

Um Informationen über eine Hub-Ressource abzurufen

Im folgenden `describe-hub` Beispiel wird das Abonnementdatum für die angegebene Hub-Ressource zurückgegeben. Die Hub-Ressource wird durch ihre identifiziertARN.

```
aws securityhub describe-hub \  
  --arn "arn:aws:securityhub:us-west-1:123456789012:hub/123456789012"
```

```
--hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Ausgabe:

```
{
  "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",
  "SubscribedAt": "2019-11-19T23:15:10.046Z"
}
```

Weitere Informationen finden Sie unter [AWS:SecurityHub: :Hub](#) im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeHub](#) in der AWS CLI Befehlsreferenz.

describe-organization-configuration

Das folgende Codebeispiel zeigt die Verwendung `describe-organization-configuration`.

AWS CLI

Um zu sehen, wie Security Hub für eine Organisation konfiguriert ist

Das folgende `describe-organization-configuration` Beispiel gibt Informationen darüber zurück, wie eine Organisation in Security Hub konfiguriert ist. In diesem Beispiel verwendet die Organisation die zentrale Konfiguration. Nur das Security Hub-Administratorkonto kann diesen Befehl ausführen.

```
aws securityhub describe-organization-configuration
```

Ausgabe:

```
{
  "AutoEnable": false,
  "MemberAccountLimitReached": false,
  "AutoEnableStandards": "NONE",
  "OrganizationConfiguration": {
    "ConfigurationType": "LOCAL",
    "Status": "ENABLED",
    "StatusMessage": "Central configuration has been enabled successfully"
  }
}
```

```
}
```

Weitere Informationen finden Sie unter [Konten bei AWS Organizations verwalten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeOrganizationConfiguration](#) in der AWS CLI Befehlsreferenz.

describe-products

Das folgende Codebeispiel zeigt die Verwendung `describe-products`.

AWS CLI

Um Informationen über verfügbare Produktintegrationen zurückzugeben

Im folgenden `describe-products` Beispiel werden die verfügbaren Produktintegrationen nacheinander zurückgegeben.

```
aws securityhub describe-products \  
  --max-results 1
```

Ausgabe:

```
{  
  "NextToken": "U2FsdGVkX18vvP10qb7RD1rWRWVFBJI46M0IAb+nZmRjMmR15NoRi2gm13sdQEn30/  
pq/78dGs+bKpgA+7HMPH00qX33/zoRI+uIG/F9yLNhc0r0WzFUdy36JcXLQji3Rpnn/  
cD1SVkGA98qI3zPOSDg==",  
  "Products": [  
    {  
      "ProductArn": "arn:aws:securityhub:us-west-1:123456789333:product/  
crowdstrike/crowdstrike-falcon",  
      "ProductName": "CrowdStrike Falcon",  
      "CompanyName": "CrowdStrike",  
      "Description": "CrowdStrike Falcon's single lightweight sensor unifies  
next-gen antivirus, endpoint detection and response, and 24/7 managed hunting, via  
the cloud.",  
      "Categories": [  
        "Endpoint Detection and Response (EDR)",  
        "AV Scanning and Sandboxing",  
        "Threat Intelligence Feeds and Reports",  
        "Endpoint Forensics",  
        "Network Forensics"  
      ]  
    }  
  ]  
}
```

```

    ],
    "IntegrationTypes": [
        "SEND_FINDINGS_TO_SECURITY_HUB"
    ],
    "MarketplaceUrl": "https://aws.amazon.com/marketplace/seller-profile?id=a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "ActivationUrl": "https://falcon.crowdstrike.com/support/documentation",
    "ProductSubscriptionResourcePolicy": "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"123456789333\"}, \"Action\": [\"securityhub:BatchImportFindings\"], \"Resource\": \"arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon\", \"Condition\": {\"StringEquals\": {\"securityhub:TargetAccount\": \"123456789012\"}}}, {\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"123456789012\"}, \"Action\": [\"securityhub:BatchImportFindings\"], \"Resource\": \"arn:aws:securityhub:us-west-1:123456789333:product/crowdstrike/crowdstrike-falcon\", \"Condition\": {\"StringEquals\": {\"securityhub:TargetAccount\": \"123456789012\"}}}]}"
  }
]
}

```

Weitere Informationen finden Sie unter [Verwaltung von Produktintegrationen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeProducts](#) in der AWS CLI Befehlsreferenz.

describe-standards-controls

Das folgende Codebeispiel zeigt die Verwendung `describe-standards-controls`.

AWS CLI

Um die Liste der Steuerelemente in einem aktivierten Standard anzufordern

Im folgenden `describe-standards-controls` Beispiel wird die Liste der Steuerelemente angefordert, die im Abonnement des PCI DSS Standardkontos des anfordernden Kontos enthalten sind. Die Anforderung gibt zwei Steuerelemente gleichzeitig zurück.

```

aws securityhub describe-standards-controls \
  --standards-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/v/3.2.1" \
  --max-results 2

```

Ausgabe:

```
{
  "Controls": [
    {
      "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1",
      "ControlStatus": "ENABLED",
      "ControlStatusUpdatedAt": "2020-05-15T18:49:04.473000+00:00",
      "ControlId": "PCI.AutoScaling.1",
      "Title": "Auto scaling groups associated with a load balancer should use
health checks",
      "Description": "This AWS control checks whether your Auto Scaling groups
that are associated with a load balancer are using Elastic Load Balancing health
checks.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.AutoScaling.1/remediation",
      "SeverityRating": "LOW",
      "RelatedRequirements": [
        "PCI DSS 2.2"
      ]
    },
    {
      "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.CW.1",
      "ControlStatus": "ENABLED",
      "ControlStatusUpdatedAt": "2020-05-15T18:49:04.498000+00:00",
      "ControlId": "PCI.CW.1",
      "Title": "A log metric filter and alarm should exist for usage of the
\"root\" user",
      "Description": "This control checks for the CloudWatch metric
filters using the following pattern { $.userIdentity.type = \"Root\" &&
$.userIdentity.invokedBy NOT EXISTS && $.eventType != \"AwsServiceEvent\" }
It checks that the log group name is configured for use with active multi-
region CloudTrail, that there is at least one Event Selector for a Trail with
IncludeManagementEvents set to true and ReadWriteType set to All, and that there is
at least one active subscriber to an SNS topic associated with the alarm.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.CW.1/remediation",
      "SeverityRating": "MEDIUM",
      "RelatedRequirements": [
        "PCI DSS 7.2.1"
      ]
    }
  ]
}
```

```

    ],
    "NextToken": "U2FsdGVkX1+eNkPoZHVl111ip5HUYQPWSWZGmftcmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveo0dvfxXCk0BaGhohIXhsIedN+LSjQV/
17kfCfJcq4PziNC1N9xe9aq2pjllLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUvoeriHZhyIiBgWKf7oNYdBVG80EortVWvSkoUTt
+B2ThcnC7l43kI0UNx1kZ6sc64AsW"
}

```

Weitere Informationen finden Sie im AWS Security Hub Hub-Benutzerhandbuch unter [Details zu Steuerelementen anzeigen](#).

- API-Einheiten finden Sie [DescribeStandardsControls](#) in der AWS CLI Befehlsreferenz.

describe-standards

Das folgende Codebeispiel zeigt die Verwendung `describe-standards`.

AWS CLI

Um eine Liste verfügbarer Standards zurückzugeben

Das folgende `describe-standards` Beispiel gibt die Liste der verfügbaren Standards zurück.

```
aws securityhub describe-standards
```

Ausgabe:

```

{
  "Standards": [
    {
      "StandardsArn": "arn:aws:securityhub:us-west-1::standards/aws-
foundational-security-best-practices/v/1.0.0",
      "Name": "AWS Foundational Security Best Practices v1.0.0",
      "Description": "The AWS Foundational Security Best Practices standard
is a set of automated security checks that detect when AWS accounts and deployed
resources do not align to security best practices. The standard is defined by AWS
security experts. This curated set of controls helps improve your security posture
in AWS, and cover AWS's most popular and foundational services.",
      "EnabledByDefault": true
    },
    {
      "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",

```

```

        "Name": "CIS AWS Foundations Benchmark v1.2.0",
        "Description": "The Center for Internet Security (CIS) AWS Foundations
Benchmark v1.2.0 is a set of security configuration best practices for AWS. This
Security Hub standard automatically checks for your compliance readiness against a
subset of CIS requirements.",
        "EnabledByDefault": true
    },
    {
        "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
        "Name": "PCI DSS v3.2.1",
        "Description": "The Payment Card Industry Data Security Standard (PCI
DSS) v3.2.1 is an information security standard for entities that store, process,
and/or transmit cardholder data. This Security Hub standard automatically checks
for your compliance readiness against a subset of PCI DSS requirements.",
        "EnabledByDefault": false
    }
]
}

```

Weitere Informationen finden Sie unter [Sicherheitsstandards in AWS Security Hub](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeStandards](#) in der AWS CLI Befehlsreferenz.

disable-import-findings-for-product

Das folgende Codebeispiel zeigt die Verwendung `disable-import-findings-for-product`.

AWS CLI

Um keine Ergebnisse mehr aus einer Produktintegration zu erhalten

Im folgenden `disable-import-findings-for-product` Beispiel wird der Ergebnisfluss für das angegebene Abonnement einer Produktintegration deaktiviert.

```

aws securityhub disable-import-findings-for-product \
  --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon"

```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Produktintegrationen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DisableImportFindingsForProductin](#) der AWS CLI Befehlsreferenz.

disable-organization-admin-account

Das folgende Codebeispiel zeigt die Verwendung `disable-organization-admin-account`.

AWS CLI

So entfernen Sie ein Security Hub-Administratorkonto

Im folgenden `disable-organization-admin-account` Beispiel wird die Zuweisung des angegebenen Kontos als Security Hub-Administratorkonto für AWS Organizations aufgehoben.

```
aws securityhub disable-organization-admin-account \  
  --admin-account-id 777788889999
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ein Security Hub-Administratorkonto einrichten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DisableOrganizationAdminAccountin](#) der AWS CLI Befehlsreferenz.

disable-security-hub

Das folgende Codebeispiel zeigt die Verwendung `disable-security-hub`.

AWS CLI

Um AWS Security Hub zu deaktivieren

Im folgenden `disable-security-hub` Beispiel wird AWS Security Hub für das anfordernde Konto deaktiviert.

```
aws securityhub disable-security-hub
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deaktivieren von AWS Security Hub](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DisableSecurityHub](#) in der AWS CLI Befehlsreferenz.

disassociate-from-administrator-account

Das folgende Codebeispiel zeigt die Verwendung `disassociate-from-administrator-account`.

AWS CLI

Um die Verbindung zu einem Administratorkonto zu trennen

Im folgenden `disassociate-from-administrator-account` Beispiel wird das anfragende Konto von seinem aktuellen Administratorkonto getrennt.

```
aws securityhub disassociate-from-administrator-account
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateFromAdministratorAccount](#) in der AWS CLI Befehlsreferenz.

disassociate-from-master-account

Das folgende Codebeispiel zeigt die Verwendung `disassociate-from-master-account`.

AWS CLI

Um die Verbindung zu einem Administratorkonto zu trennen

Im folgenden `disassociate-from-master-account` Beispiel wird das anfragende Konto von seinem aktuellen Administratorkonto getrennt.

```
aws securityhub disassociate-from-master-account
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateFromMasterAccount](#) in der AWS CLI Befehlsreferenz.

disassociate-members

Das folgende Codebeispiel zeigt die Verwendung `disassociate-members`.

AWS CLI

Um die Zuordnung von Mitgliedskonten zu trennen

Im folgenden `disassociate-members` Beispiel wird die Verbindung zwischen den angegebenen Mitgliedskonten und dem anfordernden Administratorkonto getrennt.

```
aws securityhub disassociate-members \  
  --account-ids "123456789111" "123456789222"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [DisassociateMembers](#) in der AWS CLI Befehlsreferenz.

enable-import-findings-for-product

Das folgende Codebeispiel zeigt die Verwendung `enable-import-findings-for-product`.

AWS CLI

Um zu beginnen, Erkenntnisse aus einer Produktintegration zu erhalten

Das folgende `enable-import-findings-for-product` Beispiel ermöglicht den Fluss von Erkenntnissen aus der angegebenen Produktintegration.

```
aws securityhub enable-import-findings-for-product \  
  --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/  
crowdstrike-falcon"
```

Ausgabe:

```
{
  "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon"
}
```

Weitere Informationen finden Sie unter [Verwaltung von Produktintegrationen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableImportFindingsForProduct](#) in der AWS CLI Befehlsreferenz.

enable-organization-admin-account

Das folgende Codebeispiel zeigt die Verwendung von `enable-organization-admin-account`.

AWS CLI

So legen Sie ein Organisationskonto als Security Hub-Administratorkonto fest

Im folgenden `enable-organization-admin-account` Beispiel wird das angegebene Konto als Security Hub-Administratorkonto bezeichnet.

```
aws securityhub enable-organization-admin-account \
  --admin-account-id 777788889999
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ein Security Hub-Administratorkonto einrichten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableOrganizationAdminAccount](#) in der AWS CLI Befehlsreferenz.

enable-security-hub

Das folgende Codebeispiel zeigt die Verwendung von `enable-security-hub`.

AWS CLI

Um AWS Security Hub zu aktivieren

Das folgende `enable-security-hub` Beispiel aktiviert AWS Security Hub für das anfordernde Konto. Es konfiguriert Security Hub so, dass die Standardstandards aktiviert werden. Für die Hub-Ressource weist es dem Tag den Wert `Security` zu. `Department`

```
aws securityhub enable-security-hub \  
  --enable-default-standards \  
  --tags '{"Department": "Security"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Enabling Security Hub](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [EnableSecurityHub](#) in der AWS CLI Befehlsreferenz.

get-administrator-account

Das folgende Codebeispiel zeigt die Verwendung `get-administrator-account`.

AWS CLI

Um Informationen über ein Administratorkonto abzurufen

Im folgenden `get-administrator-account` Beispiel werden Informationen über das Administratorkonto für das anfragende Konto abgerufen.

```
aws securityhub get-administrator-account
```

Ausgabe:

```
{  
  "Master": {  
    "AccountId": "123456789012",  
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",  
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,  
    "MemberStatus": "ASSOCIATED"  
  }  
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetAdministratorAccount](#) in der AWS CLI Befehlsreferenz.

get-configuration-policy-association

Das folgende Codebeispiel zeigt die Verwendung `get-configuration-policy-association`.

AWS CLI

Um Details zur Konfigurationszuweisung für ein Ziel abzurufen

Im folgenden `get-configuration-policy-association` Beispiel werden Zuordnungsdetails für das angegebene Ziel abgerufen. Sie können eine Konto-ID, eine Organisationseinheits-ID oder die Stamm-ID für das Ziel angeben.

```
aws securityhub get-configuration-policy-association \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}
```

Ausgabe:

```
{
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "TargetId": "ou-6hi7-8j91kl2m",
  "TargetType": "ORGANIZATIONAL_UNIT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
  "AssociationStatus": "SUCCESS",
  "AssociationStatusMessage": "Association applied successfully on this target."
}
```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien anzeigen](#) im AWS Security Hub Benutzerhandbuch.

- API-Einheiten finden Sie [GetConfigurationPolicyAssociation](#) in der AWS CLI Befehlsreferenz.

get-configuration-policy

Das folgende Codebeispiel zeigt die Verwendung `get-configuration-policy`.

AWS CLI

Um Details zur Konfigurationsrichtlinie anzuzeigen

Im folgenden `get-configuration-policy` Beispiel werden Details zur angegebenen Konfigurationsrichtlinie abgerufen.

```
aws securityhub get-configuration-policy \
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Ausgabe:

```
{
  "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Id": "ce5ed1e7-9639-4e2f-9313-fa87fcef944b",
  "Name": "SampleConfigurationPolicy",
  "Description": "SampleDescription",
  "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
  "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
  "ConfigurationPolicy": {
    "SecurityHub": {
      "ServiceEnabled": true,
      "EnabledStandardIdentifiers": [
        "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
        "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
      ],
      "SecurityControlsConfiguration": {
        "DisabledSecurityControlIdentifiers": [
          "CloudTrail.2"
        ],
        "SecurityControlCustomParameters": [
          {
            "SecurityControlId": "ACM.1",
            "Parameters": {
              "daysToExpiration": {
                "ValueType": "CUSTOM",
                "Value": {
                  "Integer": 15
                }
              }
            }
          }
        ]
      }
    }
  }
}
```

```

    }
  }
}

```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetConfigurationPolicy](#) in der AWS CLI Befehlsreferenz.

get-enabled-standards

Das folgende Codebeispiel zeigt die Verwendung `get-enabled-standards`.

AWS CLI

Um Informationen über einen aktivierten Standard abzurufen

Im folgenden `get-enabled-standards` Beispiel werden Informationen über den PCI DSS Standard abgerufen.

```

aws securityhub get-enabled-standards \
  --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"

```

Ausgabe:

```

{
  "StandardsSubscriptions": [
    {
      "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
      "StandardsInput": { },
      "StandardsStatus": "READY",
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Sicherheitsstandards in AWS Security Hub](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetEnabledStandards](#) in der AWS CLI Befehlsreferenz.

get-finding-aggregator

Das folgende Codebeispiel zeigt die Verwendung `get-finding-aggregator`.

AWS CLI

Um die aktuelle Konfiguration der Suchaggregation abzurufen

Im folgenden `get-finding-aggregator` Beispiel wird die aktuelle Konfiguration der Suchaggregation abgerufen.

```
aws securityhub get-finding-aggregator \
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
  aggregator/123e4567-e89b-12d3-a456-426652340000
```

Ausgabe:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
  aggregator/123e4567-e89b-12d3-a456-426652340000",
  "FindingAggregationRegion": "us-east-1",
  "RegionLinkingMode": "SPECIFIED_REGIONS",
  "Regions": "us-west-1,us-west-2"
}
```

Weitere Informationen [finden Sie unter Aktuelle Suchaggregationskonfiguration anzeigen](#) im AWS Security Hub Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetFindingAggregator AWS CLI](#) Befehlsreferenz.

get-finding-history

Das folgende Codebeispiel zeigt die Verwendung `get-finding-history`.

AWS CLI

Um den Suchverlauf zu finden

Im folgenden `get-finding-history` Beispiel wird der Verlauf der letzten 90 Tage für das angegebene Ergebnis abgerufen. In diesem Beispiel sind die Ergebnisse auf zwei Datensätze mit Fundverlauf beschränkt.

```
aws securityhub get-finding-history \
  --finding-identifier Id="arn:aws:securityhub:us-
  east-1:123456789012:security-control/S3.17/finding/a1b2c3d4-5678-90ab-cdef-
  EXAMPLE11111",ProductArn="arn:aws:securityhub:us-east-1::product/aws/securityhub"
```

Ausgabe:

```
{
  "Records": [
    {
      "FindingIdentifier": {
        "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
        S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
        securityhub"
      },
      "UpdateTime": "2023-06-02T03:15:25.685000+00:00",
      "FindingCreated": false,
      "UpdateSource": {
        "Type": "BATCH_IMPORT_FINDINGS",
        "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
      },
      "Updates": [
        {
          "UpdatedField": "Compliance.RelatedRequirements",
          "OldValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
          SC-12(3)\",\"NIST.800-53.r5 SC-12(6)\",\"NIST.800-53.r5 CM-3(6)\",\"NIST.800-53.r5
          SC-13\", \"NIST.800-53.r5 SC-28\", \"NIST.800-53.r5 SC-28(1)\", \"NIST.800-53.r5
          SC-7(10)\"]",
          "NewValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
          CM-3(6)\",\"NIST.800-53.r5 SC-13\", \"NIST.800-53.r5 SC-28\", \"NIST.800-53.r5
          SC-28(1)\", \"NIST.800-53.r5 SC-7(10)\", \"NIST.800-53.r5 CA-9(1)\", \"NIST.800-53.r5
          SI-7(6)\", \"NIST.800-53.r5 AU-9\"]"
        },
        {
          "UpdatedField": "LastObservedAt",
          "OldValue": "2023-06-01T09:15:38.587Z",
          "NewValue": "2023-06-02T03:15:22.946Z"
        }
      ]
    }
  ]
}
```

```

        {
            "UpdatedField": "UpdatedAt",
            "OldValue": "2023-06-01T09:15:31.049Z",
            "NewValue": "2023-06-02T03:15:14.861Z"
        },
        {
            "UpdatedField": "ProcessedAt",
            "OldValue": "2023-06-01T09:15:41.058Z",
            "NewValue": "2023-06-02T03:15:25.685Z"
        }
    ]
},
{
    "FindingIdentifier": {
        "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
securityhub"
    },
    "UpdateTime": "2023-05-23T02:06:51.518000+00:00",
    "FindingCreated": "true",
    "UpdateSource": {
        "Type": "BATCH_IMPORT_FINDINGS",
        "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
    },
    "Updates": []
}
]
}

```

Weitere Informationen [finden Sie unter Verlauf finden](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetFindingHistory](#) in der AWS CLI Befehlsreferenz.

get-findings

Das folgende Codebeispiel zeigt die Verwendung `get-findings`.

AWS CLI

Beispiel 1: Um Ergebnisse zurückzugeben, die für einen bestimmten Standard generiert wurden

Das folgende `get-findings` Beispiel gibt Ergebnisse für den PCI DSS Standard zurück.

```
aws securityhub get-findings \  
  --filters '{"GeneratorId":[{"Value": "pci-dss", "Comparison": "PREFIX"}]}' \  
  --max-items 1
```

Ausgabe:

```
{  
  "Findings": [  
    {  
      "SchemaVersion": "2018-10-08",  
      "Id": "arn:aws:securityhub:eu-central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub",  
      "GeneratorId": "pci-dss/v/3.2.1/PCI.Lambda.2",  
      "AwsAccountId": "123456789012",  
      "Types": [  
        "Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"  
      ],  
      "FindingProviderFields": {  
        "Severity": {  
          "Original": 0,  
          "Label": "INFORMATIONAL"  
        },  
        "Types": [  
          "Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"  
        ]  
      },  
      "FirstObservedAt": "2020-06-02T14:02:49.159Z",  
      "LastObservedAt": "2020-06-02T14:02:52.397Z",  
      "CreatedAt": "2020-06-02T14:02:49.159Z",  
      "UpdatedAt": "2020-06-02T14:02:52.397Z",  
      "Severity": {  
        "Original": 0,  
        "Label": "INFORMATIONAL",  
        "Normalized": 0  
      },  
      "Title": "PCI.Lambda.2 Lambda functions should be in a VPC",  
      "Description": "This AWS control checks whether a Lambda function is in a VPC.",  
    }  
  ]  
}
```

```
    "Remediation": {
      "Recommendation": {
        "Text": "For directions on how to fix this issue, please consult
the AWS Security Hub PCI DSS documentation.",
        "Url": "https://docs.aws.amazon.com/console/securityhub/
PCI.Lambda.2/remediation"
      }
    },
    "ProductFields": {
      "StandardsArn": "arn:aws:securityhub::standards/pci-dss/v/3.2.1",
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1",
      "ControlId": "PCI.Lambda.2",
      "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/PCI.Lambda.2/remediation",
      "RelatedAWSResources:0/name": "securityhub-lambda-inside-
vpc-0e904a3b",
      "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
      "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.Lambda.2",
      "aws/securityhub/SeverityLabel": "INFORMATIONAL",
      "aws/securityhub/ProductName": "Security Hub",
      "aws/securityhub/CompanyName": "AWS",
      "aws/securityhub/FindingId": "arn:aws:securityhub:eu-
central-1::product/aws/securityhub/arn:aws:securityhub:eu-
central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "Resources": [
      {
        "Type": "AwsAccount",
        "Id": "AWS:::Account:123456789012",
        "Partition": "aws",
        "Region": "us-west-1"
      }
    ],
    "Compliance": {
      "Status": "PASSED",
      "RelatedRequirements": [
        "PCI DSS 1.2.1",
        "PCI DSS 1.3.1",
        "PCI DSS 1.3.2",
        "PCI DSS 1.3.4"
      ]
    }
  ]
}
```

```

    },
    "WorkflowState": "NEW",
    "Workflow": {
      "Status": "NEW"
    },
    "RecordState": "ARCHIVED"
  }
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}

```

Beispiel 2: Um Ergebnisse mit kritischem Schweregrad zurückzugeben, die den Workflow-Status von NOTIFIED

Im folgenden `get-findings` Beispiel werden Ergebnisse zurückgegeben, die den Schweregradwert CRITICAL und den Workflow-Status haben. NOTIFIED Die Ergebnisse sind in absteigender Reihenfolge nach dem Wert Confidence sortiert.

```

aws securityhub get-findings \
  --filters '{"SeverityLabel":[{"Value":
"CRITICAL", "Comparison": "EQUALS"}], "WorkflowStatus":
[{"Value": "NOTIFIED", "Comparison": "EQUALS"}]}' \
  --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
  --max-items 1

```

Ausgabe:

```

{
  "Findings": [
    {
      "SchemaVersion": "2018-10-08",
      "Id": "arn:aws:securityhub:us-west-1: 123456789012:subscription/cis-aws-
foundations-benchmark/v/1.2.0/1.13/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/securityhub",
      "GeneratorId": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0/rule/1.13",
      "AwsAccountId": "123456789012",
      "Types": [
        "Software and Configuration Checks/Industry and Regulatory
Standards/CIS AWS Foundations Benchmark"
      ],
      "FindingProviderFields" {

```

```
    "Severity": {
      "Original": 90,
      "Label": "CRITICAL"
    },
    "Types": [
      "Software and Configuration Checks/Industry and Regulatory
Standards/CIS AWS Foundations Benchmark"
    ]
  },
  "FirstObservedAt": "2020-05-21T20:16:34.752Z",
  "LastObservedAt": "2020-06-09T08:16:37.171Z",
  "CreatedAt": "2020-05-21T20:16:34.752Z",
  "UpdatedAt": "2020-06-09T08:16:36.430Z",
  "Severity": {
    "Original": 90,
    "Label": "CRITICAL",
    "Normalized": 90
  },
  "Title": "1.13 Ensure MFA is enabled for the \"root\" account",
  "Description": "The root account is the most privileged user in an AWS
account. MFA adds an extra layer of protection on top of a user name and password.
With MFA enabled, when a user signs in to an AWS website, they will be prompted for
their user name and password as well as for an authentication code from their AWS
MFA device.",
  "Remediation": {
    "Recommendation": {
      "Text": "For directions on how to fix this issue, please consult
the AWS Security Hub CIS documentation.",
      "Url": "https://docs.aws.amazon.com/console/securityhub/
standards-cis-1.13/remediation"
    }
  },
  "ProductFields": {
    "StandardsGuideArn": "arn:aws:securityhub::ruleset/cis-aws-
foundations-benchmark/v/1.2.0",
    "StandardsGuideSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0",
    "RuleId": "1.13",
    "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/standards-cis-1.13/remediation",
    "RelatedAWSResources:0/name": "securityhub-root-account-mfa-
enabled-5pftha",
    "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
```

```

        "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/cis-aws-foundations-benchmark/v/1.2.0/1.13",
        "aws/securityhub/SeverityLabel": "CRITICAL",
        "aws/securityhub/ProductName": "Security Hub",
        "aws/securityhub/CompanyName": "AWS",
        "aws/securityhub/FindingId": "arn:aws:securityhub:us-
west-1::product/aws/securityhub/arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0/1.13/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "Resources": [
        {
            "Type": "AwsAccount",
            "Id": "AWS:::Account:123456789012",
            "Partition": "aws",
            "Region": "us-west-1"
        }
    ],
    "Compliance": {
        "Status": "FAILED"
    },
    "WorkflowState": "NEW",
    "Workflow": {
        "Status": "NOTIFIED"
    },
    "RecordState": "ACTIVE"
}
]
}

```

Weitere Informationen finden Sie unter [Ergebnisse filtern und gruppieren](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetFindings](#) in der AWS CLI Befehlsreferenz.

get-insight-results

Das folgende Codebeispiel zeigt die Verwendung `get-insight-results`.

AWS CLI

Um die Ergebnisse für einen Einblick abzurufen

Das folgende `get-insight-results` Beispiel gibt die Liste der Insight-Ergebnisse für den Insight mit den angegebenen Werten zurückARN.

```
aws securityhub get-insight-results \
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Ausgabe:

```
{
  "InsightResults": {
    "GroupByAttribute": "ResourceId",
    "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
    "ResultValues": [
      {
        "Count": 10,
        "GroupByAttributeValue": "AWS:::Account:123456789111"
      },
      {
        "Count": 3,
        "GroupByAttributeValue": "AWS:::Account:123456789222"
      }
    ]
  }
}
```

Weitere Informationen finden Sie im AWS Security Hub Hub-Benutzerhandbuch [unter Insight-Ergebnisse und Erkenntnisse anzeigen und Maßnahmen ergreifen](#).

- API-Einheiten finden Sie [GetInsightResults](#) in der AWS CLI Befehlsreferenz.

get-insights

Das folgende Codebeispiel zeigt die Verwendung `get-insights`.

AWS CLI

Um Details zu einem Insight abzurufen

Im folgenden `get-insights` Beispiel werden die Konfigurationsdetails für den Insight mit den angegebenen ARN Werten abgerufen.

```
aws securityhub get-insights \
  --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Ausgabe:

```
{
  "Insights": [
    {
      "Filters": {
        "ResourceType": [
          {
            "Comparison": "EQUALS",
            "Value": "AwsIamRole"
          }
        ],
        "SeverityLabel": [
          {
            "Comparison": "EQUALS",
            "Value": "CRITICAL"
          }
        ],
      },
      "GroupByAttribute": "ResourceId",
      "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
      "Name": "Critical role findings"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Insights in AWS Security Hub](#) im AWS Security Hub Benutzerhandbuch.

- API-Einheiten finden Sie [GetInsights](#) in der AWS CLI Befehlsreferenz.

get-invitations-count

Das folgende Codebeispiel zeigt die Verwendung `get-invitations-count`.

AWS CLI

Um die Anzahl der Einladungen abzurufen, die nicht akzeptiert wurden

Im folgenden `get-invitations-count` Beispiel wird die Anzahl der Einladungen abgerufen, die das anfragende Konto abgelehnt hat oder auf die das Konto nicht geantwortet hat.

```
aws securityhub get-invitations-count
```

Ausgabe:

```
{
  "InvitationsCount": 3
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetInvitationsCount](#) in der AWS CLI Befehlsreferenz.

get-master-account

Das folgende Codebeispiel zeigt die Verwendung `get-master-account`.

AWS CLI

Um Informationen über ein Administratorkonto abzurufen

Im folgenden `get-master-account` Beispiel werden Informationen über das Administratorkonto für das anfragende Konto abgerufen.

```
aws securityhub get-master-account
```

Ausgabe:

```
{
```

```
"Master": {
  "AccountId": "123456789012",
  "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
  "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
  "MemberStatus": "ASSOCIATED"
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMasterAccount](#) in der AWS CLI Befehlsreferenz.

get-members

Das folgende Codebeispiel zeigt die Verwendung `get-members`.

AWS CLI

Um Informationen über ausgewählte Mitgliedskonten abzurufen

Im folgenden `get-members` Beispiel werden Informationen zu den angegebenen Mitgliedskonten abgerufen.

```
aws securityhub get-members \
  --account-ids "444455556666" "777788889999"
```

Ausgabe:

```
{
  "Members": [
    {
      "AccountId": "123456789111",
      "AdministratorId": "123456789012",
      "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
      "MasterId": "123456789012",
      "MemberStatus": "ASSOCIATED",
      "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
    },
    {
      "AccountId": "123456789222",
      "AdministratorId": "123456789012",
```

```

    "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
    "MasterId": "123456789012",
    "MemberStatus": "ASSOCIATED",
    "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
  }
],
"UnprocessedAccounts": [ ]
}

```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMembers](#) in der AWS CLI Befehlsreferenz.

get-security-control-definition

Das folgende Codebeispiel zeigt die Verwendung von `get-security-control-definition`.

AWS CLI

Um Details zur Definition der Sicherheitskontrolle abzurufen

Im folgenden `get-security-control-definition` Beispiel werden Definitionsdetails für eine Security Hub-Sicherheitskontrolle abgerufen. Zu den Details gehören der Titel des Steuerelements, die Beschreibung, die Verfügbarkeit in der Region, Parameter und andere Informationen.

```

aws securityhub get-security-control-definition \
  --security-control-id ACM.1

```

Ausgabe:

```

{
  "SecurityControlDefinition": {
    "SecurityControlId": "ACM.1",
    "Title": "Imported and ACM-issued certificates should be renewed after a
specified time period",
    "Description": "This control checks whether an AWS Certificate Manager
(ACM) certificate is renewed within the specified time period. It checks both
imported certificates and certificates provided by ACM. The control fails if the
certificate isn't renewed within the specified time period. Unless you provide a

```

```

custom parameter value for the renewal period, Security Hub uses a default value of
30 days.",
    "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/ACM.1/
remediation",
    "SeverityRating": "MEDIUM",
    "CurrentRegionAvailability": "AVAILABLE",
    "ParameterDefinitions": {
        "daysToExpiration": {
            "Description": "Number of days within which the ACM certificate must
be renewed",
            "ConfigurationOptions": {
                "Integer": {
                    "DefaultValue": 30,
                    "Min": 14,
                    "Max": 365
                }
            }
        }
    }
}
}
}
}
}
}
}
}
}
}
}
}
}

```

Weitere Informationen finden Sie unter [Benutzerdefinierte Steuerungsparameter](#) im AWS Security Hub Benutzerhandbuch.

- API-Einheiten finden Sie [GetSecurityControlDefinition](#) in der AWS CLI Befehlsreferenz.

invite-members

Das folgende Codebeispiel zeigt die Verwendung `invite-members`.

AWS CLI

Um Einladungen an Mitgliedskonten zu senden

Im folgenden `invite-members` Beispiel werden Einladungen an die angegebenen Mitgliedskonten gesendet.

```
aws securityhub invite-members \
  --account-ids "123456789111" "123456789222"
```

Ausgabe:

```
{
  "UnprocessedAccounts": []
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [InviteMembers](#) in der AWS CLI Befehlsreferenz.

list-automation-rules

Das folgende Codebeispiel zeigt die Verwendung `list-automation-rules`.

AWS CLI

Um eine Liste von Automatisierungsregeln anzuzeigen

Das folgende `list-automation-rules` Beispiel listet die Automatisierungsregeln für ein AWS Konto auf. Nur das Security Hub-Administratorkonto kann diesen Befehl ausführen.

```
aws securityhub list-automation-rules \
  --max-results 3 \
  --next-token NULL
```

Ausgabe:

```
{
  "AutomationRulesMetadata": [
    {
      "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "RuleStatus": "ENABLED",
      "RuleOrder": 1,
      "RuleName": "Suppress informational findings",
      "Description": "Suppress GuardDuty findings with Informational severity",
      "IsTerminal": false,
      "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
      "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
      "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
    },
    {
```

```

    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "RuleStatus": "ENABLED",
    "RuleOrder": 1,
    "RuleName": "sample rule",
    "Description": "A sample rule",
    "IsTerminal": false,
    "CreatedAt": "2023-07-15T23:37:20.223000+00:00",
    "UpdatedAt": "2023-07-15T23:37:20.223000+00:00",
    "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
  },
  {
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "RuleStatus": "ENABLED",
    "RuleOrder": 1,
    "RuleName": "sample rule",
    "Description": "A sample rule",
    "IsTerminal": false,
    "CreatedAt": "2023-07-15T23:45:25.126000+00:00",
    "UpdatedAt": "2023-07-15T23:45:25.126000+00:00",
    "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
  }
]
}

```

Weitere Informationen finden Sie unter [Automatisierungsregeln anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAutomationRules](#) in der AWS CLI Befehlsreferenz.

list-configuration-policies

Das folgende Codebeispiel zeigt die Verwendung `list-configuration-policies`.

AWS CLI

Um Zusammenfassungen der Konfigurationsrichtlinien aufzulisten

Im folgenden `list-configuration-policies` Beispiel wird eine Zusammenfassung der Konfigurationsrichtlinien für die Organisation aufgeführt.

```
aws securityhub list-configuration-policies \
```

```
--max-items 3
```

Ausgabe:

```
{
  "ConfigurationPolicySummaries": [
    {
      "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Name": "SampleConfigurationPolicy1",
      "Description": "SampleDescription1",
      "UpdatedAt": "2023-09-26T21:08:36.214000+00:00",
      "ServiceEnabled": true
    },
    {
      "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "Name": "SampleConfigurationPolicy2",
      "Description": "SampleDescription2",
      "UpdatedAt": "2023-11-28T19:26:25.207000+00:00",
      "ServiceEnabled": true
    },
    {
      "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "Name": "SampleConfigurationPolicy3",
      "Description": "SampleDescription3",
      "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
      "ServiceEnabled": true
    }
  ]
}
```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListConfigurationPolicies](#) in der AWS CLI Befehlsreferenz.

list-configuration-policy-associations

Das folgende Codebeispiel zeigt die Verwendung `list-configuration-policy-associations`.

AWS CLI

Um Konfigurationszuordnungen aufzulisten

Im folgenden `list-configuration-policy-associations` Beispiel wird eine Zusammenfassung der Konfigurationszuordnungen für die Organisation aufgeführt. Die Antwort beinhaltet Verknüpfungen zu Konfigurationsrichtlinien und selbstverwaltetem Verhalten.

```
aws securityhub list-configuration-policy-associations \  
  --association-type "APPLIED" \  
  --max-items 4
```

Ausgabe:

```
{  
  "ConfigurationPolicyAssociationSummaries": [  
    {  
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "TargetId": "r-1ab2",  
      "TargetType": "ROOT",  
      "AssociationType": "APPLIED",  
      "UpdatedAt": "2023-11-28T19:26:49.417000+00:00",  
      "AssociationStatus": "FAILED",  
      "AssociationStatusMessage": "Policy association failed because 2  
organizational units or accounts under this root failed."  
    },  
    {  
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "TargetId": "ou-1ab2-c3de4f5g",  
      "TargetType": "ORGANIZATIONAL_UNIT",  
      "AssociationType": "APPLIED",  
      "UpdatedAt": "2023-09-26T21:14:05.283000+00:00",  
      "AssociationStatus": "FAILED",  
      "AssociationStatusMessage": "One or more children under this target  
failed association."  
    },  
    {  
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
      "TargetId": "ou-6hi7-8j91kl2m",  
      "TargetType": "ORGANIZATIONAL_UNIT",  
      "AssociationType": "APPLIED",  
      "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",  
      "AssociationStatus": "SUCCESS",  
    }  
  ]  
}
```

```
target."
  },
  {
    "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
    "TargetId": "111122223333",
    "TargetType": "ACCOUNT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-28T22:01:26.409000+00:00",
    "AssociationStatus": "SUCCESS"
  }
}
```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListConfigurationPolicyAssociations](#) in der AWS CLI Befehlsreferenz.

list-enabled-products-for-import

Das folgende Codebeispiel zeigt die Verwendung `list-enabled-products-for-import`.

AWS CLI

Um die Liste der aktivierten Produktintegrationen zurückzugeben

Im folgenden `list-enabled-products-for-import` Beispiel wird die Liste der Abonnements ARNs für die aktuell aktivierten Produktintegrationen zurückgegeben.

```
aws securityhub list-enabled-products-for-import
```

Ausgabe:

```
{
  "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-
west-1:123456789012:product-subscription/aws/securityhub" ]
}
```

Weitere Informationen finden Sie unter [Verwaltung von Produktintegrationen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListEnabledProductsForImport](#) in der AWS CLI Befehlsreferenz.

list-finding-aggregators

Das folgende Codebeispiel zeigt die Verwendung `list-finding-aggregators`.

AWS CLI

Um die verfügbaren Widgets aufzulisten

Das folgende `list-finding-aggregators` Beispiel gibt die Konfiguration ARN der Finding-Aggregation zurück.

```
aws securityhub list-finding-aggregators
```

Ausgabe:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000"
}
```

Weitere Informationen [finden Sie unter Aktuelle Suchaggregationskonfiguration anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListFindingAggregators AWS CLI](#) Befehlsreferenz.

list-invitations

Das folgende Codebeispiel zeigt die Verwendung `list-invitations`.

AWS CLI

Um eine Liste mit Einladungen anzuzeigen

Im folgenden `list-invitations` Beispiel wird die Liste der Einladungen abgerufen, die an das anfragende Konto gesendet wurden.

```
aws securityhub list-invitations
```

Ausgabe:

```
{
  "Invitations": [
```

```
    {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
    }
  ],
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListInvitations](#) in der AWS CLI Befehlsreferenz.

list-members

Das folgende Codebeispiel zeigt die Verwendung `list-members`.

AWS CLI

Um eine Liste von Mitgliedskonten abzurufen

Im folgenden `list-members` Beispiel wird die Liste der Mitgliedskonten für das anfragende Administratorkonto zurückgegeben.

```
aws securityhub list-members
```

Ausgabe:

```
{
  "Members": [
    {
      "AccountId": "123456789111",
      "AdministratorId": "123456789012",
      "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
      "MasterId": "123456789012",
      "MemberStatus": "ASSOCIATED",
      "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
    },
    {
      "AccountId": "123456789222",
      "AdministratorId": "123456789012",
      "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
```

```
        "MasterId": "123456789012",
        "MemberStatus": "ASSOCIATED",
        "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
    },
],
}
```

Weitere Informationen finden Sie unter [Verwaltung von Administrator- und Mitgliedskonten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListMembers](#) in der AWS CLI Befehlsreferenz.

list-organization-admin-accounts

Das folgende Codebeispiel zeigt die Verwendung `list-organization-admin-accounts`.

AWS CLI

Um die designierten Security Hub-Administratorkonten aufzulisten

Das folgende `list-organization-admin-accounts` Beispiel listet die Security Hub-Administratorkonten für eine Organisation auf.

```
aws securityhub list-organization-admin-accounts
```

Ausgabe:

```
{
  AdminAccounts": [
    { "AccountId": "777788889999" },
    { "Status": "ENABLED" }
  ]
}
```

Weitere Informationen finden Sie unter [Ein Security Hub-Administratorkonto einrichten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListOrganizationAdminAccounts](#) in der AWS CLI Befehlsreferenz.

list-security-control-definitions

Das folgende Codebeispiel zeigt die Verwendung `list-security-control-definitions`.

AWS CLI

Beispiel 1: Um alle verfügbaren Sicherheitskontrollen aufzulisten

Das folgende `list-security-control-definitions` Beispiel listet die verfügbaren Sicherheitskontrollen für alle Security Hub Hub-Standards auf. In diesem Beispiel werden die Ergebnisse auf drei Kontrollen beschränkt.

```
aws securityhub list-security-control-definitions \  
  --max-items 3
```

Ausgabe:

```
{  
  "SecurityControlDefinitions": [  
    {  
      "SecurityControlId": "ACM.1",  
      "Title": "Imported and ACM-issued certificates should be renewed after a  
specified time period",  
      "Description": "This control checks whether an AWS Certificate Manager  
(ACM) certificate is renewed within the specified time period. It checks both  
imported certificates and certificates provided by ACM. The control fails if the  
certificate isn't renewed within the specified time period. Unless you provide a  
custom parameter value for the renewal period, Security Hub uses a default value of  
30 days.",  
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/  
ACM.1/remediation",  
      "SeverityRating": "MEDIUM",  
      "CurrentRegionAvailability": "AVAILABLE",  
      "CustomizableProperties": [  
        "Parameters"  
      ]  
    },  
    {  
      "SecurityControlId": "ACM.2",  
      "Title": "RSA certificates managed by ACM should use a key length of at  
least 2,048 bits",  
      "Description": "This control checks whether RSA certificates managed by  
AWS Certificate Manager use a key length of at least 2,048 bits. The control fails  
if the key length is smaller than 2,048 bits.",  
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/  
ACM.2/remediation",  
      "SeverityRating": "HIGH",
```

```

        "CurrentRegionAvailability": "AVAILABLE",
        "CustomizableProperties": []
    },
    {
        "SecurityControlId": "APIGateway.1",
        "Title": "API Gateway REST and WebSocket API execution logging should be
enabled",
        "Description": "This control checks whether all stages of an Amazon
API Gateway REST or WebSocket API have logging enabled. The control fails if
the 'loggingLevel' isn't 'ERROR' or 'INFO' for all stages of the API. Unless you
provide custom parameter values to indicate that a specific log type should be
enabled, Security Hub produces a passed finding if the logging level is either
'ERROR' or 'INFO'.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
APIGateway.1/remediation",
        "SeverityRating": "MEDIUM",
        "CurrentRegionAvailability": "AVAILABLE",
        "CustomizableProperties": [
            "Parameters"
        ]
    }
],
    "NextToken": "U2FsdGVkX1/UpRCPzxVbkDeHikDXbDxfgJZ1w2RG1XWsFPTMTIQPVE0m/
FduIGxS70bRtAbaUt/8/RCQcg2PU0YXI20hH/Grho0Tgv+Tsm0qvQVFhkJepWmqh
+NYawjocVBeos6xzn/8qnbF9IuwGg=="
}

```

Weitere Informationen finden Sie unter [Details für einen Standard anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

Beispiel 2: Um die verfügbaren Sicherheitskontrollen für einen bestimmten Standard aufzulisten

Das folgende `list-security-control-definitions` Beispiel listet die verfügbaren Sicherheitskontrollen für den CIS AWS Foundations Benchmark v1.4.0 auf. In diesem Beispiel werden die Ergebnisse auf drei Kontrollen beschränkt.

```

aws securityhub list-security-control-definitions \
  --standards-arn "arn:aws:securityhub:us-east-1::standards/cis-aws-foundations-
benchmark/v/1.4.0" \
  --max-items 3

```

Ausgabe:

```
{
  "SecurityControlDefinitions": [
    {
      "SecurityControlId": "CloudTrail.1",
      "Title": "CloudTrail should be enabled and configured with at least one
multi-Region trail that includes read and write management events",
      "Description": "This AWS control checks that there is at least one
multi-region AWS CloudTrail trail includes read and write management events.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.1/remediation",
      "SeverityRating": "HIGH",
      "CurrentRegionAvailability": "AVAILABLE",
      "CustomizableProperties": []
    },
    {
      "SecurityControlId": "CloudTrail.2",
      "Title": "CloudTrail should have encryption at-rest enabled",
      "Description": "This AWS control checks whether AWS CloudTrail is
configured to use the server side encryption (SSE) AWS Key Management Service (AWS
KMS) customer master key (CMK) encryption. The check will pass if the KmsKeyId is
defined.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.2/remediation",
      "SeverityRating": "MEDIUM",
      "CurrentRegionAvailability": "AVAILABLE",
      "CustomizableProperties": []
    },
    {
      "SecurityControlId": "CloudTrail.4",
      "Title": "CloudTrail log file validation should be enabled",
      "Description": "This AWS control checks whether CloudTrail log file
validation is enabled.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.4/remediation",
      "SeverityRating": "MEDIUM",
      "CurrentRegionAvailability": "AVAILABLE",
      "CustomizableProperties": []
    }
  ],
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}
```


Weitere Informationen finden Sie unter [Details für einen Standard anzeigen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListSecurityControlDefinitions](#) in der AWS CLI Befehlsreferenz.

list-standards-control-associations

Das folgende Codebeispiel zeigt die Verwendung `list-standards-control-associations`.

AWS CLI

Um den Aktivierungsstatus eines Steuerelements in jedem aktivierten Standard abzurufen

Im folgenden `list-standards-control-associations` Beispiel wird der Aktivierungsstatus `CloudTrail .1` in jedem aktivierten Standard aufgeführt.

```
aws securityhub list-standards-control-associations \  
  --security-control-id CloudTrail.1
```

Ausgabe:

```
{  
  "StandardsControlAssociationSummaries": [  
    {  
      "StandardsArn": "arn:aws:securityhub:us-east-2::standards/nist-800-53/  
v/5.0.0",  
      "SecurityControlId": "CloudTrail.1",  
      "SecurityControlArn": "arn:aws:securityhub:us-  
east-2:123456789012:security-control/CloudTrail.1",  
      "AssociationStatus": "ENABLED",  
      "RelatedRequirements": [  
        "NIST.800-53.r5 AC-2(4)",  
        "NIST.800-53.r5 AC-4(26)",  
        "NIST.800-53.r5 AC-6(9)",  
        "NIST.800-53.r5 AU-10",  
        "NIST.800-53.r5 AU-12",  
        "NIST.800-53.r5 AU-2",  
        "NIST.800-53.r5 AU-3",  
        "NIST.800-53.r5 AU-6(3)",  
        "NIST.800-53.r5 AU-6(4)",  
        "NIST.800-53.r5 AU-14(1)",  
        "NIST.800-53.r5 CA-7",  
        "NIST.800-53.r5 SC-7(9)",  
      ]  
    }  
  ]  
}
```

```

        "NIST.800-53.r5 SI-3(8)",
        "NIST.800-53.r5 SI-4(20)",
        "NIST.800-53.r5 SI-7(8)",
        "NIST.800-53.r5 SA-8(22)"
    ],
    "UpdatedAt": "2023-05-15T17:52:21.304000+00:00",
    "StandardsControlTitle": "CloudTrail should be enabled and configured
with at least one multi-Region trail that includes read and write management
events",
    "StandardsControlDescription": "This AWS control checks that there is
at least one multi-region AWS CloudTrail trail includes read and write management
events."
  },
  {
    "StandardsArn": "arn:aws:securityhub::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
    "SecurityControlId": "CloudTrail.1",
    "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
    "AssociationStatus": "ENABLED",
    "RelatedRequirements": [
      "CIS AWS Foundations 2.1"
    ],
    "UpdatedAt": "2020-02-10T21:22:53.998000+00:00",
    "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
    "StandardsControlDescription": "AWS CloudTrail is a web service that
records AWS API calls for your account and delivers log files to you. The recorded
information includes the identity of the API caller, the time of the API call,
the source IP address of the API caller, the request parameters, and the response
elements returned by the AWS service."
  },
  {
    "StandardsArn": "arn:aws:securityhub:us-east-2::standards/aws-
foundational-security-best-practices/v/1.0.0",
    "SecurityControlId": "CloudTrail.1",
    "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
    "AssociationStatus": "DISABLED",
    "RelatedRequirements": [],
    "UpdatedAt": "2023-05-15T19:31:52.671000+00:00",
    "UpdatedReason": "Alternative compensating controls are in place",
    "StandardsControlTitle": "CloudTrail should be enabled and configured
with at least one multi-Region trail that includes read and write management
events",

```

```

        "StandardsControlDescription": "This AWS control checks that there is
at least one multi-region AWS CloudTrail trail includes read and write management
events."
    },
    {
        "StandardsArn": "arn:aws:securityhub:us-east-2::standards/cis-aws-
foundations-benchmark/v/1.4.0",
        "SecurityControlId": "CloudTrail.1",
        "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
        "AssociationStatus": "ENABLED",
        "RelatedRequirements": [
            "CIS AWS Foundations Benchmark v1.4.0/3.1"
        ],
        "UpdatedAt": "2022-11-10T15:40:36.021000+00:00",
        "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
        "StandardsControlDescription": "AWS CloudTrail is a web service that
records AWS API calls for your account and delivers log files to you. The recorded
information includes the identity of the API caller, the time of the API call,
the source IP address of the API caller, the request parameters, and the response
elements returned by the AWS service. CloudTrail provides a history of AWS API
calls for an account, including API calls made via the Management Console, SDKs,
command line tools, and higher-level AWS services (such as CloudFormation)."
    }
]
}

```

Weitere Informationen finden Sie unter [Aktivieren und Deaktivieren von Steuerungen in bestimmten Standards](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [ListStandardsControlAssociations](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die einer Ressource zugewiesenen Tags abzurufen

Im folgenden `list-tags-for-resource` Beispiel werden die der angegebenen Hub-Ressource zugewiesenen Tags zurückgegeben.

```
aws securityhub list-tags-for-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Ausgabe:

```
{  
  "Tags": {  
    "Department" : "Operations",  
    "Area" : "USMidwest"  
  }  
}
```

Weitere Informationen finden Sie unter [AWS:SecurityHub: :Hub](#) im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

start-configuration-policy-association

Das folgende Codebeispiel zeigt die Verwendung `start-configuration-policy-association`.

AWS CLI

Beispiel 1: Um eine Konfigurationsrichtlinie zuzuordnen

Im folgenden `start-configuration-policy-association` Beispiel wird die angegebene Konfigurationsrichtlinie der angegebenen Organisationseinheit zugeordnet. Eine Konfiguration kann einem Zielkonto, einer Organisationseinheit oder dem Stammkonto zugeordnet sein.

```
aws securityhub start-configuration-policy-association \  
  --configuration-policy-identifizier "arn:aws:securityhub:eu-  
  central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

Ausgabe:

```
{  
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
  "TargetId": "ou-6hi7-8j91kl2m",  
  "TargetType": "ORGANIZATIONAL_UNIT",  
  "AssociationType": "APPLIED",  
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",  
}
```

```
"AssociationStatus": "PENDING"
}
```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien erstellen und zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

Beispiel 2: So ordnen Sie eine selbstverwaltete Konfiguration zu

Im folgenden `start-configuration-policy-association` Beispiel wird dem angegebenen Konto eine selbstverwaltete Konfiguration zugeordnet.

```
aws securityhub start-configuration-policy-association \
  --configuration-policy-identifizier "SELF_MANAGED_SECURITY_HUB" \
  --target '{"OrganizationalUnitId": "123456789012"}
```

Ausgabe:

```
{
  "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
  "TargetId": "123456789012",
  "TargetType": "ACCOUNT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
  "AssociationStatus": "PENDING"
}
```

Weitere Informationen finden Sie unter [Security Hub Hub-Konfigurationsrichtlinien erstellen und zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie unter [StartConfigurationPolicyAssociation AWS CLI Befehlsreferenz](#).

start-configuration-policy-disassociation

Das folgende Codebeispiel zeigt die Verwendung `start-configuration-policy-disassociation`.

AWS CLI

Beispiel 1: Um die Zuordnung einer Konfigurationsrichtlinie aufzuheben

Im folgenden `start-configuration-policy-disassociation` Beispiel wird die Zuordnung einer Konfigurationsrichtlinie zur angegebenen Organisationseinheit aufgehoben.

Eine Konfiguration kann von einem Zielkonto, einer Organisationseinheit oder dem Stammkonto getrennt werden.

```
aws securityhub start-configuration-policy-disassociation \  
  --configuration-policy-identifizier "arn:aws:securityhub:eu-  
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Trennen einer Konfiguration von Konten und OUs](#) im AWS Security Hub Hub-Benutzerhandbuch.

Beispiel 2: So trennen Sie die Zuordnung einer selbstverwalteten Konfiguration

Im folgenden `start-configuration-policy-disassociation` Beispiel wird die Zuordnung einer selbstverwalteten Konfiguration zum angegebenen Konto aufgehoben.

```
aws securityhub start-configuration-policy-disassociation \  
  --configuration-policy-identifizier "SELF_MANAGED_SECURITY_HUB" \  
  --target '{"AccountId": "123456789012"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Trennen einer Konfiguration von Konten und OUs](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [StartConfigurationPolicyDisassociation](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer Ressource ein Tag zuzuweisen

Im folgenden `tag-resource` Beispiel werden der angegebenen Hub-Ressource Werte für die Tags `Department` und `Area` zugewiesen.

```
aws securityhub tag-resource \  
  --resource-arn "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --tag-key "Department" \  
  --tag-value "Engineering" \  
  --tag-key "Area" \  
  --tag-value "Development"
```

```
--resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
--tags '{"Department":"Operations", "Area":"USMidwest"}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS:SecurityHub: :Hub](#) im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um einen Tag-Wert aus einer Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das `Department`-Tag aus der angegebenen Hub-Ressource entfernt.

```
aws securityhub untag-resource \  
--resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
--tag-keys "Department"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS:SecurityHub: :Hub](#) im AWS CloudFormation Benutzerhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-action-target

Das folgende Codebeispiel zeigt die Verwendung `update-action-target`.

AWS CLI

Um eine benutzerdefinierte Aktion zu aktualisieren

Im folgenden `update-action-target` Beispiel wird der Name der benutzerdefinierten Aktion aktualisiert, die durch die angegebene Aktion identifiziert wirdARN.

```
aws securityhub update-action-target \
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/Remediation" \
  --name "Send to remediation"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Eine benutzerdefinierte Aktion erstellen und sie einer CloudWatch Ereignisregel zuordnen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateActionTarget](#) in der AWS CLI Befehlsreferenz.

update-configuration-policy

Das folgende Codebeispiel zeigt die Verwendung `update-configuration-policy`.

AWS CLI

Um eine Konfigurationsrichtlinie zu aktualisieren

Im folgenden `update-configuration-policy` Beispiel wird eine bestehende Konfigurationsrichtlinie aktualisiert, sodass sie die angegebenen Einstellungen verwendet.

```
aws securityhub update-configuration-policy \
  --identifier "arn:aws:securityhub:eu-central-1:508236694226:configuration-policy/09f37766-57d8-4ede-9d33-5d8b0fecf70e" \
  --name "SampleConfigurationPolicyUpdated" \
  --description "SampleDescriptionUpdated" \
  --configuration-policy '{"SecurityHub": {"ServiceEnabled": true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-central-1::standards/aws-foundational-security-best-practices/v/1.0.0", "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0"], "SecurityControlsConfiguration": {"DisabledSecurityControlIdentifiers": ["CloudWatch.1"], "SecurityControlCustomParameters": [{"SecurityControlId": "ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer": 21}}}}]}}' \
  --updated-reason "Disabling CloudWatch.1 and changing parameter value"
```

Ausgabe:

```
{
```



```

    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "SampleConfigurationPolicyUpdated",
    "Description": "SampleDescriptionUpdated",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
      "SecurityHub": {
        "ServiceEnabled": true,
        "EnabledStandardIdentifiers": [
          "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
          "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
        ],
        "SecurityControlsConfiguration": {
          "DisabledSecurityControlIdentifiers": [
            "CloudWatch.1"
          ],
          "SecurityControlCustomParameters": [
            {
              "SecurityControlId": "ACM.1",
              "Parameters": {
                "daysToExpiration": {
                  "ValueType": "CUSTOM",
                  "Value": {
                    "Integer": 21
                  }
                }
              }
            }
          ]
        }
      }
    }
  }
}

```

Weitere Informationen finden Sie unter [Aktualisieren der Security Hub Hub-Konfigurationsrichtlinien](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateConfigurationPolicy](#) in der AWS CLI Befehlsreferenz.

update-finding-aggregator

Das folgende Codebeispiel zeigt die Verwendung `update-finding-aggregator`.

AWS CLI

Um die aktuelle Konfiguration der Finding-Aggregation zu aktualisieren

Im folgenden `update-finding-aggregator` Beispiel wird die Konfiguration der Suchaggregation so geändert, dass ein Link aus ausgewählten Regionen hergestellt wird. Es wird von der Aggregationsregion USA Ost (Virginia) aus ausgeführt. Es wählt USA West (Nordkalifornien) und USA West (Oregon) als verknüpfte Regionen aus.

```
aws securityhub update-finding-aggregator \  
  --region us-east-1 \  
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-  
aggregator/123e4567-e89b-12d3-a456-426652340000 \  
  --region-linking-mode SPECIFIED_REGIONS \  
  --regions us-west-1,us-west-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen [finden Sie unter Aktualisierung der Suchaggregationskonfiguration](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UpdateFindingAggregator AWS CLI Befehlsreferenz](#).

update-insight

Das folgende Codebeispiel zeigt die Verwendung `update-insight`.

AWS CLI

Beispiel 1: Um den Filter für einen benutzerdefinierten Einblick zu ändern

Im folgenden `update-insight` Beispiel werden die Filter für einen benutzerdefinierten Einblick geändert. Der aktualisierte Einblick sucht nach Ergebnissen mit einem hohen Schweregrad, die sich auf AWS Rollen beziehen.

```
aws securityhub update-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \  
  --severity HIGH
```

```
--filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
"SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' \
--name "High severity role findings"
```

Beispiel 2: So ändern Sie das Gruppierungsattribut für einen benutzerdefinierten Einblick

Im folgenden `update-insight` Beispiel wird das Gruppierungsattribut für den benutzerdefinierten Einblick mit dem angegebenen Wert geändert. ARN Das neue Gruppierungsattribut ist die Ressourcen-ID.

```
aws securityhub update-insight \
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
  --group-by-attribute "ResourceId" \
  --name "Critical role findings"
```

Ausgabe:

```
{
  "Insights": [
    {
      "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
      "Name": "Critical role findings",
      "Filters": {
        "SeverityLabel": [
          {
            "Value": "CRITICAL",
            "Comparison": "EQUALS"
          }
        ],
        "ResourceType": [
          {
            "Value": "AwsIamRole",
            "Comparison": "EQUALS"
          }
        ]
      },
      "GroupByAttribute": "ResourceId"
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Managing Custom Insights](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateInsight](#) in der AWS CLI Befehlsreferenz.

update-organization-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-organization-configuration`.

AWS CLI

Um zu aktualisieren, wie Security Hub für eine Organisation konfiguriert ist

Das folgende `update-organization-configuration` Beispiel legt fest, dass Security Hub die zentrale Konfiguration verwenden sollte, um eine Organisation zu konfigurieren. Nach der Ausführung dieses Befehls kann der delegierte Security Hub-Administrator Konfigurationsrichtlinien zur Konfiguration der Organisation erstellen und verwalten. Der delegierte Administrator kann diesen Befehl auch verwenden, um von der zentralen zur lokalen Konfiguration zu wechseln. Wenn die lokale Konfiguration der Konfigurationstyp ist, kann der delegierte Administrator wählen, ob Security Hub und Standardsicherheitsstandards in neuen Organisationskonten automatisch aktiviert werden sollen.

```
aws securityhub update-organization-configuration \  
  --no-auto-enable \  
  --organization-configuration '{"ConfigurationType": "CENTRAL"}
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konten bei AWS Organizations verwalten](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateOrganizationConfiguration](#) in der AWS CLI Befehlsreferenz.

update-security-control

Das folgende Codebeispiel zeigt die Verwendung `update-security-control`.

AWS CLI

Um die Eigenschaften der Sicherheitskontrolle zu aktualisieren

Im folgenden `update-security-control` Beispiel werden benutzerdefinierte Werte für einen Security Hub-Sicherheitskontrollparameter angegeben.

```
aws securityhub update-security-control \  
  --security-control-id ACM.1 \  
  --parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":  
15}}}' \  
  --last-update-reason "Internal compliance requirement"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerdefinierte Steuerungsparameter](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSecurityControl](#) in der AWS CLI Befehlsreferenz.

update-security-hub-configuration

Das folgende Codebeispiel zeigt die Verwendung `update-security-hub-configuration`.

AWS CLI

So aktualisieren Sie die Security Hub Hub-Konfiguration

Im folgenden `update-security-hub-configuration` Beispiel wird Security Hub so konfiguriert, dass neue Steuerelemente für aktivierte Standards automatisch aktiviert werden.

```
aws securityhub update-security-hub-configuration \  
  --auto-enable-controls
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Automatisches Aktivieren neuer Steuerelemente](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSecurityHubConfiguration](#) in der AWS CLI Befehlsreferenz.

update-standards-control

Das folgende Codebeispiel zeigt die Verwendung `update-standards-control`.

AWS CLI

Beispiel 1: Um ein Steuerelement zu deaktivieren

Das folgende `update-standards-control` Beispiel deaktiviert die PCI. AutoScaling1. Steuerung.

```
aws securityhub update-standards-control \  
  --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-  
dss/v/3.2.1/PCI.AutoScaling.1" \  
  --control-status "DISABLED" \  
  --disabled-reason "Not applicable for my service"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um ein Steuerelement zu aktivieren

Das folgende `update-standards-control` Beispiel aktiviert die PCI. AutoScaling1. Steuerung.

```
aws securityhub update-standards-control \  
  --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-  
dss/v/3.2.1/PCI.AutoScaling.1" \  
  --control-status "ENABLED"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deaktivieren und Aktivieren einzelner Steuerungen](#) im AWS Security Hub Hub-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateStandardsControl](#) in der AWS CLI Befehlsreferenz.

Security Lake-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Security Lake Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-aws-logsource

Das folgende Codebeispiel zeigt die Verwendung `create-aws-logsource`.

AWS CLI

Um einen nativ unterstützten Amazon Web Service als Amazon Security Lake-Quelle hinzuzufügen

Im folgenden `create-aws-logsource` Beispiel werden VPC Flow Logs als Security Lake-Quelle in den angegebenen Konten und Regionen hinzugefügt.

```
aws securitylake create-aws-log-source \
  --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
  "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

Ausgabe:

```
{
  "failed": [
    "123456789012"
  ]
}
```

Weitere Informationen finden Sie unter [Hinzufügen eines AWS Service als Quelle](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateAwsLogsource](#) unter AWS CLI Befehlsreferenz.

create-custom-logsource

Das folgende Codebeispiel zeigt die Verwendung `create-custom-logsource`.

AWS CLI

So fügen Sie eine benutzerdefinierte Quelle als Amazon Security Lake-Quelle hinzu

Das folgende `create-custom-logsource` Beispiel fügt dem angegebenen Protokollanbieter-Konto und der angegebenen Region eine benutzerdefinierte Quelle als Security Lake-Quelle hinzu.

```
aws securitylake create-custom-log-source \
  --source-name "VPC_FLOW" \
  --event-classes ['"DNS_ACTIVITY", "NETWORK_ACTIVITY"] \
  --configuration '{"crawlerConfiguration": {"roleArn": "arn:aws:glue:eu-west-2:123456789012:crawler/E1WG1ZNPRT0D4"},"providerIdentity": {"principal": "029189416600", "externalId": "123456789012"}}' --region "us-east-1"
```

Ausgabe:

```
{
  "customLogSource": {
    "attributes": {
      "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/E1WG1ZNPRT0D4",
      "databaseArn": "arn:aws:glue:eu-west-2:123456789012:database/E1WG1ZNPRT0D4",
      "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/E1WG1ZNPRT0D4"
    },
    "provider": {
      "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
      "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-Provider-testCustom2-eu-west-2"
    },
    "sourceName": "testCustom2"
    "sourceVersion": "2.0"
  }
}
```

Weitere Informationen finden Sie unter [Hinzufügen einer benutzerdefinierten Quelle](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCustomLogsource](#) unter AWS CLI Befehlsreferenz.

create-data-lake-exception-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-data-lake-exception-subscription`.

AWS CLI

Um Benachrichtigungen über Security Lake-Ausnahmen zu senden

Im folgenden `create-data-lake-exception-subscription` Beispiel werden Benachrichtigungen über Security Lake-Ausnahmen per SMS Zustellung an das angegebene Konto gesendet. Die Ausnahmemeldung bleibt für den angegebenen Zeitraum bestehen.

```
aws securitylake create-data-lake-exception-subscription \  
  --notification-endpoint "123456789012" \  
  --exception-time-to-live 30 \  
  --subscription-protocol "sms"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Problembehandlung bei Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDataLakeExceptionSubscription](#) in der AWS CLI Befehlsreferenz.

`create-data-lake-organization-configuration`

Das folgende Codebeispiel zeigt die Verwendung `create-data-lake-organization-configuration`.

AWS CLI

Um Security Lake in neuen Organisationskonten zu konfigurieren

Das folgende `create-data-lake-organization-configuration` Beispiel aktiviert Security Lake und die Erfassung der angegebenen Quellereignisse und Protokolle in neuen Organisationskonten.

```
aws securitylake create-data-lake-organization-configuration \  
  --auto-enable-new-account '[{"region": "us-east-1", "sources":  
  [{"sourceName": "SH_FINDINGS", "sourceVersion": "1.0"}]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung mehrerer Konten bei AWS Organizations](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDataLakeOrganizationConfiguration](#) unter AWS CLI Befehlsreferenz.

create-data-lake

Das folgende Codebeispiel zeigt die Verwendung `create-data-lake`.

AWS CLI

Beispiel 1: So konfigurieren Sie Ihren Data Lake in mehreren Regionen

Das folgende `create-data-lake` Beispiel aktiviert Amazon Security Lake in mehreren AWS Regionen und konfiguriert Ihren Data Lake.

```
aws securitylake create-data-lake \
  --configurations '[{"encryptionConfiguration":
{"kmsKeyId":"S3_MANAGED_KEY","region":"us-east-1","lifecycleConfiguration":
{"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
{"kmsKeyId":"S3_MANAGED_KEY","region":"us-east-2","lifecycleConfiguration":
{"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

Ausgabe:

```
{
  "dataLakes": [
    {
      "createStatus": "COMPLETED",
      "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
      "encryptionConfiguration": {
        "kmsKeyId": "S3_MANAGED_KEY"
      },
      "lifecycleConfiguration": {
        "expiration": {
          "days": 365
        },
        "transitions": [
          {
```

```

        "days": 60,
        "storageClass": "ONEZONE_IA"
    }
  ]
},
"region": "us-east-1",
"replicationConfiguration": {
  "regions": [
    "ap-northeast-3"
  ],
  "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
},
"s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
"updateStatus": {
  "exception": {},
  "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
  "status": "INITIALIZED"
}
},
{
  "createStatus": "COMPLETED",
  "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
  "encryptionConfiguration": {
    "kmsKeyId": "S3_MANAGED_KEY"
  },
  "lifecycleConfiguration": {
    "expiration": {
      "days": 365
    },
    "transitions": [
      {
        "days": 60,
        "storageClass": "ONEZONE_IA"
      }
    ]
  },
  "region": "us-east-2",
  "replicationConfiguration": {
    "regions": [
      "ap-northeast-3"
    ],

```

```

        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
      },
      "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-east-2-
cehuiifzl5rwmhm6m62h7zhvtseogr9",
      "updateStatus": {
        "exception": {},
        "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
        "status": "INITIALIZED"
      }
    }
  ]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

Beispiel 2: So konfigurieren Sie Ihren Data Lake in einer einzelnen Region

Das folgende `create-data-lake` Beispiel aktiviert Amazon Security Lake in einer einzelnen AWS Region und konfiguriert Ihren Data Lake.

```

aws securitylake create-data-lake \
  --configurations '[{"encryptionConfiguration":
{"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}]' \
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"

```

Ausgabe:

```

{
  "dataLakes": [
    {
      "createStatus": "COMPLETED",
      "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
      "encryptionConfiguration": {
        "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      "lifecycleConfiguration": {

```

```

        "expiration": {
            "days": 500
        },
        "transitions": [
            {
                "days": 30,
                "storageClass": "GLACIER"
            }
        ]
    },
    "region": "us-east-2",
    "replicationConfiguration": {
        "regions": [
            "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
    "updateStatus": {
        "exception": {},
        "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
        "status": "INITIALIZED"
    }
}
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDataLake](#) unter AWS CLI Befehlsreferenz.

create-subscriber-data-access

Das folgende Codebeispiel zeigt die Verwendung `create-subscriber-data-access`.

AWS CLI

Um einen Abonnenten mit Datenzugriff zu erstellen

Im folgenden `create-subscriber` Beispiel wird ein Abonnent in Security Lake mit Zugriff auf Daten in der aktuellen AWS Region für die angegebene Abonnenten-Identität für eine AWS Quelle erstellt.

```
aws securitylake create-subscriber \  
  --access-types "S3" \  
  --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW", "sourceVersion":  
"2.0"}}]' \  
  --subscriber-name "opensearch-s3" \  
  --subscriber-identity '{"principal": "029189416600", "externalId":  
"123456789012"}'
```

Ausgabe:

```
{  
  "subscriber": {  
    "accessTypes": [  
      "S3"  
    ],  
    "createdAt": "2024-07-17T19:08:26.787000+00:00",  
    "roleArn": "arn:aws:iam::773172568199:role/AmazonSecurityLake-896f218b-  
cfba-40be-a255-8b49a65d0407",  
    "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-  
um632ufwpvxkyz0bc5hkb64atycnf3",  
    "sources": [  
      {  
        "awsLogSource": {  
          "sourceName": "VPC_FLOW",  
          "sourceVersion": "2.0"  
        }  
      }  
    ],  
    "subscriberArn": "arn:aws:securitylake:us-  
east-1:773172568199:subscriber/896f218b-cfba-40be-a255-8b49a65d0407",  
    "subscriberId": "896f218b-cfba-40be-a255-8b49a65d0407",  
    "subscriberIdentity": {  
      "externalId": "123456789012",  
      "principal": "029189416600"  
    },  
    "subscriberName": "opensearch-s3",  
    "subscriberStatus": "ACTIVE",  
    "updatedAt": "2024-07-17T19:08:27.133000+00:00"  
  }  
}
```

```
}
```

Weitere Informationen finden Sie unter [Einen Abonnenten mit Datenzugriff](#) erstellen im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSubscriberDataAccess](#) unter AWS CLI Befehlsreferenz.

create-subscriber-notification

Das folgende Codebeispiel zeigt die Verwendung `create-subscriber-notification`.

AWS CLI

Um eine Abonnentenbenachrichtigung zu erstellen

Das folgende `create-subscriber-notification` Beispiel zeigt, wie Sie eine Abonnentenbenachrichtigung angeben, um eine Benachrichtigung zu erstellen, wenn neue Daten in den Data Lake geschrieben werden.

```
aws securitylake create-subscriber-notification \
  --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
  --configuration '{"httpsNotificationConfiguration":
{"targetRoleArn": "arn:aws:iam::XXX:role/service-role/RoleName",
"endpoint": "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

Ausgabe:

```
{
  "subscriberEndpoint": [
    "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
  ]
}
```

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSubscriberNotification](#) in der AWS CLI Befehlsreferenz.

create-subscriber-query-access

Das folgende Codebeispiel zeigt die Verwendung `create-subscriber-query-access`.

AWS CLI

Um einen Abonnenten mit Abfragezugriff zu erstellen

Im folgenden `create-subscriber` Beispiel wird ein Abonnent in Security Lake mit Abfragezugriff in der aktuellen AWS Region für die angegebene Abonnenten-Identität erstellt.

```
aws securitylake create-subscriber \
  --access-types "LAKEFORMATION" \
  --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW", "sourceVersion":
"2.0"}}]' \
  --subscriber-name "opensearch-s3" \
  --subscriber-identity '{"principal": "029189416600", "externalId":
"123456789012}"'
```

Ausgabe:

```
{
  "subscriber": {
    "accessTypes": [
      "LAKEFORMATION"
    ],
    "createdAt": "2024-07-18T01:05:55.853000+00:00",
    "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/8c31da49-c224-4f1e-bb12-37ab756d6d8a",
    "resourceShareName": "LakeFormation-V2-NAMENAMENA-123456789012",
    "sources": [
      {
        "awsLogSource": {
          "sourceName": "VPC_FLOW",
          "sourceVersion": "2.0"
        }
      }
    ],
    "subscriberArn": "arn:aws:securitylake:us-east-1:123456789012:subscriber/
e762aabb-ce3d-4585-beab-63474597845d",
    "subscriberId": "e762aabb-ce3d-4585-beab-63474597845d",
    "subscriberIdentity": {
      "externalId": "123456789012",
      "principal": "029189416600"
    },
    "subscriberName": "opensearch-s3",
    "subscriberStatus": "ACTIVE",
```



```
    "updatedAt": "2024-07-18T01:05:58.393000+00:00"  
  }  
}
```

Weitere Informationen finden Sie unter [Einen Abonnenten mit Abfragezugriff erstellen](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateSubscriberQueryAccess](#) unter AWS CLI Befehlsreferenz.

delete-aws-logsource

Das folgende Codebeispiel zeigt die Verwendung `delete-aws-logsource`.

AWS CLI

Um einen nativ unterstützten Dienst zu entfernen. AWS

Im folgenden `delete-aws-logsource` Beispiel werden VPC Flow Logs als Security Lake-Quelle in den angegebenen Konten und Regionen gelöscht.

```
aws securitylake delete-aws-log-source \  
  --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],  
  "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

Ausgabe:

```
{  
  "failed": [  
    "123456789012"  
  ]  
}
```

Weitere Informationen finden Sie unter [Entfernen eines AWS Dienstes als Quelle](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAwsLogsource](#) unter AWS CLI Befehlsreferenz.

delete-custom-logsource

Das folgende Codebeispiel zeigt die Verwendung `delete-custom-logsource`.

AWS CLI

Um eine benutzerdefinierte Quelle zu entfernen.

Im folgenden `delete-custom-log-source` Beispiel wird eine benutzerdefinierte Quelle im angegebenen Protokollanbieter-Konto in der angegebenen Region gelöscht.

```
aws securitylake delete-custom-log-source \  
  --source-name "CustomSourceName"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Löschen einer benutzerdefinierten Quelle](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteCustomLogsource](#) unter AWS CLI Befehlsreferenz.

`delete-data-lake-organization-configuration`

Das folgende Codebeispiel zeigt die Verwendung `delete-data-lake-organization-configuration`.

AWS CLI

Um die automatische Quellenerfassung in Mitgliedskonten zu beenden

Im folgenden `delete-data-lake-organization-configuration` Beispiel wird die automatische Erfassung von AWS Security Hub Hub-Ergebnissen von neuen Mitgliedskonten, die der Organisation beitreten, gestoppt. Nur der delegierte Security Lake-Administrator kann diesen Befehl ausführen. Er verhindert, dass neue Mitgliedskonten automatisch Daten zum Data Lake beitragen.

```
aws securitylake delete-data-lake-organization-configuration \  
  --auto-enable-new-account '[{"region": "us-east-1", "sources":  
  [{"sourceName": "SH_FINDINGS"}]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung mehrerer Konten bei AWS Organizations](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDataLakeOrganizationConfiguration](#) unter AWS CLI Befehlsreferenz.

delete-data-lake

Das folgende Codebeispiel zeigt die Verwendung `delete-data-lake`.

AWS CLI

Um Ihren Data Lake zu deaktivieren

Im folgenden `delete-data-lake` Beispiel wird Ihr Data Lake in den angegebenen AWS Regionen deaktiviert. In den angegebenen Regionen tragen Quellen keine Daten mehr zum Data Lake bei. Bei einer Security Lake-Bereitstellung, die AWS Organizations verwendet, kann nur der delegierte Security Lake-Administrator für die Organisation Security Lake für Konten in der Organisation deaktivieren.

```
aws securitylake delete-data-lake \  
  --regions "ap-northeast-1" "eu-central-1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deaktivierung von Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteDataLake AWS CLI](#) Befehlsreferenz.

delete-subscriber-notification

Das folgende Codebeispiel zeigt die Verwendung `delete-subscriber-notification`.

AWS CLI

Um eine Abonnentenbenachrichtigung zu löschen

Das folgende `delete-subscriber-notification` Beispiel zeigt, wie die Abonnentenbenachrichtigung für einen bestimmten Security Lake-Abonnenten gelöscht wird.

```
aws securitylake delete-subscriber-notification \  
  --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSubscriberNotification](#) in der AWS CLI Befehlsreferenz.

delete-subscriber

Das folgende Codebeispiel zeigt die Verwendung `delete-subscriber`.

AWS CLI

Um einen Abonnenten zu löschen

Das folgende `delete-subscriber` Beispiel zeigt, wie Sie einen Abonnenten entfernen, wenn Sie nicht mehr möchten, dass ein Abonnent Daten aus Security Lake nutzt.

```
aws securitylake delete-subscriber \  
  --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteSubscriber](#) in der AWS CLI Befehlsreferenz.

get-data-lake-exception-subscription

Das folgende Codebeispiel zeigt die Verwendung `get-data-lake-exception-subscription`.

AWS CLI

Um Details zu einem Ausnahmeabonnement zu erhalten

Das folgende `get-data-lake-exception-subscription` Beispiel enthält Details zu einem Security Lake-Ausnahmeabonnement. In diesem Beispiel wird der Benutzer des angegebenen AWS Kontos bei der SMS Zustellung über Fehler informiert. Die Ausnahmemeldung verbleibt für den angegebenen Zeitraum im Konto. Ein Ausnahmeabonnement benachrichtigt einen Security Lake-Benutzer über das bevorzugte Protokoll des Antragstellers über einen Fehler.

```
aws securitylake get-data-lake-exception-subscription
```

Ausgabe:

```
{
  "exceptionTimeToLive": 30,
  "notificationEndpoint": "123456789012",
  "subscriptionProtocol": "sms"
}
```

Weitere Informationen finden Sie unter [Fehlerbehebung beim Data Lake-Status](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [GetDataLakeExceptionSubscription](#) in der AWS CLI Befehlsreferenz.

get-data-lake-organization-configuration

Das folgende Codebeispiel zeigt die Verwendung `get-data-lake-organization-configuration`.

AWS CLI

Um Details zur Konfiguration für neue Organisationskonten zu erhalten

Im folgenden `get-data-lake-organization-configuration` Beispiel werden Details zu den Quellprotokollen abgerufen, die neue Organisationskonten nach dem Onboarding in Amazon Security Lake senden.

```
aws securitylake get-data-lake-organization-configuration
```

Ausgabe:

```
{
  "autoEnableNewAccount": [
    {
      "region": "us-east-1",
      "sources": [
        {
          "sourceName": "VPC_FLOW",
          "sourceVersion": "1.0"
        }
      ]
    }
  ]
}
```

```
    {
      "sourceName": "ROUTE53",
      "sourceVersion": "1.0"
    },
    {
      "sourceName": "SH_FINDINGS",
      "sourceVersion": "1.0"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwaltung mehrerer Konten bei AWS Organizations](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [GetDataLakeOrganizationConfiguration](#) unter AWS CLI Befehlsreferenz.

get-data-lake-sources

Das folgende Codebeispiel zeigt die Verwendung `get-data-lake-sources`.

AWS CLI

Um den Status der Protokollsammlung abzurufen

Im folgenden `get-data-lake-sources` Beispiel wird eine Momentaufnahme der Protokollsammlung für das angegebene Konto in der aktuellen AWS Region abgerufen. Für das Konto ist Amazon Security Lake aktiviert.

```
aws securitylake get-data-lake-sources \
  --accounts "123456789012"
```

Ausgabe:

```
{
  "dataLakeSources": [
    {
      "account": "123456789012",
      "sourceName": "SH_FINDINGS",
      "sourceStatuses": [
```

```
        {
            "resource": "vpc-1234567890abcdef0",
            "status": "COLLECTING"
        }
    ]
},
{
    "account": "123456789012",
    "sourceName": "VPC_FLOW",
    "sourceStatuses": [
        {
            "resource": "vpc-1234567890abcdef0",
            "status": "NOT_COLLECTING"
        }
    ]
},
{
    "account": "123456789012",
    "sourceName": "LAMBDA_EXECUTION",
    "sourceStatuses": [
        {
            "resource": "vpc-1234567890abcdef0",
            "status": "COLLECTING"
        }
    ]
},
{
    "account": "123456789012",
    "sourceName": "ROUTE53",
    "sourceStatuses": [
        {
            "resource": "vpc-1234567890abcdef0",
            "status": "COLLECTING"
        }
    ]
},
{
    "account": "123456789012",
    "sourceName": "CLOUD_TRAIL_MGMT",
    "sourceStatuses": [
        {
            "resource": "vpc-1234567890abcdef0",
            "status": "COLLECTING"
        }
    ]
}
```

```
    ]
  }
],
"dataLakeArn": null
}
```

Weitere Informationen finden Sie unter [Sammeln von Daten aus AWS Diensten](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [GetDataLakeSources](#) unter AWS CLI Befehlsreferenz.

get-subscriber

Das folgende Codebeispiel zeigt die Verwendung `get-subscriber`.

AWS CLI

Um die Abonnementinformationen abzurufen

Im folgenden `get-subscriber` Beispiel werden die Abonnementinformationen für den angegebenen Security Lake-Abonnenten abgerufen.

```
aws securitylake get-subscriber \
  --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "subscriber": {
    "accessTypes": [
      "LAKEFORMATION"
    ],
    "createdAt": "2024-04-19T15:19:44.421803+00:00",
    "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
    "sources": [
      {
        "awsLogSource": {
          "sourceName": "LAMBDA_EXECUTION",
          "sourceVersion": "1.0"
        }
      }
    ],
  },
}
```



```
{
  "awsLogSource": {
    "sourceName": "EKS_AUDIT",
    "sourceVersion": "2.0"
  }
},
{
  "awsLogSource": {
    "sourceName": "ROUTE53",
    "sourceVersion": "1.0"
  }
},
{
  "awsLogSource": {
    "sourceName": "SH_FINDINGS",
    "sourceVersion": "1.0"
  }
},
{
  "awsLogSource": {
    "sourceName": "VPC_FLOW",
    "sourceVersion": "1.0"
  }
},
{
  "customLogSource": {
    "attributes": {
      "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
testCustom2",
      "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
      "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom2"
    },
    "provider": {
      "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfjblwdwbgf4vge98av9/ext/testCustom2/",
      "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-testCustom2-eu-west-2"
    },
    "sourceName": "testCustom2"
  }
},
{
```

```

        "customLogSource": {
            "attributes": {
                "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
TestCustom",
                "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom"
            },
            "provider": {
                "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjblwdwbgf4vge98av9/ext/TestCustom/",
                "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-TestCustom-eu-west-2"
            },
            "sourceName": "TestCustom"
        }
    ],
    "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "subscriberIdentity": {
        "externalId": "123456789012",
        "principal": "123456789012"
    },
    "subscriberName": "test",
    "subscriberStatus": "ACTIVE",
    "updatedAt": "2024-04-19T15:19:55.230588+00:00"
}
}

```

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [GetSubscriber](#) in der AWS CLI Befehlsreferenz.

list-data-lake-exceptions

Das folgende Codebeispiel zeigt die Verwendung `list-data-lake-exceptions`.

AWS CLI

Um die Probleme aufzulisten, die Ihren Data Lake betreffen

Im folgenden `list-data-lake-exceptions` Beispiel werden die Probleme aufgeführt, die sich in den letzten 14 Tagen in den angegebenen AWS Regionen auf Ihren Data Lake ausgewirkt haben.

```
aws securitylake list-data-lake-exceptions \  
  --regions "us-east-1" "eu-west-3"
```

Ausgabe:

```
{  
  "exceptions": [  
    {  
      "exception": "The account does not have the required role permissions.  
Update your role permissions to use the new data source version.",  
      "region": "us-east-1",  
      "timestamp": "2024-02-29T12:24:15.641725+00:00"  
    },  
    {  
      "exception": "The account does not have the required role permissions.  
Update your role permissions to use the new data source version.",  
      "region": "eu-west-3",  
      "timestamp": "2024-02-29T12:24:15.641725+00:00"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Problembehandlung bei Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [ListDataLakeExceptions](#) in der AWS CLI Befehlsreferenz.

list-data-lakes

Das folgende Codebeispiel zeigt die Verwendung `list-data-lakes`.

AWS CLI

Um das Security Lake-Konfigurationsobjekt aufzulisten

Das folgende `list-data-lakes` Beispiel listet das Amazon Security Lake-Konfigurationsobjekt für die angegebene AWS Region auf. Sie können diesen Befehl verwenden, um festzustellen, ob Security Lake in einer oder mehreren bestimmten Regionen aktiviert ist.

```
aws securitylake list-data-lakes \  
  --regions "us-east-1"
```

Ausgabe:

```
{  
  "dataLakes": [  
    {  
      "createStatus": "COMPLETED",  
      "dataLakeArn": "arn:aws:securitylake:us-east-1:123456789012:data-lake/  
default",  
      "encryptionConfiguration": {  
        "kmsKeyId": "S3_MANAGED_KEY"  
      },  
      "lifecycleConfiguration": {  
        "expiration": {  
          "days": 365  
        },  
        "transitions": [  
          {  
            "days": 60,  
            "storageClass": "ONEZONE_IA"  
          }  
        ]  
      },  
      "region": "us-east-1",  
      "replicationConfiguration": {  
        "regions": [  
          "ap-northeast-3"  
        ],  
        "roleArn": "arn:aws:securitylake:ap-northeast-3:123456789012:data-  
lake/default"  
      },  
      "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-  
east-1-1234567890abcdef0",  
      "updateStatus": {  
        "exception": {  
          "code": "software.amazon.awssdk.services.s3.model.S3Exception",  
          "reason": ""  
        }  
      }  
    }  
  ]  
}
```

```
    },
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "status": "FAILED"
  }
}
]
```

Weitere Informationen finden Sie unter [Überprüfen des Regionsstatus](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [ListDataLakes](#) unter AWS CLI Befehlsreferenz.

list-log-sources

Das folgende Codebeispiel zeigt die Verwendung `list-log-sources`.

AWS CLI

Um die Amazon Security Lake-Protokollquellen abzurufen

Das folgende `list-log-sources` Beispiel listet die Amazon Security Lake-Protokollquellen in einem angegebenen Konto auf.

```
aws securitylake list-log-sources \
  --accounts "123456789012"
```

Ausgabe:

```
{
  "account": "123456789012",
  "region": "xy-region-1",
  "sources": [
    {
      "awsLogSource": {
        "sourceName": "VPC_FLOW",
        "sourceVersion": "2.0"
      }
    },
    {
      "awsLogSource": {
        "sourceName": "SH_FINDINGS",
        "sourceVersion": "2.0"
      }
    }
  ]
}
```

```
    }
  }
]
}
```

Weitere Informationen finden Sie unter [Quellenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [ListLogSources](#) in der AWS CLI Befehlsreferenz.

list-subscribers

Das folgende Codebeispiel zeigt die Verwendung `list-subscribers`.

AWS CLI

Um die Amazon Security Lake-Abonnenten abzurufen

Das folgende `list-subscribers` Beispiel listet alle Amazon Security Lake-Abonnenten in einem bestimmten Konto auf.

```
aws securitylake list-subscribers
```

Ausgabe:

```
{
  "subscribers": [
    {
      "accessTypes": [
        "S3"
      ],
      "createdAt": "2024-06-04T15:02:28.921000+00:00",
      "roleArn": "arn:aws:iam:123456789012:role/AmazonSecurityLake-
E1WG1ZNPRXT0D4",
      "s3BucketArn": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
      "sources": [
        {
          "awsLogSource": {
            "sourceName": "CLOUD_TRAIL_MGMT",
            "sourceVersion": "2.0"
          }
        }
      ],
    }
  ]
}
```

```

        "awsLogSource": {
            "sourceName": "LAMBDA_EXECUTION",
            "sourceVersion": "1.0"
        }
    },
    {
        "customLogSource": {
            "attributes": {
                "crawlerArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4",
                "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
            },
            "provider": {
                "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
                "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
            },
            "sourceName": "testCustom2"
        }
    }
],
    "subscriberArn": "arn:aws:securitylake:eu-
west-2:123456789012:subscriber/E1WG1ZNPRXT0D4",
    "subscriberEndpoint": "arn:aws:sqs:eu-
west-2:123456789012:AmazonSecurityLake-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-Main-
Queue",
    "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "subscriberIdentity": {
        "externalId": "ext123456789012",
        "principal": "123456789012"
    },
    "subscriberName": "Test",
    "subscriberStatus": "ACTIVE",
    "updatedAt": "2024-06-04T15:02:35.617000+00:00"
}
]
}

```

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [ListSubscribers](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für eine vorhandene Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet Tags für den angegebenen Amazon Security Lake-Abonnenten auf. In diesem Beispiel ist dem Tagschlüssel `Owner` kein Tag-Wert zugeordnet. Mit diesem Vorgang können Sie auch Tags für andere vorhandene Security Lake-Ressourcen auflisten.

```
aws securitylake list-tags-for-resource \
  --resource-arn "arn:aws:securitylake:us-
  east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab"
```

Ausgabe:

```
{
  "tags": [
    {
      "key": "Environment",
      "value": "Cloud"
    },
    {
      "key": "CostCenter",
      "value": "12345"
    },
    {
      "key": "Owner",
      "value": ""
    }
  ]
}
```

Weitere Informationen finden Sie unter [Tagging Amazon Security Lake-Ressourcen](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListTagsForResource AWS CLI](#) Befehlsreferenz.

register-data-lake-delegated-administrator

Das folgende Codebeispiel zeigt die Verwendung `register-data-lake-delegated-administrator`.

AWS CLI

Um die delegierten Administratoren zu bestimmen

Im folgenden `register-data-lake-delegated-administrator` Beispiel wird das angegebene AWS Konto als delegierter Amazon Security Lake-Administrator bezeichnet.

```
aws securitylake register-data-lake-delegated-administrator \  
  --account-id 123456789012
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwaltung mehrerer Konten bei AWS Organizations](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterDataLakeDelegatedAdministrator](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um einer vorhandenen Ressource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel werden einer vorhandenen Abonnentenressource Tags hinzugefügt. Verwenden Sie diesen Vorgang nicht, um eine neue Ressource zu erstellen und ihr ein oder mehrere Tags hinzuzufügen. Verwenden Sie stattdessen den entsprechenden Erstellungsvorgang für den Ressourcentyp, den Sie erstellen möchten.

```
aws securitylake tag-resource \  
  --resource-arn "arn:aws:securitylake:us-east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags key=Environment,value=Cloud
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Amazon Security Lake-Ressourcen](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie unter [TagResource AWS CLI](#) Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um Tags aus einer vorhandenen Ressource zu entfernen

Im folgenden `untag-resource` Beispiel werden die angegebenen Tags aus einer vorhandenen Abonnementressource entfernt.

```
aws securitylake untag-resource \  
  --resource-arn "arn:aws:securitylake:us-  
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags Environment Owner
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Amazon Security Lake-Ressourcen](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie unter [UntagResource AWS CLI](#) Befehlsreferenz.

update-data-lake-exception-subscription

Das folgende Codebeispiel zeigt die Verwendung `update-data-lake-exception-subscription`.

AWS CLI

Um das Benachrichtigungsabonnement für Security Lake-Ausnahmen zu aktualisieren

Im folgenden `update-data-lake-exception-subscription` Beispiel wird das Benachrichtigungsabonnement aktualisiert, das Benutzer über Security Lake-Ausnahmen informiert.

```
aws securitylake update-data-lake-exception-subscription \  
  --resource-arn "arn:aws:securitylake:us-east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags Environment Owner
```

```
--notification-endpoint "123456789012" \  
--exception-time-to-live 30 \  
--subscription-protocol "email"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Problembehandlung bei Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDataLakeExceptionSubscription](#) in der AWS CLI Befehlsreferenz.

update-data-lake

Das folgende Codebeispiel zeigt die Verwendung `update-data-lake`.

AWS CLI

Beispiel 1: Um Ihre Data Lake-Einstellungen zu aktualisieren

Das folgende `update-data-lake` Beispiel aktualisiert die Einstellungen Ihres Amazon Security Lake Data Lakes. Mit diesem Vorgang können Sie die Regionseinstellungen für Datenverschlüsselung, Speicherung und Rollup angeben.

```
aws securitylake update-data-lake \  
  --configurations '[{"encryptionConfiguration":  
  {"kmsKeyId": "S3_MANAGED_KEY", "region": "us-east-1", "lifecycleConfiguration":  
  {"expiration": {"days": 365}, "transitions":  
  [{"days": 60, "storageClass": "ONEZONE_IA"}]}}, {"encryptionConfiguration":  
  {"kmsKeyId": "S3_MANAGED_KEY", "region": "us-east-2", "lifecycleConfiguration":  
  {"expiration": {"days": 365}, "transitions":  
  [{"days": 60, "storageClass": "ONEZONE_IA"}]}}]' \  
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-  
  role/AmazonSecurityLakeMetaStoreManager"
```

Ausgabe:

```
{  
  "dataLakes": [  
    {  
      "createStatus": "COMPLETED",
```

```
    "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
    "encryptionConfiguration": {
      "kmsKeyId": "S3_MANAGED_KEY"
    },
    "lifecycleConfiguration": {
      "expiration": {
        "days": 365
      },
      "transitions": [
        {
          "days": 60,
          "storageClass": "ONEZONE_IA"
        }
      ]
    },
    "region": "us-east-1",
    "replicationConfiguration": {
      "regions": [
        "ap-northeast-3"
      ],
      "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
    "updateStatus": {
      "exception": {},
      "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
      "status": "INITIALIZED"
    }
  },
  {
    "createStatus": "COMPLETED",
    "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
    "encryptionConfiguration": {
      "kmsKeyId": "S3_MANAGED_KEY"
    },
    "lifecycleConfiguration": {
      "expiration": {
        "days": 365
      },
      "transitions": [
```

```

        {
            "days": 60,
            "storageClass": "ONEZONE_IA"
        }
    ],
    "region": "us-east-2",
    "replicationConfiguration": {
        "regions": [
            "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
    "updateStatus": {
        "exception": {},
        "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
        "status": "INITIALIZED"
    }
}
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

Beispiel 2: So konfigurieren Sie Ihren Data Lake in einer einzelnen Region

Das folgende `create-data-lake` Beispiel aktiviert Amazon Security Lake in einer einzelnen AWS Region und konfiguriert Ihren Data Lake.

```

aws securitylake create-data-lake \
  --configurations '[{"encryptionConfiguration":
  {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}]' \
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"

```

Ausgabe:

```
{
  "dataLakes": [
    {
      "createStatus": "COMPLETED",
      "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
      "encryptionConfiguration": {
        "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      "lifecycleConfiguration": {
        "expiration": {
          "days": 500
        },
        "transitions": [
          {
            "days": 30,
            "storageClass": "GLACIER"
          }
        ]
      },
      "region": "us-east-2",
      "replicationConfiguration": {
        "regions": [
          "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
      },
      "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
      "updateStatus": {
        "exception": {},
        "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
        "status": "INITIALIZED"
      }
    }
  ]
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon Security Lake](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDataLake](#) unter AWS CLI Befehlsreferenz.

update-subscriber-notification

Das folgende Codebeispiel zeigt die Verwendung `update-subscriber-notification`.

AWS CLI

Um eine Abonnentenbenachrichtigung zu aktualisieren

Das folgende `update-subscriber-notification` Beispiel zeigt, wie Sie die Benachrichtigungsmethode für einen Abonnenten aktualisieren können.

```
aws securitylake update-subscriber-notification \  
  --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \  
  --configuration '{"httpsNotificationConfiguration":  
{"targetRoleArn": "arn:aws:iam::XXX:role/service-role/RoleName",  
"endpoint": "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

Ausgabe:

```
{  
  "subscriberEndpoint": [  
    "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"  
  ]  
}
```

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSubscriberNotification](#) in der AWS CLI Befehlsreferenz.

update-subscriber

Das folgende Codebeispiel zeigt die Verwendung `update-subscriber`.

AWS CLI

Um einen Amazon Security Lake-Abonnenten zu aktualisieren.

Im folgenden `update-subscriber` Beispiel werden die Security Lake-Datenzugriffsquellen für einen bestimmten Security Lake-Abonnenten aktualisiert.

```
aws securitylake update-subscriber \  

```

```
--subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "subscriber": {
    "accessTypes": [
      "LAKEFORMATION"
    ],
    "createdAt": "2024-04-19T15:19:44.421803+00:00",
    "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
    "sources": [
      {
        "awsLogSource": {
          "sourceName": "LAMBDA_EXECUTION",
          "sourceVersion": "1.0"
        }
      },
      {
        "awsLogSource": {
          "sourceName": "EKS_AUDIT",
          "sourceVersion": "2.0"
        }
      },
      {
        "awsLogSource": {
          "sourceName": "ROUTE53",
          "sourceVersion": "1.0"
        }
      },
      {
        "awsLogSource": {
          "sourceName": "SH_FINDINGS",
          "sourceVersion": "1.0"
        }
      },
      {
        "awsLogSource": {
          "sourceName": "VPC_FLOW",
          "sourceVersion": "1.0"
        }
      }
    ]
  }
}
```



```

    },
    {
      "customLogSource": {
        "attributes": {
          "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRT0D4",
          "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRT0D4",
          "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRT0D4"
        },
        "provider": {
          "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
          "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRT0D4"
        },
        "sourceName": "testCustom2"
      }
    }
  ],
  "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "subscriberIdentity": {
    "externalId": "123456789012",
    "principal": "123456789012"
  },
  "subscriberName": "test",
  "subscriberStatus": "ACTIVE",
  "updatedAt": "2024-07-18T20:47:37.098000+00:00"
}
}

```

Weitere Informationen finden Sie unter [Abonnentenverwaltung](#) im Amazon Security Lake-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateSubscriber](#) in der AWS CLI Befehlsreferenz.

AWS Serverless Application Repository Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Serverless Application Repository.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

put-application-policy

Das folgende Codebeispiel zeigt die Verwendung `put-application-policy`.

AWS CLI

Beispiel 1: Um eine Anwendung öffentlich zu teilen

Im Folgenden `put-application-policy` wird eine Anwendung öffentlich freigegeben, sodass jeder Ihre Anwendung im AWS Serverless Application Repository finden und bereitstellen kann.

```
aws serverlessrepo put-application-policy \  
  --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-  
test-application \  
  --statements Principals='*',Actions=Deploy
```

Ausgabe:

```
{  
  "Statements": [  
    {
```

```

    "Actions": [
      "Deploy"
    ],
    "Principals": [
      ""
    ],
    "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
]
}

```

Beispiel 2: Um eine Anwendung privat zu teilen

Im Folgenden `put-application-policy` wird eine Anwendung privat gemeinsam genutzt, sodass nur bestimmte AWS Konten Ihre Anwendung im AWS Serverless Application Repository finden und bereitstellen können.

```

aws serverlessrepo put-application-policy \
  --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
  --statements Principals=111111111111,222222222222,Actions=Deploy

```

Ausgabe:

```

{
  "Statements": [
    {
      "Actions": [
        "Deploy"
      ],
      "Principals": [
        "111111111111",
        "222222222222"
      ],
      "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
  ]
}

```

Weitere Informationen finden Sie unter [Freigeben einer Anwendung über die Konsole](#) im AWS Serverless Application Repository Developer Guide

- API-Einheiten finden Sie [PutApplicationPolicy](#) in der AWS CLI Befehlsreferenz.

Servicekatalog-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe des AWS Command Line Interface with Service Catalog Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

accept-portfolio-share

Das folgende Codebeispiel zeigt die Verwendung `accept-portfolio-share`.

AWS CLI

Um eine Portfolioaktie anzunehmen

Im folgenden `accept-portfolio-share` Beispiel wird ein Angebot eines anderen Benutzers zur gemeinsamen Nutzung des angegebenen Portfolios akzeptiert.

```
aws servicecatalog accept-portfolio-share \  
  --portfolio-id port-2s6wuabcdefghijk
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AcceptPortfolioShare](#) unter AWS CLI Befehlsreferenz.

associate-principal-with-portfolio

Das folgende Codebeispiel zeigt die Verwendung `associate-principal-with-portfolio`.

AWS CLI

Um einen Principal einem Portfolio zuzuordnen

Im folgenden `associate-principal-with-portfolio` Beispiel wird ein Benutzer dem angegebenen Portfolio zugeordnet.

```
aws servicecatalog associate-principal-with-portfolio \  
  --portfolio-id port-2s6abcdefwdh4 \  
  --principal-arn arn:aws:iam::123456789012:user/usertest \  
  --principal-type IAM
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AssociatePrincipalWithPortfolio](#) unter AWS CLI Befehlsreferenz.

`associate-product-with-portfolio`

Das folgende Codebeispiel zeigt die Verwendung `associate-product-with-portfolio`.

AWS CLI

Um ein Produkt einem Portfolio zuzuordnen

Im folgenden `associate-product-with-portfolio` Beispiel wird das angegebene Produkt dem angegebenen Portfolio zugeordnet.

```
aws servicecatalog associate-product-with-portfolio \  
  --product-id prod-3p5abcdef3oyk \  
  --portfolio-id port-2s6abcdef5wdh4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AssociateProductWithPortfolio](#) in der AWS CLI Befehlsreferenz.

`associate-tag-option-with-resource`

Das folgende Codebeispiel zeigt die Verwendung `associate-tag-option-with-resource`.

AWS CLI

Um a TagOption mit einer Ressource zu verknüpfen

Das folgende `associate-tag-option-with-resource` Beispiel verknüpft die angegebene Ressource TagOption mit der angegebenen Ressource.

```
aws servicecatalog associate-tag-option-with-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id tag-p3abc2pkpz5qc
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AssociateTagOptionWithResource](#) unter AWS CLI Befehlsreferenz.

copy-product

Das folgende Codebeispiel zeigt die Verwendung `copy-product`.

AWS CLI

Um ein Produkt zu kopieren

Im folgenden `copy-product` Beispiel wird eine Kopie des angegebenen Produkts erstellt, wobei eine JSON-Datei verwendet wird, um Parameter zu übergeben.

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

Inhalt von `copy-product-input.json`:

```
{  
  "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
tcabcd3syn2xy",  
  "TargetProductName": "copy-of-myproduct",  
  "CopyOptions": [  
    "CopyTags"  
  ]  
}
```

Ausgabe:

```
{  
  "CopyProductToken": "copyproduct-abc5defgjdkdji"  
}
```

- API-Einheiten finden Sie [CopyProduct](#) unter AWS CLI Befehlsreferenz.

create-portfolio-share

Das folgende Codebeispiel zeigt die Verwendung `create-portfolio-share`.

AWS CLI

Um ein Portfolio mit einem Konto zu teilen

Im folgenden `create-portfolio-share` Beispiel wird das angegebene Portfolio gemeinsam mit dem angegebenen Konto verwendet.

```
aws servicecatalog create-portfolio-share \  
  --portfolio-id port-2s6abcdef5wdh4 \  
  --account-id 794123456789
```

Dieser Befehl erzeugt keine Ausgabe.

- API-Einheiten finden Sie [CreatePortfolioShare](#) in der AWS CLI Befehlsreferenz.

create-portfolio

Das folgende Codebeispiel zeigt die Verwendung `create-portfolio`.

AWS CLI

Um ein Portfolio zu erstellen

Im folgenden `create-portfolio` Beispiel wird ein Portfolio erstellt.

```
aws servicecatalog create-portfolio \  
  --provider-name my-provider \  
  --display-name my-portfolio
```

Ausgabe:

```
{  
  "PortfolioDetail": {  
    "ProviderName": "my-provider",  
    "DisplayName": "my-portfolio",
```

```
    "CreatedTime": 1571337221.555,  
    "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/  
port-2s6xmplq5wdh4",  
    "Id": "port-2s6xmplq5wdh4"  
  }  
}
```

- API-Einheiten finden Sie [CreatePortfolio](#) in der AWS CLI Befehlsreferenz.

create-product

Das folgende Codebeispiel zeigt die Verwendung `create-product`.

AWS CLI

Um ein Produkt zu erstellen

Im folgenden `create-product` Beispiel wird ein Produkt erstellt, wobei eine JSON-Datei verwendet wird, um Parameter zu übergeben.

```
aws servicecatalog create-product \  
  --cli-input-json file://create-product-input.json
```

Inhalt von `create-product-input.json`:

```
{  
  "AcceptLanguage": "en",  
  "Name": "test-product",  
  "Owner": "test-owner",  
  "Description": "test-description",  
  "Distributor": "test-distributor",  
  "SupportDescription": "test-support",  
  "SupportEmail": "test@amazon.com",  
  "SupportUrl": "https://aws.amazon.com",  
  "ProductType": "CLOUD_FORMATION_TEMPLATE",  
  "Tags": [  
    {  
      "Key": "region",  
      "Value": "us-east-1"  
    }  
  ],  
  "ProvisioningArtifactParameters": {
```



```
    "Name": "test-version-name",
    "Description": "test-version-description",
    "Info": {
      "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
    },
    "Type": "CLOUD_FORMATION_TEMPLATE"
  }
}
```

Ausgabe:

```
{
  "Tags": [
    {
      "Key": "region",
      "Value": "us-east-1"
    }
  ],
  "ProductViewDetail": {
    "CreatedTime": 1576025036.0,
    "ProductARN": "arn:aws:catalog:us-west-2:1234568542028:product/
prod-3p5abcdef3oyk",
    "Status": "CREATED",
    "ProductViewSummary": {
      "Type": "CLOUD_FORMATION_TEMPLATE",
      "Distributor": "test-distributor",
      "SupportUrl": "https://aws.amazon.com",
      "SupportEmail": "test@amazon.com",
      "Id": "prodview-abcd42wvx45um",
      "SupportDescription": "test-support",
      "ShortDescription": "test-description",
      "Owner": "test-owner",
      "Name": "test-product2",
      "HasDefaultPath": false,
      "ProductId": "prod-3p5abcdef3oyk"
    }
  },
  "ProvisioningArtifactDetail": {
    "CreatedTime": 1576025036.0,
    "Active": true,
    "Id": "pa-pq3p5lil12a34",
    "Description": "test-version-description",
```

```
    "Name": "test-version-name",
    "Type": "CLOUD_FORMATION_TEMPLATE"
  }
}
```

- API-Einheiten finden Sie [CreateProduct](#) in der AWS CLI Befehlsreferenz.

create-provisioning-artifact

Das folgende Codebeispiel zeigt die Verwendung `create-provisioning-artifact`.

AWS CLI

Um ein Bereitstellungsartefakt zu erstellen

Im folgenden `create-provisioning-artifact` Beispiel wird ein Bereitstellungsartefakt erstellt, wobei eine JSON-Datei zur Übergabe von Parametern verwendet wird.

```
aws servicecatalog create-provisioning-artifact \
  --cli-input-json file://create-provisioning-artifact-input.json
```

Inhalt von `create-provisioning-artifact-input.json`:

```
{
  "ProductId": "prod-nfi2abcdefghi",
  "Parameters": {
    "Name": "test-provisioning-artifact",
    "Description": "test description",
    "Info": {
      "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
    },
    "Type": "CLOUD_FORMATION_TEMPLATE"
  }
}
```

Ausgabe:

```
{
  "Info": {
```

```

    "TemplateUrl": "https://s3-us-west-1.amazonaws.com/cloudformation-templates-
us-west-1/my-cfn-template.template"
  },
  "Status": "CREATING",
  "ProvisioningArtifactDetail": {
    "Id": "pa-bb4abcdefwnaio",
    "Name": "test-provisioning-artifact",
    "Description": "test description",
    "Active": true,
    "Type": "CLOUD_FORMATION_TEMPLATE",
    "CreatedTime": 1576022545.0
  }
}

```

- API-Einheiten finden Sie unter [CreateProvisioningArtifact AWS CLI Befehlsreferenz](#).

create-tag-option

Das folgende Codebeispiel zeigt die Verwendung `create-tag-option`.

AWS CLI

Um ein zu erstellen TagOption

Das folgende `create-tag-option` Beispiel erstellt eine TagOption.

```

aws servicecatalog create-tag-option
  --key 1234
  --value name

```

Ausgabe:

```

{
  "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
  }
}

```

- API-Einheiten finden Sie [CreateTagOption](#) in der AWS CLI Befehlsreferenz.

delete-portfolio-share

Das folgende Codebeispiel zeigt die Verwendung `delete-portfolio-share`.

AWS CLI

Um die gemeinsame Nutzung eines Portfolios mit einem Konto zu beenden

Im folgenden `delete-portfolio-share` Beispiel wird die gemeinsame Nutzung des Portfolios mit dem angegebenen Konto beendet.

```
aws servicecatalog delete-portfolio-share \  
  --portfolio-id port-2s6abcdq5wdh4 \  
  --account-id 123456789012
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePortfolioShare](#) in der AWS CLI Befehlsreferenz.

delete-portfolio

Das folgende Codebeispiel zeigt die Verwendung `delete-portfolio`.

AWS CLI

Um ein Portfolio zu löschen

Im folgenden `delete-portfolio` Beispiel wird das angegebene Portfolio gelöscht.

```
aws servicecatalog delete-portfolio \  
  --id port-abcdlx4gox4do
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePortfolio](#) in der AWS CLI Befehlsreferenz.

delete-product

Das folgende Codebeispiel zeigt die Verwendung `delete-product`.

AWS CLI

Um ein Produkt zu löschen

Im folgenden `delete-product` Beispiel wird das angegebene Produkt gelöscht.

```
aws servicecatalog delete-product \  
  --id prod-abcdcek6yhbx
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteProduct](#) in der AWS CLI Befehlsreferenz.

delete-provisioning-artifact

Das folgende Codebeispiel zeigt die Verwendung `delete-provisioning-artifact`.

AWS CLI

Um ein Bereitstellungsartefakt zu löschen

Im folgenden `delete-provisioning-artifact` Beispiel wird das angegebene Bereitstellungsartefakt gelöscht.

```
aws servicecatalog delete-provisioning-artifact \  
  --product-id prod-abc2uebuplcpw \  
  --provisioning-artifact-id pa-pqabcddii7ouc
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteProvisioningArtifact](#) in der AWS CLI Befehlsreferenz.

delete-tag-option

Das folgende Codebeispiel zeigt die Verwendung `delete-tag-option`.

AWS CLI

Um ein zu löschen TagOption

Das folgende `delete-tag-option` Beispiel löscht das angegebene TagOption.

```
aws servicecatalog delete-tag-option \  
  --id tag-iabcdn4fzjjms
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteTagOption](#) in der AWS CLI Befehlsreferenz.

describe-copy-product-status

Das folgende Codebeispiel zeigt die Verwendung `describe-copy-product-status`.

AWS CLI

Um den Status des Vorgangs zum Kopieren des Produkts zu beschreiben

Im folgenden `describe-copy-product-status` Beispiel wird der aktuelle Status des angegebenen asynchronen Kopiervorgangs für das Produkt angezeigt.

```
aws servicecatalog describe-copy-product-status \  
  --copy-product-token copyproduct-znn5tf5abcd3w
```

Ausgabe:

```
{  
  "CopyProductStatus": "SUCCEEDED",  
  "TargetProductId": "prod-os6hog7abcdt2"  
}
```

- API-Einheiten finden Sie unter [DescribeCopyProductStatus AWS CLI](#) Befehlsreferenz.

describe-portfolio

Das folgende Codebeispiel zeigt die Verwendung `describe-portfolio`.

AWS CLI

Um ein Portfolio zu beschreiben

Im folgenden `describe-portfolio` Beispiel werden Details für das angegebene Portfolio angezeigt.

```
aws servicecatalog describe-portfolio \  
  --id port-2s6abcdq5wdh4
```

Ausgabe:

```
{
  "TagOptions": [],
  "PortfolioDetail": {
    "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/
port-2s6abcdq5wdh4",
    "Id": "port-2s6wuzyzq5wdh4",
    "CreatedTime": 1571337221.555,
    "DisplayName": "my-portfolio",
    "ProviderName": "my-provider"
  },
  "Tags": []
}
```

- API-Einheiten finden Sie [DescribePortfolio](#) unter AWS CLI Befehlsreferenz.

describe-product-as-admin

Das folgende Codebeispiel zeigt die Verwendung `describe-product-as-admin`.

AWS CLI

Um ein Produkt als Administrator zu beschreiben

Im folgenden `describe-product-as-admin` Beispiel werden Details für das angegebene Produkt mit Administratorrechten angezeigt.

```
aws servicecatalog describe-product-as-admin \
  --id prod-abcdek6yhbx1
```

Ausgabe:

```
{
  "TagOptions": [],
  "ProductViewDetail": {
    "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-
abcdek6yhbx1",
    "ProductViewSummary": {
      "SupportEmail": "test@amazon.com",
      "Type": "CLOUD_FORMATION_TEMPLATE",
      "Distributor": "test-distributor",
```

```

    "ShortDescription": "test-description",
    "Owner": "test-owner",
    "Id": "prodview-wi3l2j4abc6vc",
    "SupportDescription": "test-support",
    "ProductId": "prod-abcdcek6yhbxix",
    "HasDefaultPath": false,
    "Name": "test-product3",
    "SupportUrl": "https://aws.amazon.com"
  },
  "CreatedTime": 1577136715.0,
  "Status": "CREATED"
},
"ProvisioningArtifactSummaries": [
  {
    "CreatedTime": 1577136715.0,
    "Description": "test-version-description",
    "ProvisioningArtifactMetadata": {
      "SourceProvisioningArtifactId": "pa-abcdxkkiv5fcm"
    },
    "Name": "test-version-name-3",
    "Id": "pa-abcdxkkiv5fcm"
  }
],
"Tags": [
  {
    "Value": "iad",
    "Key": "region"
  }
]
}

```

- API-Einheiten finden Sie [DescribeProductAsAdmin](#) in der AWS CLI Befehlsreferenz.

describe-provisioned-product

Das folgende Codebeispiel zeigt die Verwendung `describe-provisioned-product`.

AWS CLI

Um ein bereitgestelltes Produkt zu beschreiben

Im folgenden `describe-provisioned-product` Beispiel werden Details für das angegebene bereitgestellte Produkt angezeigt.


```
aws servicecatalog describe-provisioned-product \  
  --id pp-dpom27bm4abcd
```

Ausgabe:

```
{  
  "ProvisionedProductDetail": {  
    "Status": "ERROR",  
    "CreatedTime": 1577222793.358,  
    "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/mytestppname3/  
pp-dpom27bm4abcd",  
    "Id": "pp-dpom27bm4abcd",  
    "StatusMessage": "AmazonCloudFormationException Parameters: [KeyName]  
must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:  
ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",  
    "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",  
    "LastRecordId": "rec-tfuawdjovzxe",  
    "Type": "CFN_STACK",  
    "Name": "mytestppname3"  
  },  
  "CloudWatchDashboards": []  
}
```

- API-Einheiten finden Sie [DescribeProvisionedProduct](#) in der AWS CLI Befehlsreferenz.

describe-provisioning-artifact

Das folgende Codebeispiel zeigt die Verwendung `describe-provisioning-artifact`.

AWS CLI

Um ein Bereitstellungsartefakt zu beschreiben

Im folgenden `describe-provisioning-artifact` Beispiel werden Details für das angegebene Bereitstellungsartefakt angezeigt.

```
aws servicecatalog describe-provisioning-artifact \  
  --provisioning-artifact-id pa-pcz347abcdcfm \  
  --product-id prod-abcdfz3syn2rg
```

Ausgabe:

```
{
  "Info": {
    "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
  },
  "ProvisioningArtifactDetail": {
    "Id": "pa-pcz347abcdcfm",
    "Active": true,
    "Type": "CLOUD_FORMATION_TEMPLATE",
    "Description": "updated description",
    "CreatedTime": 1562097906.0,
    "Name": "updated name"
  },
  "Status": "AVAILABLE"
}
```

- API-Einheiten finden Sie unter [DescribeProvisioningArtifact AWS CLI-Befehlsreferenz](#).

describe-tag-option

Das folgende Codebeispiel zeigt die Verwendung `describe-tag-option`.

AWS CLI

Um einen zu beschreiben TagOption

Im folgenden `describe-tag-option` Beispiel werden Details für das angegebene Objekt angezeigt TagOption.

```
aws servicecatalog describe-tag-option \
  --id tag-p3tej2abcd5qc
```

Ausgabe:

```
{
  "TagOptionDetail": {
    "Active": true,
    "Id": "tag-p3tej2abcd5qc",
    "Value": "value-3",
    "Key": "1234"
  }
}
```

```
}
```

- API-Einheiten finden Sie [DescribeTagOption](#) unter AWS CLI Befehlsreferenz.

disassociate-principal-from-portfolio

Das folgende Codebeispiel zeigt die Verwendung `disassociate-principal-from-portfolio`.

AWS CLI

Um einen Principal von einem Portfolio zu trennen

Im folgenden `disassociate-principal-from-portfolio` Beispiel wird die Zuordnung des angegebenen Prinzipals zum Portfolio aufgehoben.

```
aws servicecatalog disassociate-principal-from-portfolio \  
  --portfolio-id port-2s6abcdq5wdh4 \  
  --principal-arn arn:aws:iam::123456789012:group/myendusers
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DisassociatePrincipalFromPortfolio AWS CLI](#) Befehlsreferenz.

disassociate-product-from-portfolio

Das folgende Codebeispiel zeigt die Verwendung `disassociate-product-from-portfolio`.

AWS CLI

Um ein Produkt von einem Portfolio zu trennen

Im folgenden `disassociate-product-from-portfolio` Beispiel wird die Zuordnung des angegebenen Produkts zum Portfolio aufgehoben.

```
aws servicecatalog disassociate-product-from-portfolio \  
  --product-id prod-3p5abcdmu3oyk \  
  --portfolio-id port-2s6abcdq5wdh4
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DisassociateProductFromPortfolio](#) in der AWS CLI Befehlsreferenz.

disassociate-tag-option-from-resource

Das folgende Codebeispiel zeigt die Verwendung `disassociate-tag-option-from-resource`.

AWS CLI

Um die Zuordnung von a zu einer TagOption Ressource zu trennen

Im folgenden `disassociate-tag-option-from-resource` Beispiel wird die Verknüpfung zwischen dem angegebenen Objekt und der TagOption Ressource aufgehoben.

```
aws servicecatalog disassociate-tag-option-from-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id tag-p3abc2pkpz5qc
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API Einheiten finden Sie unter [DisassociateTagOptionFromResource AWS CLI](#) Befehlsreferenz.

list-accepted-portfolio-shares

Das folgende Codebeispiel zeigt die Verwendung `list-accepted-portfolio-shares`.

AWS CLI

Um akzeptierte Portfolioaktien aufzulisten

Das folgende `list-accepted-portfolio-shares` Beispiel listet alle Portfolios auf, für die das Teilen von diesem Konto akzeptiert wurde, einschließlich nur der standardmäßigen Service Catalog-Portfolios.

```
aws servicecatalog list-accepted-portfolio-shares \  
  --portfolio-share-type "AWS_SERVICECATALOG"
```

Ausgabe:

```
{  
  "PortfolioDetails": [  
    {  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-  
d2abcd5dpkuma",
```

```

        "Description": "AWS Service Catalog Reference blueprints for often-used
AWS services such as EC2, S3, RDS, VPC and EMR.",
        "CreatedTime": 1574456190.687,
        "ProviderName": "AWS Service Catalog",
        "DisplayName": "Reference Architectures",
        "Id": "port-d2abcd5dpkuma"
    },
    {
        "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-
abcdefaua7zpu",
        "Description": "AWS well-architected blueprints for high reliability
applications.",
        "CreatedTime": 1574461496.092,
        "ProviderName": "AWS Service Catalog",
        "DisplayName": "High Reliability Architectures",
        "Id": "port-abcdefaua7zpu"
    }
]
}

```

- API-Einheiten finden Sie [ListAcceptedPortfolioShares](#) in der AWS CLI Befehlsreferenz.

list-portfolio-access

Das folgende Codebeispiel zeigt die Verwendung `list-portfolio-access`.

AWS CLI

Um Konten mit Zugriff auf ein Portfolio aufzulisten

Das folgende `list-portfolio-access` Beispiel listet die AWS Konten auf, die Zugriff auf das angegebene Portfolio haben.

```
aws servicecatalog list-portfolio-access \
  --portfolio-id port-2s6abcdq5wdh4
```

Ausgabe:

```
{
  "AccountIds": [
    "123456789012"
  ]
}
```

```
}
```

- API-Einheiten finden Sie [ListPortfolioAccess](#) unter AWS CLI Befehlsreferenz.

list-portfolios-for-product

Das folgende Codebeispiel zeigt die Verwendung `list-portfolios-for-product`.

AWS CLI

Um Portfolios aufzulisten, die einem Produkt zugeordnet sind

Das folgende `list-portfolios-for-product` Beispiel listet die Portfolios auf, die dem angegebenen Produkt zugeordnet sind.

```
aws servicecatalog list-portfolios-for-product \  
  --product-id prod-abcdefz3syn2rg
```

Ausgabe:

```
{  
  "PortfolioDetails": [  
    {  
      "CreatedTime": 1571337221.555,  
      "Id": "port-2s6abcdq5wdh4",  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/  
port-2s6abcdq5wdh4",  
      "DisplayName": "my-portfolio",  
      "ProviderName": "my-provider"  
    },  
    {  
      "CreatedTime": 1559665256.348,  
      "Id": "port-5abcd3e5st4ei",  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/  
port-5abcd3e5st4ei",  
      "DisplayName": "test",  
      "ProviderName": "provider-name"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListPortfoliosForProduct](#) unter AWS CLI Befehlsreferenz.

list-portfolios

Das folgende Codebeispiel zeigt die Verwendung `list-portfolios`.

AWS CLI

Um Portfolios aufzulisten

Im folgenden `list-portfolios` Beispiel werden die Service Catalog-Portfolios in der aktuellen Region aufgeführt.

```
aws servicecatalog list-portfolios
```

Ausgabe:

```
{
  "PortfolioDetails": [
    {
      "CreatedTime": 1559665256.348,
      "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmlst4ei",
      "DisplayName": "my-portfolio",
      "Id": "port-5pzcxmlst4ei",
      "ProviderName": "my-user"
    }
  ]
}
```

- API-Einheiten finden Sie [ListPortfolios](#) in der AWS CLI Befehlsreferenz.

list-principals-for-portfolio

Das folgende Codebeispiel zeigt die Verwendung `list-principals-for-portfolio`.

AWS CLI

Um alle Principals für ein Portfolio aufzulisten

Im folgenden `list-principals-for-portfolio` Beispiel werden alle Principals für das angegebene Portfolio aufgeführt.

```
aws servicecatalog list-principals-for-portfolio \
```

```
--portfolio-id port-2s6abcdq5wdh4
```

Ausgabe:

```
{
  "Principals": [
    {
      "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
      "PrincipalType": "IAM"
    }
  ]
}
```

- API-Einheiten finden Sie unter [ListPrincipalsForPortfolio AWS CLI Befehlsreferenz](#).

list-provisioning-artifacts

Das folgende Codebeispiel zeigt die Verwendung `list-provisioning-artifacts`.

AWS CLI

Um alle Bereitstellungsartefakte für ein Produkt aufzulisten

Im folgenden `list-provisioning-artifacts` Beispiel werden alle Bereitstellungsartefakte für das angegebene Produkt aufgeführt.

```
aws servicecatalog list-provisioning-artifacts \
  --product-id prod-nfi2abcdefgcpw
```

Ausgabe:

```
{
  "ProvisioningArtifactDetails": [
    {
      "Id": "pa-abcdef54ipm6z",
      "Description": "test-version-description",
      "Type": "CLOUD_FORMATION_TEMPLATE",
      "CreatedTime": 1576021147.0,
      "Active": true,
      "Name": "test-version-name"
    },
  ],
}
```



```

    {
      "Id": "pa-bb4zyxwwnaio",
      "Description": "test description",
      "Type": "CLOUD_FORMATION_TEMPLATE",
      "CreatedTime": 1576022545.0,
      "Active": true,
      "Name": "test-provisioning-artifact-2"
    }
  ]
}

```

- API-Einheiten finden Sie [ListProvisioningArtifacts](#) in der AWS CLI Befehlsreferenz.

list-resources-for-tag-option

Das folgende Codebeispiel zeigt die Verwendung `list-resources-for-tag-option`.

AWS CLI

Um Ressourcen aufzulisten, die einem zugeordnet sind `TagOption`

Das folgende `list-resources-for-tag-option` Beispiel listet die Ressourcen auf, die dem angegebenen Objekt zugeordnet sind `TagOption`.

```

aws servicecatalog list-resources-for-tag-option \
  --tag-option-id tag-p3tej2abcd5qc

```

Ausgabe:

```

{
  "ResourceDetails": [
    {
      "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
      "Name": "my product",
      "Description": "description",
      "CreatedTime": 1562097906.0,
      "Id": "prod-abcdfz3syn2rg"
    }
  ]
}

```

- API-Einheiten finden Sie [ListResourcesForTagOption](#) unter AWS CLI Befehlsreferenz.

list-tag-options

Das folgende Codebeispiel zeigt die Verwendung `list-tag-options`.

AWS CLI

Das folgende `list-tag-options` Beispiel listet alle Werte für `listTagOptions` auf.

```
aws servicecatalog list-tag-options
```

Ausgabe:

```
{
  "TagOptionDetails": [
    {
      "Value": "newvalue",
      "Active": true,
      "Id": "tag-iabcdn4fzjjms",
      "Key": "1234"
    },
    {
      "Value": "value1",
      "Active": true,
      "Id": "tag-e3abcdvmwvrzy",
      "Key": "key"
    }
  ]
}
```

- API-Einheiten finden Sie [ListTagOptions](#) in der AWS CLI Befehlsreferenz.

provision-product

Das folgende Codebeispiel zeigt die Verwendung `provision-product`.

AWS CLI

Um ein Produkt bereitzustellen

Im folgenden `provision-product` Beispiel wird das angegebene Produkt mithilfe des angegebenen Bereitstellungsartefakts bereitgestellt.

```
aws servicecatalog provision-product \  
  --product-id prod-abcdefz3syn2rg \  
  --provisioning-artifact-id pa-abc347pcscfm \  
  --provisioned-product-name "mytestppname3"
```

Ausgabe:

```
{  
  "RecordDetail": {  
    "RecordId": "rec-tfuawdabcdege",  
    "CreatedTime": 1577222793.362,  
    "ProvisionedProductId": "pp-abcd27bm4mldq",  
    "PathId": "lpv2-abcdg3jp6t5k6",  
    "RecordErrors": [],  
    "ProductId": "prod-abcdefz3syn2rg",  
    "UpdatedTime": 1577222793.362,  
    "RecordType": "PROVISION_PRODUCT",  
    "ProvisionedProductName": "mytestppname3",  
    "ProvisioningArtifactId": "pa-pcz347abcdcfm",  
    "RecordTags": [],  
    "Status": "CREATED",  
    "ProvisionedProductType": "CFN_STACK"  
  }  
}
```

- API-Einheiten finden Sie [ProvisionProduct](#) in der AWS CLI Befehlsreferenz.

reject-portfolio-share

Das folgende Codebeispiel zeigt die Verwendung `reject-portfolio-share`.

AWS CLI

Um eine Portfolioaktie abzulehnen

Im folgenden `reject-portfolio-share` Beispiel wird der Portfolioanteil für das angegebene Portfolio abgelehnt.

```
aws servicecatalog reject-portfolio-share \  
  --portfolio-id prod-abcdefz3syn2rg \  
  --portfolio-name "mytestppname3"
```

```
--portfolio-id port-2s6wuabcdefghijk
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [RejectPortfolioShare](#) in der AWS CLI Befehlsreferenz.

scan-provisioned-products

Das folgende Codebeispiel zeigt die Verwendung `scan-provisioned-products`.

AWS CLI

Um alle verfügbaren bereitgestellten Produkte aufzulisten

Das folgende `scan-provisioned-products` Beispiel listet die verfügbaren bereitgestellten Produkte auf.

```
aws servicecatalog scan-provisioned-products
```

Ausgabe:

```
{
  "ProvisionedProducts": [
    {
      "Status": "ERROR",
      "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/mytestppname3/pp-abcd27bm4mldq",
      "StatusMessage": "AmazonCloudFormationException Parameters: [KeyName] must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code: ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
      "Id": "pp-abcd27bm4mldq",
      "Type": "CFN_STACK",
      "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
      "CreatedTime": 1577222793.358,
      "Name": "mytestppname3",
      "LastRecordId": "rec-tfuawdabcdxge"
    }
  ]
}
```

- API-Einheiten finden Sie [ScanProvisionedProducts](#) in der AWS CLI Befehlsreferenz.

search-products-as-admin

Das folgende Codebeispiel zeigt die Verwendung `search-products-as-admin`.

AWS CLI

Um nach Produkten mit Administratorrechten zu suchen

Im folgenden `search-products-as-admin` Beispiel wird nach Produkten mit Administratorrechten gesucht, wobei eine Portfolio-ID als Filter verwendet wird.

```
aws servicecatalog search-products-as-admin \  
  --portfolio-id port-5abcd3e5st4ei
```

Ausgabe:

```
{  
  "ProductViewDetails": [  
    {  
      "ProductViewSummary": {  
        "Name": "my product",  
        "Owner": "owner name",  
        "Type": "CLOUD_FORMATION_TEMPLATE",  
        "ProductId": "prod-abcdefz3syn2rg",  
        "HasDefaultPath": false,  
        "Id": "prodview-abcdmyuzv2dlu",  
        "ShortDescription": "description"  
      },  
      "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
abcdefz3syn2rg",  
      "CreatedTime": 1562097906.0,  
      "Status": "CREATED"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [SearchProductsAsAdmin](#) in der AWS CLI Befehlsreferenz.

search-provisioned-products

Das folgende Codebeispiel zeigt die Verwendung `search-provisioned-products`.

AWS CLI

Um bereitgestellte Produkte zu suchen

Im folgenden `search-provisioned-products` Beispiel wird nach bereitgestellten Produkten gesucht, die der angegebenen Produkt-ID entsprechen, wobei eine JSON Datei verwendet wird, um Parameter zu übergeben.

```
aws servicecatalog search-provisioned-products \  
  --cli-input-json file://search-provisioned-products-input.json
```

Inhalt von `search-provisioned-products-input.json`:

```
{  
  "Filters": {  
    "SearchQuery": [  
      "prod-tcjevz3syn2rg"  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "ProvisionedProducts": [  
    {  
      "ProvisioningArtifactId": "pa-pcz347abcdcfm",  
      "Name": "mytestppname3",  
      "CreatedTime": 1577222793.358,  
      "Id": "pp-abcd27bm4mldq",  
      "Status": "ERROR",  
      "UserArn": "arn:aws:iam::123456789012:user/cliuser",  
      "StatusMessage": "AmazonCloudFormationException Parameters: [KeyName]  
must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:  
ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",  
      "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/  
mytestppname3/pp-abcd27bm4mldq",  
      "Tags": [  
        {  
          "Value": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
abcdfz3syn2rg",  
          "Key": "aws:servicecatalog:productArn"  
        }  
      ]  
    }  
  ]  
}
```

```

    },
    {
      "Value": "arn:aws:iam::123456789012:user/cliuser",
      "Key": "aws:servicecatalog:provisioningPrincipalArn"
    },
    {
      "Value": "value-3",
      "Key": "1234"
    },
    {
      "Value": "pa-pcz347abcdcfm",
      "Key": "aws:servicecatalog:provisioningArtifactIdentifier"
    },
    {
      "Value": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
      "Key": "aws:servicecatalog:portfolioArn"
    },
    {
      "Value": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
      "Key": "aws:servicecatalog:provisionedProductArn"
    }
  ],
  "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
  "UserArnSession": "arn:aws:iam::123456789012:user/cliuser",
  "Type": "CFN_STACK",
  "LastRecordId": "rec-tfuawdabcdxge",
  "ProductId": "prod-abcdfz3syn2rg"
}
],
"TotalResultsCount": 1
}

```

- API-Einheiten finden Sie unter [SearchProvisionedProducts AWS CLI](#) Befehlsreferenz.

update-portfolio

Das folgende Codebeispiel zeigt die Verwendung `update-portfolio`.

AWS CLI

Um ein Portfolio zu aktualisieren

Im folgenden `update-portfolio` Beispiel wird der Name des angegebenen Portfolios aktualisiert.

```
aws servicecatalog update-portfolio \  
  --id port-5abcd3e5st4ei \  
  --display-name "New portfolio name"
```

Ausgabe:

```
{  
  "PortfolioDetail": {  
    "DisplayName": "New portfolio name",  
    "ProviderName": "provider",  
    "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/  
port-5abcd3e5st4ei",  
    "Id": "port-5abcd3e5st4ei",  
    "CreatedTime": 1559665256.348  
  },  
  "Tags": []  
}
```

- API-Einheiten finden Sie [UpdatePortfolio](#) unter AWS CLI Befehlsreferenz.

update-product

Das folgende Codebeispiel zeigt die Verwendung `update-product`.

AWS CLI

Um ein Produkt zu aktualisieren

Im folgenden `update-product` Beispiel werden der Name und der Besitzer des angegebenen Produkts aktualisiert.

```
aws servicecatalog update-product \  
  --id prod-os6abc7drqlt2 \  
  --name "New product name" \  
  --owner "Updated product owner"
```

Ausgabe:

```
{
```



```
"Tags": [
  {
    "Value": "iad",
    "Key": "region"
  }
],
"ProductViewDetail": {
  "ProductViewSummary": {
    "Owner": "Updated product owner",
    "ProductId": "prod-os6abc7drqlt2",
    "Distributor": "test-distributor",
    "SupportUrl": "https://aws.amazon.com",
    "Name": "New product name",
    "ShortDescription": "test-description",
    "HasDefaultPath": false,
    "Id": "prodview-6abcdgrfhvidy",
    "SupportDescription": "test-support",
    "SupportEmail": "test@amazon.com",
    "Type": "CLOUD_FORMATION_TEMPLATE"
  },
  "Status": "CREATED",
  "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
os6abc7drqlt2",
  "CreatedTime": 1577136255.0
}
}
```

- API-Einheiten finden Sie [UpdateProduct](#) unter AWS CLI Befehlsreferenz.

update-provisioning-artifact

Das folgende Codebeispiel zeigt die Verwendung `update-provisioning-artifact`.

AWS CLI

Um ein Provisioning-Artefakt zu aktualisieren

Im folgenden `update-provisioning-artifact` Beispiel werden der Name und die Beschreibung des angegebenen Bereitstellungsartefakts aktualisiert, wobei eine JSON-Datei zur Übergabe von Parametern verwendet wird.

```
aws servicecatalog update-provisioning-artifact \
```

```
--cli-input-json file://update-provisioning-artifact-input.json
```

Inhalt von `update-provisioning-artifact-input.json`:

```
{
  "ProductId": "prod-abcdefz3syn2rg",
  "ProvisioningArtifactId": "pa-pcz347abcdcfm",
  "Name": "updated name",
  "Description": "updated description"
}
```

Ausgabe:

```
{
  "Info": {
    "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
  },
  "Status": "AVAILABLE",
  "ProvisioningArtifactDetail": {
    "Active": true,
    "Description": "updated description",
    "Id": "pa-pcz347abcdcfm",
    "Name": "updated name",
    "Type": "CLOUD_FORMATION_TEMPLATE",
    "CreatedTime": 1562097906.0
  }
}
```

- API-Einheiten finden Sie unter [UpdateProvisioningArtifact AWS CLI Befehlsreferenz](#).

update-tag-option

Das folgende Codebeispiel zeigt die Verwendung `update-tag-option`.

AWS CLI

Um ein zu aktualisieren TagOption

Im folgenden `update-tag-option` Beispiel wird der Wert von a TagOption mithilfe der angegebenen JSON Datei aktualisiert.

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-input.json
```

Inhalt von `update-tag-option-input.json`:

```
{
  "Id": "tag-iabcdn4fzjjms",
  "Value": "newvalue",
  "Active": true
}
```

Ausgabe:

```
{
  "TagOptionDetail": {
    "Value": "newvalue",
    "Key": "1234",
    "Active": true,
    "Id": "tag-iabcdn4fzjjms"
  }
}
```

- API-Einheiten finden Sie [UpdateTagOption](#) unter AWS CLI Befehlsreferenz.

Beispiele für Service Quotas mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Service Quotas Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-aws-default-service-quota

Das folgende Codebeispiel zeigt die Verwendung `get-aws-default-service-quota`.

AWS CLI

Um ein Standard-Servicekontingent zu beschreiben

Im folgenden `get-aws-default-service-quota` Beispiel werden Details für das angegebene Kontingent angezeigt.

```
aws service-quotas get-aws-default-service-quota \  
  --service-code ec2 \  
  --quota-code L-1216C47A
```

Ausgabe:

```
{  
  "Quota": {  
    "ServiceCode": "ec2",  
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",  
    "QuotaArn": "arn:aws:servicequotas:us-east-2::ec2/L-1216C47A",  
    "QuotaCode": "L-1216C47A",  
    "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)  
instances",  
    "Value": 5.0,  
    "Unit": "None",  
    "Adjustable": true,  
    "GlobalQuota": false,  
    "UsageMetric": {  
      "MetricNamespace": "AWS/Usage",  
      "MetricName": "ResourceCount",  
      "MetricDimensions": {  
        "Class": "Standard/OnDemand",  
        "Resource": "vCPU",  
        "Service": "EC2",  
        "Type": "Resource"  
      },  
      "MetricStatisticRecommendation": "Maximum"  
    }  
  }  
}
```

```
}

```

- API-Einheiten finden Sie [GetAwsDefaultServiceQuota](#) in der AWS CLI Befehlsreferenz.

get-requested-service-quota-change

Das folgende Codebeispiel zeigt, wie man es benutzt `get-requested-service-quota-change`.

AWS CLI

Um eine Anfrage zur Erhöhung des Servicekontingents zu beschreiben

Das folgende `get-requested-service-quota-change` Beispiel beschreibt die angegebene Anfrage zur Erhöhung des Kontingents.

```
aws service-quotas get-requested-service-quota-change \
  --request-id d187537d15254312a9609aa51bbf7624u7W49tP0
```

Ausgabe:

```
{
  "RequestedQuota": {
    "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",
    "CaseId": "6780195351",
    "ServiceCode": "ec2",
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
    "QuotaCode": "L-20F13EBD",
    "QuotaName": "Running Dedicated c5n Hosts",
    "DesiredValue": 2.0,
    "Status": "CASE_OPENED",
    "Created": 1580446904.067,
    "LastUpdated": 1580446953.265,
    "Requester": "{\"accountId\":\"123456789012\", \"callerArn\": \"arn:aws:iam::123456789012:root\"}",
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",
    "GlobalQuota": false,
    "Unit": "None"
  }
}
```

- API-Einheiten finden Sie [GetRequestedServiceQuotaChange](#) in der AWS CLI Befehlsreferenz.

get-service-quota

Das folgende Codebeispiel zeigt, wie man es benutzt `get-service-quota`.

AWS CLI

Um ein Servicekontingent zu beschreiben

Im folgenden `get-service-quota` Beispiel werden Details zum angegebenen Kontingent angezeigt.

```
aws service-quotas get-service-quota \  
  --service-code ec2 \  
  --quota-code L-1216C47A
```

Ausgabe:

```
{  
  "Quota": {  
    "ServiceCode": "ec2",  
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",  
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-1216C47A",  
    "QuotaCode": "L-1216C47A",  
    "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)  
instances",  
    "Value": 1920.0,  
    "Unit": "None",  
    "Adjustable": true,  
    "GlobalQuota": false,  
    "UsageMetric": {  
      "MetricNamespace": "AWS/Usage",  
      "MetricName": "ResourceCount",  
      "MetricDimensions": {  
        "Class": "Standard/OnDemand",  
        "Resource": "vCPU",  
        "Service": "EC2",  
        "Type": "Resource"  
      },  
      "MetricStatisticRecommendation": "Maximum"  
    }  
  }  
}
```

- API-Einheiten finden Sie [GetServiceQuota](#) unter AWS CLI Befehlsreferenz.

list-aws-default-service-quotas

Das folgende Codebeispiel zeigt, wie man es benutzt `list-aws-default-service-quotas`.

AWS CLI

Um die Standardkontingente für einen Dienst aufzulisten

Das folgende `list-aws-default-service-quotas` Beispiel listet die Standardwerte für die Kontingente für den angegebenen Dienst auf.

```
aws service-quotas list-aws-default-service-quotas \
  --service-code xray
```

Ausgabe:

```
{
  "Quotas": [
    {
      "ServiceCode": "xray",
      "ServiceName": "AWS X-Ray",
      "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-C6B6F05D",
      "QuotaCode": "L-C6B6F05D",
      "QuotaName": "Indexed annotations per trace",
      "Value": 50.0,
      "Unit": "None",
      "Adjustable": false,
      "GlobalQuota": false
    },
    {
      "ServiceCode": "xray",
      "ServiceName": "AWS X-Ray",
      "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-D781C0FD",
      "QuotaCode": "L-D781C0FD",
      "QuotaName": "Segment document size",
      "Value": 64.0,
      "Unit": "Kilobytes",
      "Adjustable": false,
      "GlobalQuota": false
    }
  ],
}
```

```

    {
      "ServiceCode": "xray",
      "ServiceName": "AWS X-Ray",
      "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-998BFF16",
      "QuotaCode": "L-998BFF16",
      "QuotaName": "Trace and service graph retention in days",
      "Value": 30.0,
      "Unit": "None",
      "Adjustable": false,
      "GlobalQuota": false
    }
  ]
}

```

- API-Einheiten finden Sie [ListAwsDefaultServiceQuotas](#) in der AWS CLI Befehlsreferenz.

list-requested-service-quota-change-history-by-quota

Das folgende Codebeispiel zeigt, wie man es benutzt `list-requested-service-quota-change-history-by-quota`.

AWS CLI

Um Ihre Anfragen zur Erhöhung des Kontingents aufzulisten

Im folgenden `list-requested-service-quota-change-history-by-quota` Beispiel werden die Anfragen zur Erhöhung des Kontingents für das angegebene Kontingent aufgeführt.

```

aws service-quotas list-requested-service-quota-change-history-by-quota \
  --service-code ec2 \
  --quota-code L-20F13EBD

```

Ausgabe:

```

{
  "RequestedQuotas": [
    {
      "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",
      "CaseId": "6780195351",
      "ServiceCode": "ec2",
      "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
      "QuotaCode": "L-20F13EBD",
    }
  ]
}

```



```

    "QuotaName": "Running Dedicated c5n Hosts",
    "DesiredValue": 2.0,
    "Status": "CASE_OPENED",
    "Created": 1580446904.067,
    "LastUpdated": 1580446953.265,
    "Requester": "{\"accountId\":\"123456789012\", \"callerArn\":
\\\"arn:aws:iam::123456789012:root\\\"}",
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
    "GlobalQuota": false,
    "Unit": "None"
  }
]
}

```

- API-Einheiten finden Sie [ListRequestedServiceQuotaChangeHistoryByQuota](#) in der AWS CLI Befehlsreferenz.

list-requested-service-quota-change-history

Das folgende Codebeispiel zeigt, wie man es benutzt `list-requested-service-quota-change-history`.

AWS CLI

Um Ihre Anfragen zur Erhöhung des Kontingents aufzulisten

Im folgenden `list-requested-service-quota-change-history` Beispiel werden die Anfragen zur Erhöhung des Kontingents für den angegebenen Dienst aufgeführt.

```
aws service-quotas list-requested-service-quota-change-history \
  --service-code ec2
```

Ausgabe:

```

{
  "RequestedQuotas": [
    {
      "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",
      "CaseId": "6780195351",
      "ServiceCode": "ec2",
      "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",

```

```

    "QuotaCode": "L-20F13EBD",
    "QuotaName": "Running Dedicated c5n Hosts",
    "DesiredValue": 2.0,
    "Status": "CASE_OPENED",
    "Created": 1580446904.067,
    "LastUpdated": 1580446953.265,
    "Requester": "{\"accountId\":\"123456789012\", \"callerArn\":
\\\"arn:aws:iam::123456789012:root\\\"}\",
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
    "GlobalQuota": false,
    "Unit": "None"
  }
]
}

```

- API-Einheiten finden Sie [ListRequestedServiceQuotaChangeHistory](#) in der AWS CLI Befehlsreferenz.

list-service-quotas

Das folgende Codebeispiel zeigt, wie man es benutzt `list-service-quotas`.

AWS CLI

Um die Kontingente für einen Dienst aufzulisten

Im folgenden `list-service-quotas` Beispiel werden Details zu den Kontingenten für angezeigt AWS CloudFormation.

```

aws service-quotas list-service-quotas \
  --service-code cloudformation

```

Ausgabe:

```

{
  "Quotas": [
    {
      "ServiceCode": "cloudformation",
      "ServiceName": "AWS CloudFormation",
      "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-87D14FB7",

```

```

        "QuotaCode": "L-87D14FB7",
        "QuotaName": "Output count in CloudFormation template",
        "Value": 60.0,
        "Unit": "None",
        "Adjustable": false,
        "GlobalQuota": false
    },
    {
        "ServiceCode": "cloudformation",
        "ServiceName": "AWS CloudFormation",
        "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-0485CB21",
        "QuotaCode": "L-0485CB21",
        "QuotaName": "Stack count",
        "Value": 200.0,
        "Unit": "None",
        "Adjustable": true,
        "GlobalQuota": false
    }
]
}

```

- API-Einheiten finden Sie [ListServiceQuotas](#) in der AWS CLI Befehlsreferenz.

list-services

Das folgende Codebeispiel zeigt, wie man es benutzt `list-services`.

AWS CLI

Um die verfügbaren Dienste aufzulisten

Der folgende Befehl listet die Dienste auf, die in Service Quotas verfügbar sind.

```
aws service-quotas list-services
```

Ausgabe:

```

{
  "Services": [
    {
      "ServiceCode": "AWSCloudMap",

```

```

    "ServiceName": "AWS Cloud Map"
  },
  {
    "ServiceCode": "access-analyzer",
    "ServiceName": "Access Analyzer"
  },
  {
    "ServiceCode": "acm",
    "ServiceName": "AWS Certificate Manager (ACM)"
  },
  ...truncated...
  {
    "ServiceCode": "xray",
    "ServiceName": "AWS X-Ray"
  }
]
}

```

Sie können den `--query` Parameter hinzufügen, um die Anzeige nach den Informationen zu filtern, an denen Sie interessiert sind. Im folgenden Beispiel werden nur die Servicecodes angezeigt.

```

aws service-quotas list-services \
  --query Services[*].ServiceCode

```

Ausgabe:

```

[
  "AWSCloudMap",
  "access-analyzer",
  "acm",
  "acm-pca",
  "amplify",
  "apigateway",
  "application-autoscaling",
  ...truncated...
  "xray"
]

```

- API-Einheiten finden Sie [ListServices](#) in der AWS CLI Befehlsreferenz.

request-service-quota-increase

Das folgende Codebeispiel zeigt, wie man es benutzt `request-service-quota-increase`.

AWS CLI

Um eine Erhöhung des Servicekontingents zu beantragen

Im folgenden `request-service-quota-increase` Beispiel wird eine Erhöhung des angegebenen Servicekontingents angefordert.

```
aws service-quotas request-service-quota-increase \  
  --service-code ec2 \  
  --quota-code L-20F13EBD \  
  --desired-value 2
```

Ausgabe:

```
{  
  "RequestedQuota": {  
    "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",  
    "ServiceCode": "ec2",  
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",  
    "QuotaCode": "L-20F13EBD",  
    "QuotaName": "Running Dedicated c5n Hosts",  
    "DesiredValue": 2.0,  
    "Status": "PENDING",  
    "Created": 1580446904.067,  
    "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":  
  \":arn:aws:iam::123456789012:root\"}",  
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",  
    "GlobalQuota": false,  
    "Unit": "None"  
  }  
}
```

- API-Einheiten finden Sie [RequestServiceQuotaIncrease](#) in der AWS CLI Befehlsreferenz.

SES Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden SES.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

delete-identity

Das folgende Codebeispiel zeigt die Verwendung `delete-identity`.

AWS CLI

So löschen Sie eine Identität

Im folgenden Beispiel wird der `delete-identity` Befehl verwendet, um eine Identität aus der Liste der bei Amazon SES verifizierten Identitäten zu löschen:

```
aws ses delete-identity --identity user@example.com
```

Weitere Informationen zu verifizierten Identitäten finden Sie unter [Verifying Email Addresses and Domains in Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [DeleteIdentity](#) in der AWS CLI Befehlsreferenz.

get-identity-dkim-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-identity-dkim-attributes`.

AWS CLI

Um die Amazon SES DKIM Easy-Attribute für eine Liste von Identitäten abzurufen

Im folgenden Beispiel wird der `get-identity-dkim-attributes` Befehl verwendet, um die Amazon SES DKIM Easy-Attribute für eine Liste von Identitäten abzurufen:

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

Ausgabe:

```
{
  "DkimAttributes": {
    "example.com": {
      "DkimTokens": [
        "EXAMPLEjcs5xoyqytjsotsijas7236gr",
        "EXAMPLEjr76cvoc6mysspnioorxsn6ep",
        "EXAMPLEkbnkqkhlm2lyz77ppkulerm4k"
      ],
      "DkimEnabled": true,
      "DkimVerificationStatus": "Success"
    },
    "user@example.com": {
      "DkimEnabled": false,
      "DkimVerificationStatus": "NotStarted"
    }
  }
}
```

Wenn Sie diesen Befehl mit einer Identität aufrufen, die Sie noch nie zur Überprüfung eingereicht haben, wird diese Identität nicht in der Ausgabe angezeigt.

Weitere Informationen zu Easy DKIM finden Sie unter Easy DKIM in Amazon SES im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [GetIdentityDkimAttributes](#) unter AWS CLI Befehlsreferenz.

get-identity-notification-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-identity-notification-attributes`.

AWS CLI

Um die SES Amazon-Benachrichtigungsattribute für eine Liste von Identitäten abzurufen

Im folgenden Beispiel wird der `get-identity-notification-attributes` Befehl verwendet, um die SES Amazon-Benachrichtigungsattribute für eine Liste von Identitäten abzurufen:

```
aws ses get-identity-notification-attributes --
identities "user1@example.com" "user2@example.com"
```

Ausgabe:

```
{
  "NotificationAttributes": {
    "user1@example.com": {
      "ForwardingEnabled": false,
      "ComplaintTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
      "BounceTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
      "DeliveryTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic"
    },
    "user2@example.com": {
      "ForwardingEnabled": true
    }
  }
}
```

Dieser Befehl gibt den Status der Weiterleitung von E-Mail-Feedback und gegebenenfalls die Amazon-Ressourcennamen (ARNs) der SNS Amazon-Themen zurück, an die Bounce-, Beschwerde- und Lieferbenachrichtigungen gesendet werden.

Wenn Sie diesen Befehl mit einer Identität aufrufen, die Sie noch nie zur Überprüfung eingereicht haben, wird diese Identität nicht in der Ausgabe angezeigt.

Weitere Informationen zu Benachrichtigungen finden Sie unter [Using Notifications With Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [GetIdentityNotificationAttributes](#) unter AWS CLI Befehlsreferenz.

get-identity-verification-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-identity-verification-attributes`.

AWS CLI

Um den SES Amazon-Bestätigungsstatus für eine Liste von Identitäten abzurufen

Im folgenden Beispiel wird der `get-identity-verification-attributes` Befehl verwendet, um den SES Amazon-Bestätigungsstatus für eine Liste von Identitäten abzurufen:


```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

Ausgabe:

```
{
  "VerificationAttributes": {
    "user1@example.com": {
      "VerificationStatus": "Success"
    },
    "user2@example.com": {
      "VerificationStatus": "Pending"
    }
  }
}
```

Wenn Sie diesen Befehl mit einer Identität aufrufen, die Sie noch nie zur Überprüfung eingereicht haben, wird diese Identität nicht in der Ausgabe angezeigt.

Weitere Informationen zu verifizierten Identitäten finden Sie unter [Verifying Email Addresses and Domains in Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [GetIdentityVerificationAttributes](#) in der AWS CLI Befehlsreferenz.

get-send-quota

Das folgende Codebeispiel zeigt die Verwendung `get-send-quota`.

AWS CLI

So erhalten Sie Ihre SES Amazon-Sendelimits

Im folgenden Beispiel wird der `get-send-quota` Befehl verwendet, um Ihre SES Amazon-Sendelimits zurückzugeben:

```
aws ses get-send-quota
```

Ausgabe:

```
{
  "Max24HourSend": 200.0,
  "SentLast24Hours": 1.0,
```

```
"MaxSendRate": 1.0
}
```

Max24 HourSend ist Ihr Versandkontingent, das ist die maximale Anzahl von E-Mails, die Sie in einem Zeitraum von 24 Stunden versenden können. Die Sendequote bezieht sich auf einen gleitenden Zeitraum. Jedes Mal, wenn Sie versuchen, eine E-Mail zu senden, SES überprüft Amazon, wie viele E-Mails Sie in den letzten 24 Stunden gesendet haben. Solange die Gesamtzahl der von Ihnen gesendeten E-Mails unter Ihrer Quote liegt, wird Ihre Sendeanforderung akzeptiert und Ihre E-Mail versendet.

SentLast24 Stunden ist die Anzahl der E-Mails, die Sie in den letzten 24 Stunden gesendet haben.

MaxSendRate ist die maximale Anzahl von E-Mails, die Sie pro Sekunde versenden können.

Beachten Sie, dass Sendelimits auf der Anzahl der Empfänger, nicht der Anzahl der Nachrichten basieren. Beispielsweise zählt eine E-Mail mit 10 Empfängern bei Ihrem Sendekontingent als 10.

Weitere Informationen finden Sie unter Verwaltung Ihrer SES Amazon-Sendelimits im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [GetSendQuota](#) unter AWS CLI Befehlsreferenz.

get-send-statistics

Das folgende Codebeispiel zeigt die Verwendung `get-send-statistics`.

AWS CLI

Um Ihre SES Amazon-Versandstatistiken abzurufen

Im folgenden Beispiel wird der `get-send-statistics` Befehl verwendet, um Ihre SES Amazon-Versandstatistiken zurückzugeben

```
aws ses get-send-statistics
```

Ausgabe:

```
{
  "SendDataPoints": [
    {
      "Complaints": 0,
```

```

    "Timestamp": "2013-06-12T19:32:00Z",
    "DeliveryAttempts": 2,
    "Bounces": 0,
    "Rejects": 0
  },
  {
    "Complaints": 0,
    "Timestamp": "2013-06-12T00:47:00Z",
    "DeliveryAttempts": 1,
    "Bounces": 0,
    "Rejects": 0
  }
]
}

```

Das Ergebnis ist eine Liste von Datenpunkten, die die Sendeaktivitäten der letzten zwei Wochen repräsentieren. Jeder Datenpunkt in der Liste enthält Statistiken für ein 15-Minuten-Intervall.

In diesem Beispiel gibt es nur zwei Datenpunkte, da die einzigen E-Mails, die der Benutzer in den letzten zwei Wochen gesendet hat, innerhalb von zwei 15-Minuten-Intervallen fielen.

Weitere Informationen finden Sie unter Überwachung Ihrer SES Amazon-Nutzungsstatistiken im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [GetSendStatistics](#) unter AWS CLI Befehlsreferenz.

list-identities

Das folgende Codebeispiel zeigt die Verwendung `list-identities`.

AWS CLI

Um alle Identitäten (E-Mail-Adressen und Domains) für ein bestimmtes AWS Konto aufzulisten

Im folgenden Beispiel wird der `list-identities` Befehl verwendet, um alle Identitäten aufzulisten, die zur Überprüfung bei Amazon SES eingereicht wurden:

```
aws ses list-identities
```

Ausgabe:

```
{
  "Identities": [
```

```
    "user@example.com",  
    "example.com"  
  ]  
}
```

Die zurückgegebene Liste enthält alle Identitäten unabhängig vom Überprüfungsstatus (verifiziert, Überprüfung ausstehend, fehlgeschlagen usw.).

In diesem Beispiel werden E-Mail-Adressen und Domains zurückgegeben, weil wir den Parameter `identity-type` nicht angegeben haben.

Weitere Informationen zur Überprüfung finden Sie unter [Verifizieren von E-Mail-Adressen und Domains bei Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [ListIdentities](#) in der AWS CLI Befehlsreferenz.

send-email

Das folgende Codebeispiel zeigt die Verwendung `send-email`.

AWS CLI

Um eine formatierte E-Mail mit Amazon zu senden SES

Im folgenden Beispiel wird der `send-email`-Befehl verwendet, um eine formatierte E-Mail zu senden:

```
aws ses send-email --from sender@example.com --destination file://destination.json  
--message file://message.json
```

Ausgabe:

```
{  
  "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"  
}
```

Das Ziel und die Nachricht sind JSON-Datenstrukturen, die in JSON-Dateien im aktuellen Verzeichnis gespeichert sind. Es handelt sich dabei um die folgenden Dateien:

`destination.json`:

```
{
```

```
"ToAddresses": ["recipient1@example.com", "recipient2@example.com"],
"CcAddresses": ["recipient3@example.com"],
"BccAddresses": []
}
```

message.json:

```
{
  "Subject": {
    "Data": "Test email sent using the AWS CLI",
    "Charset": "UTF-8"
  },
  "Body": {
    "Text": {
      "Data": "This is the message body in text format.",
      "Charset": "UTF-8"
    },
    "Html": {
      "Data": "This message body contains HTML formatting. It can, for example,
contain links like this one: <a class=\"ulink\" href=\"http://docs.aws.amazon.com/
ses/latest/DeveloperGuide\" target=\"_blank\">Amazon SES Developer Guide</a>.",
      "Charset": "UTF-8"
    }
  }
}
```

Ersetzen Sie die Absender- und Empfänger-E-Mail-Adressen durch die Adressen, die Sie verwenden möchten. Beachten Sie, dass die E-Mail-Adresse des Absenders bei Amazon verifiziert werden muss. Bis Ihnen Produktionszugriff auf Amazon SES gewährt wird, müssen Sie auch die E-Mail-Adresse jedes Empfängers verifizieren, es sei denn, der Empfänger ist der SES Amazon-Postfachsimulator. Weitere Informationen zur Überprüfung finden Sie unter [Verifizieren von E-Mail-Adressen und Domains bei Amazon SES](#) im Amazon Simple Email Service Developer Guide.

Die Nachrichten-ID in der Ausgabe gibt an, dass der Aufruf von `send-email` erfolgreich war.

Wenn Sie die E-Mail nicht erhalten, überprüfen Sie Ihr Junk-Postfach.

Weitere Informationen zum Senden formatierter E-Mails finden Sie unter [Senden formatierter E-Mails über Amazon SES API](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [SendEmail](#) in der AWS CLI Befehlsreferenz.

send-raw-email

Das folgende Codebeispiel zeigt die Verwendung send-raw-email.

AWS CLI

Um eine Roh-E-Mail mit Amazon zu versenden SES

Im folgenden Beispiel wird der send-raw-email Befehl verwendet, um eine E-Mail mit einem TXT Anhang zu senden:

```
aws ses send-raw-email --raw-message file://message.json
```

Ausgabe:

```
{
  "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

Die Rohnachricht ist eine JSON Datenstruktur, die in einer Datei mit dem Namen message.json im aktuellen Verzeichnis gespeichert ist. Sie enthält Folgendes:

```
{
  "Data": "From: sender@example.com\nTo: recipient@example.com\nSubject: Test email sent using the AWS CLI (contains an attachment)\nMIME-Version: 1.0\nContent-type: Multipart/Mixed; boundary=\"NextPart\"\n\n--NextPart\nContent-Type: text/plain\n\nThis is the message body.\n\n--NextPart\nContent-Type: text/plain;\nContent-Disposition: attachment; filename=\"attachment.txt\"\n\nThis is the text in the attachment.\n\n--NextPart--"
}
```

Wie Sie sehen können, ist „Data“ eine lange Zeichenfolge, die den gesamten E-Mail-Rohinhalt im MIME Format enthält, einschließlich eines Anhangs namens attachment.txt.

Ersetzen Sie sender@example.com und recipient@example.com durch die Adressen, die Sie verwenden möchten. Beachten Sie, dass die E-Mail-Adresse des Absenders bei Amazon verifiziert werden muss. Bis Ihnen Produktionszugriff auf Amazon gewährt wird, müssen Sie auch die E-Mail-Adresse des Empfängers verifizieren, es sei denn, der Empfänger ist der SES Amazon-Postfachsimulator. Weitere Informationen zur Überprüfung finden Sie unter Verifizieren von E-Mail-Adressen und Domains bei Amazon SES im Amazon Simple Email Service Developer Guide.

Die Nachrichten-ID in der Ausgabe gibt an, dass der Aufruf von `send-raw-email` war.

Wenn Sie die E-Mail nicht erhalten, überprüfen Sie Ihr Junk-Postfach.

Weitere Informationen zum Senden von Roh-E-Mails finden Sie unter [Senden von Roh-E-Mails über Amazon SES API](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [SendRawEmail](#) in der AWS CLI Befehlsreferenz.

set-identity-dkim-enabled

Das folgende Codebeispiel zeigt die Verwendung `set-identity-dkim-enabled`.

AWS CLI

Um Easy DKIM für eine von Amazon SES verifizierte Identität zu aktivieren oder zu deaktivieren

Im folgenden Beispiel wird der `set-identity-dkim-enabled` Befehl zur Deaktivierung DKIM für eine verifizierte E-Mail-Adresse verwendet:

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Weitere Informationen zu Easy DKIM finden Sie unter [Easy DKIM in Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [SetIdentityDkimEnabled](#) unter AWS CLI Befehlsreferenz.

set-identity-feedback-forwarding-enabled

Das folgende Codebeispiel zeigt die Verwendung `set-identity-feedback-forwarding-enabled`.

AWS CLI

So aktivieren oder deaktivieren Sie die Feedback-Weiterleitung von Bounce- und Beschwerde-Mails für eine von Amazon SES verifizierte Identität

Im folgenden Beispiel wird der `set-identity-feedback-forwarding-enabled` Befehl verwendet, um einer verifizierten E-Mail-Adresse den Empfang von Benachrichtigungen über unberechtigte Anfragen und Beschwerden per E-Mail zu ermöglichen:

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --forwarding-enabled
```

Sie müssen Benachrichtigungen über Absagen und Beschwerden entweder über Amazon SNS oder über die Feedback-Weiterleitung per E-Mail erhalten. Sie können die Weiterleitung von E-Mail-Feedback also nur deaktivieren, wenn Sie ein SNS Amazon-Thema sowohl für Bounce- als auch für Beschwerdebenachrichtigungen auswählen.

Weitere Informationen zu Benachrichtigungen finden Sie unter [Using Notifications With Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [SetIdentityFeedbackForwardingEnabled](#) unter AWS CLI Befehlsreferenz.

set-identity-notification-topic

Das folgende Codebeispiel zeigt die Verwendung `set-identity-notification-topic`.

AWS CLI

Um das SNS Amazon-Thema festzulegen, zu dem Amazon SES Benachrichtigungen über Rücksendungen, Beschwerden und/oder Lieferungen für eine verifizierte Identität veröffentlicht

Im folgenden Beispiel wird der `set-identity-notification-topic` Befehl verwendet, um das SNS Amazon-Thema anzugeben, an das eine verifizierte E-Mail-Adresse Bounce-Benachrichtigungen erhalten soll:

```
aws ses set-identity-notification-topic --identity user@example.com --notification-type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

Weitere Informationen zu Benachrichtigungen finden Sie unter [Using Notifications With Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie [SetIdentityNotificationTopic](#) unter AWS CLI Befehlsreferenz.

verify-domain-dkim

Das folgende Codebeispiel zeigt die Verwendung `verify-domain-dkim`.

AWS CLI

Um die DKIM Tokens einer verifizierten Domain für das DKIM Signieren bei Amazon zu generieren SES

Im folgenden Beispiel wird der `verify-domain-dkim` Befehl verwendet, um DKIM Token für eine Domain zu generieren, die bei Amazon verifiziert wurde SES:

```
aws ses verify-domain-dkim --domain example.com
```

Ausgabe:

```
{
  "DkimTokens": [
    "EXAMPLEq76owjnks3lnluwg65scbemvw",
    "EXAMPLEi3dnsj67hstzaj673klariwx2",
    "EXAMPLEwfbtcukvimehexktdtaz6naj"
  ]
}
```

Für die Einrichtung müssen Sie die zurückgegebenen DKIM Token verwenden DKIM, um die DNS Einstellungen Ihrer Domain mit CNAME Datensätzen zu aktualisieren, die auf von Amazon gehostete DKIM öffentliche Schlüssel verweisen SES. Weitere Informationen finden Sie unter Easy DKIM in Amazon SES im Amazon Simple Email Service Developer Guide.

- API Einheiten finden Sie [VerifyDomainDkim](#) in der AWS CLI Befehlsreferenz.

`verify-domain-identity`

Das folgende Codebeispiel zeigt die Verwendung `verify-domain-identity`.

AWS CLI

Um eine Domain bei Amazon zu verifizieren SES

Im folgenden Beispiel wird der `verify-domain-identity`-Befehl verwendet, um eine Domain zu verifizieren:

```
aws ses verify-domain-identity --domain example.com
```

Ausgabe:

```
{
  "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wmvjuEXAMPLE"
}
```

Um die Domainverifizierung abzuschließen, müssen Sie den DNS Einstellungen Ihrer Domain einen TXT Datensatz mit dem zurückgegebenen Verifizierungstoken hinzufügen. Weitere Informationen finden Sie unter Verifying Domains in Amazon SES im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie unter [VerifyDomainIdentity AWS CLI](#) Befehlsreferenz.

verify-email-identity

Das folgende Codebeispiel zeigt die Verwendung `verify-email-identity`.

AWS CLI

Um eine E-Mail-Adresse bei Amazon zu verifizieren SES

Im folgenden Beispiel wird der `verify-email-identity`-Befehl verwendet, um eine E-Mail-Adresse zu verifizieren:

```
aws ses verify-email-identity --email-address user@example.com
```

Bevor Sie eine E-Mail über Amazon versenden können SES, müssen Sie die Adresse oder Domain verifizieren, von der aus Sie die E-Mail versenden, um nachzuweisen, dass Sie der Eigentümer der E-Mail sind. Wenn Sie noch keinen Produktionszugriff haben, müssen Sie auch alle E-Mail-Adressen verifizieren, an die Sie E-Mails senden, mit Ausnahme der vom SES Amazon-Postfachsimulator bereitgestellten E-Mail-Adressen.

Nach dem Anruf `verify-email-identity` erhält die E-Mail-Adresse eine Bestätigungs-E-Mail. Der Benutzer muss auf den Link in der E-Mail klicken, um den Verifizierungsvorgang abzuschließen.

Weitere Informationen finden Sie unter Verifizieren von E-Mail-Adressen in Amazon SES im Amazon Simple Email Service Developer Guide.

- API-Einheiten finden Sie unter [VerifyEmailIdentity AWS CLI](#) Befehlsreferenz.

Shield-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Shield Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-drt-log-bucket

Das folgende Codebeispiel zeigt die Verwendung `associate-drt-log-bucket`.

AWS CLI

Um den Zugriff auf einen Amazon S3 S3-Bucket DRT zu autorisieren

Das folgende `associate-drt-log-bucket` Beispiel erstellt eine Zuordnung zwischen dem DRT und dem angegebenen S3-Bucket. Dadurch kann der DRT im Namen des Kontos auf den Bucket zugreifen. :

```
aws shield associate-drt-log-bucket \  
  --log-bucket flow-logs-for-website-lb
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Autorisieren des DDoS Response-Teams](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [AssociateDrtLogBucket](#) in der AWS CLI Befehlsreferenz.

associate-drt-role

Das folgende Codebeispiel zeigt die Verwendung `associate-drt-role`.

AWS CLI

Um die DRT zu autorisieren, potenzielle Angriffe in Ihrem Namen abzuwehren

Im folgenden `associate-drt-role` Beispiel wird eine Zuordnung zwischen der DRT und der angegebenen Rolle hergestellt. Sie DRT können die Rolle verwenden, um auf das Konto zuzugreifen und es zu verwalten.

```
aws shield associate-drt-role \  
  --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Autorisieren des DDoS Response-Teams](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [AssociateDrtRole](#) in der AWS CLI Befehlsreferenz.

create-protection

Das folgende Codebeispiel zeigt die Verwendung `create-protection`.

AWS CLI

Um AWS Shield Advanced-Schutz für eine einzelne AWS Ressource zu aktivieren

Das folgende `create-protection` Beispiel aktiviert Shield Advanced-Schutz für die angegebene AWS CloudFront Distribution.

```
aws shield create-protection \  
  --name "Protection for CloudFront distribution" \  
  --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FX0WY8
```

Ausgabe:

```
{  
  "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
}
```

Weitere Informationen finden Sie unter [Spezifizieren Sie Ihre zu schützenden Ressourcen](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [CreateProtection](#) in der AWS CLI Befehlsreferenz.

create-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-subscription`.

AWS CLI

Um AWS Shield Advanced-Schutz für ein Konto zu aktivieren

Im folgenden `create-subscription` Beispiel wird Shield Advanced-Schutz für das Konto aktiviert.

```
aws shield create-subscription
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit AWS Shield Advanced](#) im AWS Shield Advanced-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateSubscription](#) in der AWS CLI Befehlsreferenz.

delete-protection

Das folgende Codebeispiel zeigt die Verwendung `delete-protection`.

AWS CLI

Um AWS Shield Advanced-Schutz von einer AWS Ressource zu entfernen

Im folgenden `delete-protection` Beispiel wird der angegebene AWS Shield Advanced-Schutz entfernt.

```
aws shield delete-protection \  
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Entfernen von AWS Shield Advanced aus einer AWS Ressource](#) im AWS Shield Advanced-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteProtection](#) in der AWS CLI Befehlsreferenz.

describe-attack

Das folgende Codebeispiel zeigt die Verwendung `describe-attack`.

AWS CLI

Um eine detaillierte Beschreibung eines Angriffs abzurufen

Im folgenden `describe-attack` Beispiel werden Details zum DDoS Angriff mit der angegebenen Angriffs-ID angezeigt. Sie können einen Angriff ausführen, IDs indem Sie den `list-attacks` Befehl ausführen.

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Ausgabe:

```
{
  "Attack": {
    "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "ResourceArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/testElb",
    "SubResources": [
      {
        "Type": "IP",
        "Id": "192.0.2.2",
        "AttackVectors": [
          {
            "VectorType": "SYN_FLOOD",
            "VectorCounters": [
              {
                "Name": "SYN_FLOOD_BPS",
                "Max": 982184.0,
                "Average": 982184.0,
                "Sum": 11786208.0,
                "N": 12,
                "Unit": "BPS"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
    }
  ],
  "Counters": []
},
{
  "Type": "IP",
  "Id": "192.0.2.3",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 9821840.0,
          "N": 10,
          "Unit": "BPS"
        }
      ]
    }
  ]
},
  ],
  "Counters": []
},
{
  "Type": "IP",
  "Id": "192.0.2.4",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 7857472.0,
          "N": 8,
          "Unit": "BPS"
        }
      ]
    }
  ]
},
  ],
  "Counters": []
},
},
```

```
{
  "Type": "IP",
  "Id": "192.0.2.5",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 1964368.0,
          "N": 2,
          "Unit": "BPS"
        }
      ]
    }
  ],
  "Counters": []
},
{
  "Type": "IP",
  "Id": "2001:DB8::bcde:4321:8765:0:0",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 1964368.0,
          "N": 2,
          "Unit": "BPS"
        }
      ]
    }
  ],
  "Counters": []
},
{
  "Type": "IP",
  "Id": "192.0.2.6",
  "AttackVectors": [
```



```
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 1964368.0,
          "N": 2,
          "Unit": "BPS"
        }
      ]
    },
    "Counters": []
  ],
  "StartTime": 1576024927.457,
  "EndTime": 1576025647.457,
  "AttackCounters": [],
  "AttackProperties": [
    {
      "AttackLayer": "NETWORK",
      "AttackPropertyIdentifier": "SOURCE_IP_ADDRESS",
      "TopContributors": [
        {
          "Name": "198.51.100.5",
          "Value": 2024475682
        },
        {
          "Name": "198.51.100.8",
          "Value": 1311380863
        },
        {
          "Name": "203.0.113.4",
          "Value": 900599855
        },
        {
          "Name": "198.51.100.4",
          "Value": 769417366
        },
        {
          "Name": "203.1.113.13",
          "Value": 757992847
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "Unit": "BYTES",
  "Total": 92773354841
},
{
  "AttackLayer": "NETWORK",
  "AttackPropertyIdentifier": "SOURCE_COUNTRY",
  "TopContributors": [
    {
      "Name": "United States",
      "Value": 80938161764
    },
    {
      "Name": "Brazil",
      "Value": 9929864330
    },
    {
      "Name": "Netherlands",
      "Value": 1635009446
    },
    {
      "Name": "Mexico",
      "Value": 144832971
    },
    {
      "Name": "Japan",
      "Value": 45369000
    }
  ],
  "Unit": "BYTES",
  "Total": 92773354841
},
{
  "AttackLayer": "NETWORK",
  "AttackPropertyIdentifier": "SOURCE_ASN",
  "TopContributors": [
    {
      "Name": "12345",
      "Value": 74953625841
    },
    {
      "Name": "12346",
      "Value": 4440087595
    }
  ]
}
```

```

        },
        {
            "Name": "12347",
            "Value": 1635009446
        },
        {
            "Name": "12348",
            "Value": 1221230000
        },
        {
            "Name": "12349",
            "Value": 1199425294
        }
    ],
    "Unit": "BYTES",
    "Total": 92755479921
}
],
"Mitigations": []
}
}

```

Weitere Informationen finden Sie unter Reviews [DDoSIncidents](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DescribeAttack](#) in der AWS CLI Befehlsreferenz.

describe-drt-access

Das folgende Codebeispiel zeigt die Verwendung `describe-drt-access`.

AWS CLI

Um eine Beschreibung der Autorisierungen abzurufen, muss DRT er Angriffe in Ihrem Namen abwehren.

Im folgenden `describe-drt-access` Beispiel werden die Rollen- und S3-Bucket-Autorisierungen abgerufen, sodass DRT über die er verfügt, sodass er in Ihrem Namen auf potenzielle Angriffe reagieren kann.

```
aws shield describe-drt-access
```

Ausgabe:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
  "LogBucketList": [
    "flow-logs-for-website-lb"
  ]
}
```

Weitere Informationen finden Sie unter [Autorisieren des DDoS Response-Teams](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DescribeDrtAccess](#) in der AWS CLI Befehlsreferenz.

describe-emergency-contact-settings

Das folgende Codebeispiel zeigt die Verwendung `describe-emergency-contact-settings`.

AWS CLI

Um Notfall-E-Mail-Adressen abzurufen, die Sie bei der gespeichert haben DRT

Im folgenden `describe-emergency-contact-settings` Beispiel werden die E-Mail-Adressen abgerufen, die in der Datei DRT für das Konto gespeichert sind. Dies sind die Adressen, an die sie sich wenden DRT sollten, wenn sie auf einen vermuteten Angriff reagieren.

```
aws shield describe-emergency-contact-settings
```

Ausgabe:

```
{
  "EmergencyContactList": [
    {
      "EmailAddress": "ops@example.com"
    },
    {
      "EmailAddress": "ddos-notifications@example.com"
    }
  ]
}
```

Weitere Informationen finden Sie unter So funktioniert AWS Shield < <https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html> > im Shield Advanced Developer Guide.AWS

- Einzelheiten API finden Sie in der Befehlsreferenz. [DescribeEmergencyContactSettingsAWS CLI](#)

describe-protection

Das folgende Codebeispiel zeigt die Verwendung `describe-protection`.

AWS CLI

Um die Details für einen AWS Shield Advanced-Schutz abzurufen

Im folgenden `describe-protection` Beispiel werden Details zum Shield Advanced-Schutz mit der angegebenen ID angezeigt. Sie können den Schutz erhalten, IDs indem Sie den `list-protections` Befehl ausführen.

```
aws shield describe-protection \
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "Protection": {
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "1.2.3.4",
    "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/
eipalloc-0ac1537af40742a6d"
  }
}
```

Weitere Informationen finden [Sie unter Spezifizieren Sie Ihre zu schützenden Ressourcen](#) im AWS Shield Advanced Developer Guide.

- API Einzelheiten finden Sie [DescribeProtection](#) in der AWS CLI Befehlsreferenz.

describe-subscription

Das folgende Codebeispiel zeigt die Verwendung `describe-subscription`.

AWS CLI

Um die Details des AWS Shield Advanced-Schutzes für das Konto abzurufen

Das folgende `describe-subscription` Beispiel zeigt Details zum Shield Advanced-Schutz, der für das Konto bereitgestellt wurde. :

```
aws shield describe-subscription
```

Ausgabe:

```
{
  "Subscription": {
    "StartTime": 1534368978.0,
    "EndTime": 1597613778.0,
    "TimeCommitmentInSeconds": 63244800,
    "AutoRenew": "ENABLED",
    "Limits": [
      {
        "Type": "GLOBAL_ACCELERATOR",
        "Max": 1000
      },
      {
        "Type": "ROUTE53_HOSTED_ZONE",
        "Max": 1000
      },
      {
        "Type": "CF_DISTRIBUTION",
        "Max": 1000
      },
      {
        "Type": "ELB_LOAD_BALANCER",
        "Max": 1000
      },
      {
        "Type": "EC2_ELASTIC_IP_ALLOCATION",
        "Max": 1000
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [So funktioniert AWS Shield](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DescribeSubscription](#) in der AWS CLI Befehlsreferenz.

disassociate-drt-log-bucket

Das folgende Codebeispiel zeigt die Verwendung `disassociate-drt-log-bucket`.

AWS CLI

Um die Autorisierung für den DRT-Zugriff auf einen Amazon S3 S3-Bucket in Ihrem Namen zu entfernen

Das folgende `disassociate-drt-log-bucket` Beispiel entfernt die Zuordnung zwischen dem DRT und dem angegebenen S3-Bucket. Nach Abschluss dieses Befehls DRT können sie im Namen des Kontos nicht mehr auf den Bucket zugreifen.

```
aws shield disassociate-drt-log-bucket \  
  --log-bucket flow-logs-for-website-lb
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Autorisieren des DDoS Response-Teams](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DisassociateDrtLogBucket](#) in der AWS CLI Befehlsreferenz.

disassociate-drt-role

Das folgende Codebeispiel zeigt die Verwendung `disassociate-drt-role`.

AWS CLI

Um die Autorisierung zu entfernen DRT, um potenzielle Angriffe in Ihrem Namen abzuwehren

Im folgenden `disassociate-drt-role` Beispiel wird die Verknüpfung zwischen dem DRT und dem Konto aufgehoben. Nach diesem Aufruf DRT können sie nicht mehr auf Ihr Konto zugreifen oder es verwalten.

```
aws shield disassociate-drt-role
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Autorisieren des DDoS Response-Teams](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DisassociateDrtRole](#) in der AWS CLI Befehlsreferenz.

get-subscription-state

Das folgende Codebeispiel zeigt die Verwendung `get-subscription-state`.

AWS CLI

Um den aktuellen Status des AWS Shield Advanced-Abonnements des Kontos abzurufen

Im folgenden `get-subscription-state` Beispiel wird der Status des Shield Advanced-Schutzes für das Konto abgerufen.

```
aws shield get-subscription-state
```

Ausgabe:

```
{
  "SubscriptionState": "ACTIVE"
}
```

Weitere Informationen finden Sie unter [So funktioniert AWS Shield](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [GetSubscriptionState](#) in der AWS CLI Befehlsreferenz.

list-attacks

Das folgende Codebeispiel zeigt die Verwendung `list-attacks`.

AWS CLI

Um Angriffszusammenfassungen von AWS Shield Advanced abzurufen

Im folgenden `list-attacks` Beispiel werden Zusammenfassungen der Angriffe für die angegebene AWS CloudFront Verteilung im angegebenen Zeitraum abgerufen. Die Antwort

beinhaltet einen AngriffIDs, den Sie dem `describe-attack` Befehl zur Verfügung stellen können, um detaillierte Informationen zu einem Angriff zu erhalten.

```
aws shield list-attacks \  
  --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXMP22ZVFAOR \  
  --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

Ausgabe:

```
{  
  "AttackSummaries": [  
    {  
      "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/  
E1PXMP22ZVFAOR",  
      "StartTime": 1529280000.0,  
      "EndTime": 1529449200.0,  
      "AttackVectors": [  
        {  
          "VectorType": "SYN_FLOOD"  
        }  
      ]  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter Reviews [DDoSIncidents](#) im AWS Shield Advanced Developer Guide.

- APIEinzelheiten finden Sie [ListAttacks](#) in der AWS CLI Befehlsreferenz.

list-protections

Das folgende Codebeispiel zeigt die Verwendung `list-protections`.

AWS CLI

Um Schutzübersichten von AWS Shield Advanced abzurufen

Im folgenden `list-protections` Beispiel werden Zusammenfassungen der Schutzmaßnahmen abgerufen, die für das Konto aktiviert sind.

```
aws shield list-protections
```

Ausgabe:

```
{
  "Protections": [
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Name": "Protection for CloudFront distribution",
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FX0WY8"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Spezifizieren Sie Ihre zu schützenden Ressourcen](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [ListProtections](#) in der AWS CLI Befehlsreferenz.

update-emergency-contact-settings

Das folgende Codebeispiel zeigt die Verwendung `update-emergency-contact-settings`.

AWS CLI

Um die Notfall-E-Mail-Adressen zu definieren, die in der Datei gespeichert sind DRT

Im folgenden `update-emergency-contact-settings` Beispiel werden zwei E-Mail-Adressen definiert, an die sie sich wenden DRT sollen, wenn sie auf einen vermuteten Angriff reagieren.

```
aws shield update-emergency-contact-settings \
  --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-
notifications@example.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [So funktioniert AWS Shield](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [UpdateEmergencyContactSettings](#) in der AWS CLI Befehlsreferenz.

update-subscription

Das folgende Codebeispiel zeigt die Verwendung `update-subscription`.

AWS CLI

Um das AWS Shield Advanced-Abonnement des Kontos zu ändern

Das folgende `update-subscription` Beispiel aktiviert die automatische Verlängerung des AWS Shield Advanced-Abonnements für das Konto.

```
aws shield update-subscription \  
  --auto-renew ENABLED
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [So funktioniert AWS Shield](#) im AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [UpdateSubscription](#) in der AWS CLI Befehlsreferenz.

Beispiele für Unterzeichner mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Signer Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

cancel-signing-profile

Das folgende Codebeispiel zeigt die Verwendung `cancel-signing-profile`.

AWS CLI

Um ein Signaturprofil zu löschen

Im folgenden `cancel-signing-profile` Beispiel wird ein vorhandenes Signaturprofil aus AWS Signer entfernt.

```
aws signer cancel-signing-profile \  
  --profile-name MyProfile1
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [CancelSigningProfile AWS CLI](#) Befehlsreferenz.

describe-signing-job

Das folgende Codebeispiel zeigt die Verwendung `describe-signing-job`.

AWS CLI

Um Details zu einem Signaturauftrag anzuzeigen

Im folgenden `describe-signing-job` Beispiel werden Details zum angegebenen Signaturauftrag angezeigt.

```
aws signer describe-signing-job \  
  --job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

Ausgabe:

```
{  
  "status": "Succeeded",  
  "completedAt": 1568412037,  
  "platformId": "AmazonFreeRTOS-Default",  
  "signingMaterial": {  
    "certificateArn": "arn:aws:acm:us-  
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"  
  },  
  "statusReason": "Signing Succeeded",  
  "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",  
  "source": {
```

```

    "s3": {
      "version": "PNyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4",
      "bucketName": "signer-source",
      "key": "MyCode.rb"
    }
  },
  "profileName": "MyProfile2",
  "signedObject": {
    "s3": {
      "bucketName": "signer-destination",
      "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
    }
  },
  "requestedBy": "arn:aws:iam::123456789012:user/maria",
  "createdAt": 1568412036
}

```

- API-Einheiten finden Sie [DescribeSigningJob](#) unter AWS CLI Befehlsreferenz.

get-signing-platform

Das folgende Codebeispiel zeigt die Verwendung `get-signing-platform`.

AWS CLI

Um Details zu einer Signaturplattform anzuzeigen

Im folgenden `get-signing-platform` Beispiel werden Details zur angegebenen Signierplattform angezeigt.

```

aws signer get-signing-platform \
  --platform-id AmazonFreeRTOS-TI-CC3220SF

```

Ausgabe:

```

{
  "category": "AWS",
  "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
  "target": "SHA1-RSA-TISHA1",
  "platformId": "AmazonFreeRTOS-TI-CC3220SF",
  "signingConfiguration": {
    "encryptionAlgorithmOptions": {

```

```

        "defaultValue": "RSA",
        "allowedValues": [
            "RSA"
        ]
    },
    "hashAlgorithmOptions": {
        "defaultValue": "SHA1",
        "allowedValues": [
            "SHA1"
        ]
    }
},
"maxSizeInMB": 16,
"partner": "AmazonFreeRTOS",
"signingImageFormat": {
    "defaultFormat": "JSONEmbedded",
    "supportedFormats": [
        "JSONEmbedded"
    ]
}
}
}

```

- API-Einheiten finden Sie [GetSigningPlatform](#) unter AWS CLI Befehlsreferenz.

get-signing-profile

Das folgende Codebeispiel zeigt die Verwendung `get-signing-profile`.

AWS CLI

Um Details zu einem Signaturprofil anzuzeigen

Im folgenden `get-signing-profile` Beispiel werden Details zum angegebenen Signaturprofil angezeigt.

```
aws signer get-signing-profile \
  --profile-name MyProfile3
```

Ausgabe:

```
{
  "platformId": "AmazonFreeRTOS-TI-CC3220SF",
```

```
"profileName": "MyProfile3",
"status": "Active",
"signingMaterial": {
  "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
}
}
```

- API-Einheiten finden Sie [GetSigningProfile](#) unter AWS CLI Befehlsreferenz.

list-signing-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-signing-jobs`.

AWS CLI

Um alle Signieraufträge aufzulisten

Im folgenden `list-signing-jobs` Beispiel werden Details zu allen Signieraufträgen für das Konto angezeigt.

```
aws signer list-signing-jobs
```

In diesem Beispiel werden zwei Aufträge zurückgegeben, einer erfolgreich und einer fehlgeschlagen.

```
{
  "jobs": [
    {
      "status": "Succeeded",
      "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
      },
      "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
      "source": {
        "s3": {
          "version": "PNyFaUTgsQh5ZdMccoCe6pT1g0pgB_M4",
          "bucketName": "signer-source",
          "key": "MyCode.rb"
        }
      }
    }
  ]
}
```

```

    },
    "signedObject": {
      "s3": {
        "bucketName": "signer-destination",
        "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
      }
    },
    "createdAt": 1568412036
  },
  {
    "status": "Failed",
    "source": {
      "s3": {
        "version": "PNyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4",
        "bucketName": "signer-source",
        "key": "MyOtherCode.rb"
      }
    },
    "signingMaterial": {
      "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
    },
    "createdAt": 1568402690,
    "jobId": "74d9825e-22fc-4a0d-b962-0123456789abc"
  }
]
}

```

- API-Einheiten finden Sie [ListSigningJobs](#) in der AWS CLI Befehlsreferenz.

list-signing-platforms

Das folgende Codebeispiel zeigt die Verwendung `list-signing-platforms`.

AWS CLI

Um alle Signierplattformen aufzulisten

Im folgenden `list-signing-platforms` Beispiel werden Details zu allen verfügbaren Signierplattformen angezeigt.

```
aws signer list-signing-platforms
```


Ausgabe:

```
{
  "platforms": [
    {
      "category": "AWS",
      "displayName": "AWS IoT Device Management SHA256-ECDSA ",
      "target": "SHA256-ECDSA",
      "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
      "signingConfiguration": {
        "encryptionAlgorithmOptions": {
          "defaultValue": "ECDSA",
          "allowedValues": [
            "ECDSA"
          ]
        },
        "hashAlgorithmOptions": {
          "defaultValue": "SHA256",
          "allowedValues": [
            "SHA256"
          ]
        }
      },
      "maxSizeInMB": 2048,
      "partner": "AWSIoTDeviceManagement",
      "signingImageFormat": {
        "defaultFormat": "JSONDetached",
        "supportedFormats": [
          "JSONDetached"
        ]
      }
    },
    {
      "category": "AWS",
      "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
      "target": "SHA1-RSA-TISHA1",
      "platformId": "AmazonFreeRTOS-TI-CC3220SF",
      "signingConfiguration": {
        "encryptionAlgorithmOptions": {
          "defaultValue": "RSA",
          "allowedValues": [
            "RSA"
          ]
        }
      }
    }
  ]
}
```

```
        "hashAlgorithmOptions": {
            "defaultValue": "SHA1",
            "allowedValues": [
                "SHA1"
            ]
        },
    },
    "maxSizeInMB": 16,
    "partner": "AmazonFreeRTOS",
    "signingImageFormat": {
        "defaultFormat": "JSONEmbedded",
        "supportedFormats": [
            "JSONEmbedded"
        ]
    }
},
{
    "category": "AWS",
    "displayName": "Amazon FreeRTOS SHA256-ECDSA",
    "target": "SHA256-ECDSA",
    "platformId": "AmazonFreeRTOS-Default",
    "signingConfiguration": {
        "encryptionAlgorithmOptions": {
            "defaultValue": "ECDSA",
            "allowedValues": [
                "ECDSA"
            ]
        },
        "hashAlgorithmOptions": {
            "defaultValue": "SHA256",
            "allowedValues": [
                "SHA256"
            ]
        }
    },
    "maxSizeInMB": 16,
    "partner": "AmazonFreeRTOS",
    "signingImageFormat": {
        "defaultFormat": "JSONEmbedded",
        "supportedFormats": [
            "JSONEmbedded"
        ]
    }
}
```

```
]
}
```

- API-Einheiten finden Sie [ListSigningPlatforms](#) unter AWS CLI Befehlsreferenz.

list-signing-profiles

Das folgende Codebeispiel zeigt die Verwendung `list-signing-profiles`.

AWS CLI

Um alle Signaturprofile aufzulisten

Im folgenden `list-signing-profiles` Beispiel werden Details zu allen Signaturprofilen für das Konto angezeigt.

```
aws signer list-signing-profiles
```

Ausgabe:

```
{
  "profiles": [
    {
      "platformId": "AmazonFreeRTOS-TI-CC3220SF",
      "profileName": "MyProfile4",
      "status": "Active",
      "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
      }
    },
    {
      "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
      "profileName": "MyProfile5",
      "status": "Active",
      "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
      }
    }
  ]
}
```

- API-Einheiten finden Sie [ListSigningProfiles](#) unter AWS CLI Befehlsreferenz.

put-signing-profile

Das folgende Codebeispiel zeigt die Verwendung `put-signing-profile`.

AWS CLI

Um ein Signaturprofil zu erstellen

Im folgenden `put-signing-profile` Beispiel wird ein Signaturprofil unter Verwendung des angegebenen Zertifikats und der angegebenen Plattform erstellt.

```
aws signer put-signing-profile \  
  --profile-name MyProfile6 \  
  --signing-material certificateArn=arn:aws:acm:us-west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc \  
  --platform AmazonFreeRTOS-TI-CC3220SF
```

Ausgabe:

```
{  
  "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"  
}
```

- API-Einheiten finden Sie [PutSigningProfile](#) unter AWS CLI Befehlsreferenz.

start-signing-job

Das folgende Codebeispiel zeigt die Verwendung `start-signing-job`.

AWS CLI

Um einen Signierjob zu starten

Im folgenden `start-signing-job` Beispiel wird ein Signaturauftrag für den Code gestartet, der sich in der angegebenen Quelle befindet. Es verwendet das angegebene Profil für die Signierung und platziert den signierten Code im angegebenen Ziel.

```
aws signer start-signing-job \  
  --code-location s3://code-bucket/signing-job \  
  --profile-name MyProfile6 \  
  --target s3://code-bucket/signing-job
```

```
--source 's3={bucketName=signer-  
source,key=MyCode.rb,version=PMyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4}' \  
--destination 's3={bucketName=signer-destination,prefix=signed-}' \  
--profile-name MyProfile7
```

Die Ausgabe ist die ID des Signaturauftrags.

```
{  
  "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"  
}
```

- API-Einheiten finden Sie [StartSigningJob](#) in der AWS CLI Befehlsreferenz.

Schneeball-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Snowball Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-snowball-usage

Das folgende Codebeispiel zeigt die Verwendung `get-snowball-usage`.

AWS CLI

Um Informationen über das Snowball-Servicelimit für Ihr Konto zu erhalten

Im folgenden `get-snowball-usage` Beispiel werden Informationen zum Snowball-Servicelimit für Ihr Konto sowie zur Anzahl der Snowballs angezeigt, die Ihr Konto verwendet.

```
aws snowball get-snowball-usage
```

Ausgabe:

```
{
  "SnowballLimit": 1,
  "SnowballsInUse": 0
}
```

FORWeitere Informationen finden Sie unter [AWS Snowball Edge Limits](#) im AWS Snowball Developer Guide.

- APIEinzelheiten finden Sie [GetSnowballUsage](#) in der AWS CLI Befehlsreferenz.

list-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-jobs`.

AWS CLI

Um die aktuellen Snowball-Jobs in Ihrem Konto aufzulisten

Im folgenden `list-jobs` Beispiel wird eine Reihe von `JobListEntry` Objekten angezeigt. In diesem Beispiel wird ein einzelner Job aufgeführt.

```
aws snowball list-jobs
```

Ausgabe:

```
{
  "JobListEntries": [
    {
      "CreationDate": 2016-09-27T14:50Z,
      "Description": "Important Photos 2016-08-11",
      "IsMaster": TRUE,
      "JobId": "ABCd1e324fe-022f-488e-a98b-3b0566063db1",
      "JobState": "Complete",
      "JobType": "IMPORT",
      "SnowballType": "EDGE"
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Jobs für AWS Snowball Edge-Geräte](#) im AWS Snowball Developer Guide.

- API-Einheiten finden Sie unter [ListJobs AWS CLI](#) Befehlsreferenz.

SNS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren SNS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Dienstes oder in Kombination mit anderen aufrufen AWS-Services.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)
- [Szenarien](#)

Aktionen

add-permission

Das folgende Codebeispiel zeigt die Verwendung `add-permission`.

AWS CLI

Um einem Thema eine Berechtigung hinzuzufügen

Im folgenden `add-permission` Beispiel wird dem AWS Konto die Berechtigung hinzugefügt, die `Publish` Aktion mit dem angegebenen Thema unter AWS Konto `987654321098` zu verwenden `123456789012`.

```
aws sns add-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission \  
  --aws-account-id 987654321098 \  
  --action-name Publish
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AddPermission](#) unter AWS CLI Befehlsreferenz.

check-if-phone-number-is-opted-out

Das folgende Codebeispiel zeigt die Verwendung `check-if-phone-number-is-opted-out`.

AWS CLI

Um die SMS-Nachricht zu überprüfen, melden Sie sich für eine Telefonnummer ab.

Im folgenden `check-if-phone-number-is-opted-out`-Beispiel wird geprüft, ob für die angegebene Telefonnummer der Empfang von SMS-Nachrichten vom AWS-Girokonto deaktiviert ist.

```
aws sns check-if-phone-number-is-opted-out \  
  --phone-number +1555550100
```

Ausgabe:

```
{  
  "isOptedOut": false  
}
```

- API-Einheiten finden Sie [CheckIfPhoneNumberIsOptedOut](#) unter AWS CLI Befehlsreferenz.

confirm-subscription

Das folgende Codebeispiel zeigt die Verwendung `confirm-subscription`.

AWS CLI

So bestätigen Sie ein Abonnement.

Mit dem folgenden `confirm-subscription` Befehl wird der Bestätigungsvorgang abgeschlossen, der gestartet wurde, als Sie ein SNS Thema mit dem Namen `my-topic` abonniert haben. Der `--token`-Parameter stammt aus der Bestätigungsnachricht, die an den im Abonnementaufruf angegebenen Benachrichtigungsendpunkt gesendet wurde.

```
aws sns confirm-subscription \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --  
  token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5a
```

Ausgabe:

```
{  
  "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"  
}
```

- API-Einheiten finden Sie [ConfirmSubscription](#) in der AWS CLI Befehlsreferenz.

create-platform-application

Das folgende Codebeispiel zeigt die Verwendung `create-platform-application`.

AWS CLI

Um eine Plattformanwendung zu erstellen

Im folgenden `create-platform-application` Beispiel wird eine Google Firebase-Plattformanwendung mit den angegebenen Plattformanmeldeinformationen erstellt.

```
aws sns create-platform-application \  
  --name MyApplication \  
  --platform GCM \  
  --attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

Ausgabe:

```
{  
  "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/  
MyApplication"
```

```
}
```

- API-Einheiten finden Sie [CreatePlatformApplication](#) in der AWS CLI Befehlsreferenz.

create-topic

Das folgende Codebeispiel zeigt die Verwendung `create-topic`.

AWS CLI

So erstellen Sie ein SNS-Thema

Im folgenden `create-topic` Beispiel wird ein SNS Thema mit dem Namen `my-topic` erstellt.

```
aws sns create-topic \  
  --name my-topic
```

Ausgabe:

```
{  
  "ResponseMetadata": {  
    "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"  
  },  
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"  
}
```

Weitere Informationen finden Sie unter [Verwenden der AWS Befehlszeilenschnittstelle mit Amazon SQS und Amazon SNS](#) im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle.

- API-Einheiten finden Sie [CreateTopic](#) in der AWS CLI Befehlsreferenz.

delete-endpoint

Das folgende Codebeispiel zeigt die Verwendung `delete-endpoint`.

AWS CLI

Um einen Plattformanwendungsendpunkt zu löschen

Im folgenden `delete-endpoint` Beispiel wird der angegebene Plattformanwendungsendpunkt gelöscht.

```
aws sns delete-endpoint \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/MyApplication/12345678-abcd-9012-efgh-345678901234
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteEndpoint](#) in der AWS CLI Befehlsreferenz.

delete-platform-application

Das folgende Codebeispiel zeigt die Verwendung `delete-platform-application`.

AWS CLI

Um eine Plattformanwendung zu löschen

Im folgenden `delete-platform-application` Beispiel wird die angegebene Plattformanwendung gelöscht.

```
aws sns delete-platform-application \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/MyApplication
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeletePlatformApplication](#) in der AWS CLI Befehlsreferenz.

delete-topic

Das folgende Codebeispiel zeigt die Verwendung `delete-topic`.

AWS CLI

Um ein SNS Thema zu löschen

Im folgenden `delete-topic` Beispiel wird das angegebene SNS Thema gelöscht.

```
aws sns delete-topic \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteTopic](#) in der AWS CLI Befehlsreferenz.

get-endpoint-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-endpoint-attributes`.

AWS CLI

Um die Endpunktattribute von Plattformanwendungen aufzulisten

Im folgenden `get-endpoint-attributes` Beispiel werden die Attribute für den angegebenen Plattformanwendungsendpunkt aufgeführt.

```
aws sns get-endpoint-attributes \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234
```

Ausgabe:

```
{  
  "Attributes": {  
    "Enabled": "true",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

- API-Einheiten finden Sie [GetEndpointAttributes](#) in der AWS CLI Befehlsreferenz.

get-platform-application-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-platform-application-attributes`.

AWS CLI

Um die Attribute der Plattformanwendung aufzulisten

Im folgenden `get-platform-application-attributes` Beispiel werden die Attribute für die angegebene Plattformanwendung aufgeführt.

```
aws sns get-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/  
MyApplication
```

Ausgabe:

```
{
  "Attributes": {
    "Enabled": "true",
    "SuccessFeedbackSampleRate": "100"
  }
}
```

- API-Einheiten finden Sie [GetPlatformApplicationAttributes](#) in der AWS CLI Befehlsreferenz.

get-sms-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-sms-attributes`.

AWS CLI

Um die SMS Standard-Nachrichtenattribute aufzulisten

Das folgende `get-sms-attributes` Beispiel listet die Standardattribute für das Senden von SMS Nachrichten auf.

```
aws sns get-sms-attributes
```

Ausgabe:

```
{
  "attributes": {
    "DefaultSenderId": "MyName"
  }
}
```

- API-Einheiten finden Sie unter [GetSMSAttributes](#) in der AWS CLI Befehlsreferenz.

get-subscription-attributes

Das folgende Codebeispiel zeigt, wie man `get-subscription-attributes` verwendet.

AWS CLI

Um Abonnementattribute für ein Thema abzurufen

Im Folgenden `get-subscription-attributes` werden die Attribute des angegebenen Abonnements angezeigt. Sie können das `subscription-arn` aus der Ausgabe des `list-subscriptions` Befehls abrufen.

```
aws sns get-subscription-attributes \  
  --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

Ausgabe:

```
{  
  "Attributes": {  
    "Endpoint": "my-email@example.com",  
    "Protocol": "email",  
    "RawMessageDelivery": "false",  
    "ConfirmationWasAuthenticated": "false",  
    "Owner": "123456789012",  
    "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f",  
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"  
  }  
}
```

- API-Einheiten finden Sie [GetSubscriptionAttributes](#) in der AWS CLI Befehlsreferenz.

get-topic-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-topic-attributes`.

AWS CLI

So rufen Sie die Attribute eines Themas ab

Im folgenden `get-topic-attributes`-Beispiel werden die Attribute für das angegebene Thema angezeigt.

```
aws sns get-topic-attributes \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Ausgabe:

```
{
  "Attributes": {
    "SubscriptionsConfirmed": "1",
    "DisplayName": "my-topic",
    "SubscriptionsDeleted": "0",
    "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":{\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries\":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear\"},\"disableSubscriptionOverrides\":false}}",
    "Owner": "123456789012",
    "Policy": "{\"Version\":\"2008-10-17\",\"Id\":\"__default_policy_ID\",\"Statement\": [{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\": [\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:DeleteTopic\",\"SNS:GetTopicAttributes\",\"SNS:Publish\",\"SNS:RemovePermission\",\"SNS:AddPermission\",\"SNS:SetTopicAttributes\"],\"Resource\":\"arn:aws:sns:us-west-2:123456789012:my-topic\",\"Condition\": {\"StringEquals\":{\"AWS:SourceOwner\":\"0123456789012\"}}}]}",
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
    "SubscriptionsPending": "0"
  }
}
```

- API-Einheiten finden Sie [GetTopicAttributes](#) in der AWS CLI Befehlsreferenz.

list-endpoints-by-platform-application

Das folgende Codebeispiel zeigt die Verwendung `list-endpoints-by-platform-application`.

AWS CLI

Um die Endpunkte für eine Plattformanwendung aufzulisten

Im folgenden `list-endpoints-by-platform-application` Beispiel werden die Endpunkte und Endpunktattribute für die angegebene Plattformanwendung aufgeführt.

```
aws sns list-endpoints-by-platform-application \
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication
```

Ausgabe:

```
{
```

```
"Endpoints": [  
  {  
    "Attributes": {  
      "Token": "EXAMPLE12345...",  
      "Enabled": "true"  
    },  
    "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234"  
  }  
]
```

- API-Einheiten finden Sie [ListEndpointsByPlatformApplication](#) in der AWS CLI Befehlsreferenz.

list-phone-numbers-opted-out

Das folgende Codebeispiel zeigt die Verwendung `list-phone-numbers-opted-out`.

AWS CLI

Um die Abmeldungen für SMS Nachrichten aufzulisten

Das folgende `list-phone-numbers-opted-out` Beispiel listet die Telefonnummern auf, denen der Empfang SMS von Nachrichten verweigert wurde.

```
aws sns list-phone-numbers-opted-out
```

Ausgabe:

```
{  
  "phoneNumbers": [  
    "+15555550100"  
  ]  
}
```

- API-Einheiten finden Sie [ListPhoneNumbersOptedOut](#) in der AWS CLI Befehlsreferenz.

list-platform-applications

Das folgende Codebeispiel zeigt die Verwendung `list-platform-applications`.

AWS CLI

Um Plattformanwendungen aufzulisten

Das folgende `list-platform-applications` Beispiel listet die Plattformanwendungen für ADM und aufMPNS.

```
aws sns list-platform-applications
```

Ausgabe:

```
{
  "PlatformApplications": [
    {
      "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/ADM/MyApplication",
      "Attributes": {
        "SuccessFeedbackSampleRate": "100",
        "Enabled": "true"
      }
    },
    {
      "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/MPNS/MyOtherApplication",
      "Attributes": {
        "SuccessFeedbackSampleRate": "100",
        "Enabled": "true"
      }
    }
  ]
}
```

- API-Einheiten finden Sie [ListPlatformApplications](#) in der AWS CLI Befehlsreferenz.

list-subscriptions-by-topic

Das folgende Codebeispiel zeigt die Verwendung `list-subscriptions-by-topic`.

AWS CLI

Um die Abonnements aufzulisten, die einem Thema zugeordnet sind

Im Folgenden wird eine Liste der SNS Abonnements `list-subscriptions-by-topic` abgerufen, die dem angegebenen Thema zugeordnet sind.

```
aws sns list-subscriptions-by-topic \  
--topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Ausgabe:

```
{  
  "Subscriptions": [  
    {  
      "Owner": "123456789012",  
      "Endpoint": "my-email@example.com",  
      "Protocol": "email",  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",  
      "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListSubscriptionsByTopic](#) in der AWS CLI Befehlsreferenz.

list-subscriptions

Das folgende Codebeispiel zeigt die Verwendung `list-subscriptions`.

AWS CLI

Um Ihre SNS Abonnements aufzulisten

Im folgenden `list-subscriptions` Beispiel wird eine Liste der SNS Abonnements in Ihrem AWS Konto angezeigt.

```
aws sns list-subscriptions
```

Ausgabe:

```
{  
  "Subscriptions": [  
    {
```

```

    "Owner": "123456789012",
    "Endpoint": "my-email@example.com",
    "Protocol": "email",
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
    "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
  }
]
}

```

- API-Einheiten finden Sie [ListSubscriptions](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um Tags für ein Thema aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für das angegebene SNS Amazon-Thema auf.

```

aws sns list-tags-for-resource \
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic

```

Ausgabe:

```

{
  "Tags": [
    {
      "Key": "Team",
      "Value": "Alpha"
    }
  ]
}

```

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-topics

Das folgende Codebeispiel zeigt die Verwendung `list-topics`.

AWS CLI

Um Ihre SNS Themen aufzulisten

Das folgende `list-topics` Beispiel listet alle SNS Themen in Ihrem AWS Konto auf.

```
aws sns list-topics
```

Ausgabe:

```
{
  "Topics": [
    {
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
    }
  ]
}
```

- API-Einheiten finden Sie [ListTopics](#) in der AWS CLI Befehlsreferenz.

`opt-in-phone-number`

Das folgende Codebeispiel zeigt die Verwendung `opt-in-phone-number`.

AWS CLI

Um sich für SMS Nachrichten anzumelden

Im folgenden `opt-in-phone-number` Beispiel wird die angegebene Telefonnummer für den Empfang SMS von Nachrichten aktiviert.

```
aws sns opt-in-phone-number \
  --phone-number +15555550100
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [OptInPhoneNumber AWS CLI](#) Befehlsreferenz.

`publish`

Das folgende Codebeispiel zeigt die Verwendung `publish`.

AWS CLI

Beispiel 1: So veröffentlichen Sie eine Nachricht für ein Thema

Im folgenden `publish` Beispiel wird die angegebene Nachricht zum angegebenen SNS Thema veröffentlicht. Die Nachricht stammt aus einer Textdatei, in der Sie Zeilenumbrüche einfügen können.

```
aws sns publish \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \  
  --message file://message.txt
```

Inhalt von `message.txt`:

```
Hello World  
Second Line
```

Ausgabe:

```
{  
  "MessageId": "123a45b6-7890-12c3-45d6-111122223333"  
}
```

Beispiel 2: Um eine SMS Nachricht an eine Telefonnummer zu veröffentlichen

Im folgenden `publish`-Beispiel wird Nachricht `Hello world!` an Telefonnummer `+1-555-555-0100` veröffentlicht.

```
aws sns publish \  
  --message "Hello world!" \  
  --phone-number +1-555-555-0100
```

Ausgabe:

```
{  
  "MessageId": "123a45b6-7890-12c3-45d6-333322221111"  
}
```

- API-Einheiten finden Sie unter In der AWS CLI Befehlsreferenz [veröffentlichen](#).

put-data-protection-policy

Das folgende Codebeispiel zeigt die Verwendung `put-data-protection-policy`.

AWS CLI

Um eine Datenschutzrichtlinie festzulegen

Beispiel 1: Um Publishern die Veröffentlichung von Nachrichten mit zu verbieten
CreditCardNumber

Im folgenden `put-data-protection-policy` Beispiel wird Verlegern das Veröffentlichen von Nachrichten mit CreditCardNumber verweigert.

```
aws sns put-data-protection-policy \  
  --resource-arn arn:aws:sns:us-east-1:123456789012:mytopic \  
  --data-protection-policy '{"Name":"data_protection_policy","Description\  
": "Example data protection policy","Version":"2021-06-01","Statement\  
": [{"DataDirection":"Inbound","Principal":["*"],"DataIdentifier":\  
["arn:aws:dataprotection::aws:data-identifier/CreditCardNumber"],"Operation":\  
{"Deny":{}}}]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um Parameter aus einer Datei zu laden

Im Folgenden werden Parameter aus einer Datei `put-data-protection-policy` geladen.

```
aws sns put-data-protection-policy \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --data-protection-policy file://policy.json
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutDataProtectionPolicy](#) in der AWS CLI Befehlsreferenz.

remove-permission

Das folgende Codebeispiel zeigt die Verwendung `remove-permission`.

AWS CLI

Um eine Berechtigung aus einem Thema zu entfernen

Im folgenden `remove-permission` Beispiel wird die Berechtigung `Publish-Permission` für das angegebene Thema entfernt.

```
aws sns remove-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [RemovePermission](#) unter AWS CLI Befehlsreferenz.

set-endpoint-attributes

Das folgende Codebeispiel zeigt die Verwendung `set-endpoint-attributes`.

AWS CLI

Um Endpunktattribute festzulegen

Im folgenden `set-endpoint-attributes` Beispiel wird der angegebene Plattformanwendungsendpoint deaktiviert.

```
aws sns set-endpoint-attributes \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/MyApplication/12345678-abcd-9012-efgh-345678901234 \  
  --attributes Enabled=false
```

Ausgabe:

```
{  
  "Attributes": {  
    "Enabled": "false",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

- API-Einheiten finden Sie [SetEndpointAttributes](#) in der AWS CLI Befehlsreferenz.

set-platform-application-attributes

Das folgende Codebeispiel zeigt die Verwendung `set-platform-application-attributes`.

AWS CLI

So legen Sie Plattformanwendungsattribute fest

Im folgenden `set-platform-application-attributes` Beispiel wird das `EventDeliveryFailure` Attribut für die angegebene Plattformanwendung auf das ARN des angegebenen SNS Amazon-Themas gesetzt.

```
aws sns set-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/MyApplication \  
  --attributes EventDeliveryFailure=arn:aws:sns:us-west-2:123456789012:AnotherTopic
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SetPlatformApplicationAttributes](#) unter AWS CLI Befehlsreferenz.

set-sms-attributes

Das folgende Codebeispiel zeigt die Verwendung `set-sms-attributes`.

AWS CLI

Um SMS Nachrichtenattribute festzulegen

Im folgenden `set-sms-attributes` Beispiel wird die standardmäßige Absender-ID für SMS Nachrichten auf `festgelegtMyName` festgelegt.

```
aws sns set-sms-attributes \  
  --attributes DefaultSenderId=MyName
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [SetSMSAttributes](#) in der AWS CLI Befehlsreferenz.

set-subscription-attributes

Das folgende Codebeispiel zeigt, wie man `set-subscription-attributes` verwendet.

AWS CLI

So legen Sie Abonnementattribute fest

Im folgenden `set-subscription-attributes` Beispiel wird das `RawMessageDelivery` Attribut auf ein SQS Abonnement festgelegt.

```
aws sns set-subscription-attributes \  
  --subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
  --attribute-name RawMessageDelivery \  
  --attribute-value true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Im folgenden `set-subscription-attributes` Beispiel wird einem SQS Abonnement ein `FilterPolicy` Attribut zugewiesen.

```
aws sns set-subscription-attributes \  
  --subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
  --attribute-name FilterPolicy \  
  --attribute-value "{ \"anyMandatoryKey\": [\"any\", \"of\", \"these\"] }"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Im folgenden `set-subscription-attributes` Beispiel wird das `FilterPolicy` Attribut aus einem SQS Abonnement entfernt.

```
aws sns set-subscription-attributes \  
  --subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
  --attribute-name FilterPolicy \  
  --attribute-value "{}"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SetSubscriptionAttributes](#) in der AWS CLI Befehlsreferenz.

set-topic-attributes

Das folgende Codebeispiel zeigt die Verwendung `set-topic-attributes`.

AWS CLI

So legen Sie ein Attribut für ein Thema fest

Im folgenden `set-topic-attributes`-Beispiel wird das `DisplayName`-Attribute für das angegebene Thema festgelegt.

```
aws sns set-topic-attributes \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --attribute-name DisplayName \  
  --attribute-value MyTopicDisplayName
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [SetTopicAttributes](#) in der AWS CLI Befehlsreferenz.

subscribe

Das folgende Codebeispiel zeigt die Verwendung `subscribe`.

AWS CLI

So abonnieren Sie ein Thema

Der folgende `subscribe`-Befehl abonniert das angegebene Thema mit einer E-Mail-Adresse.

```
aws sns subscribe \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --protocol email \  
  --notification-endpoint my-email@example.com
```

Ausgabe:

```
{  
  "SubscriptionArn": "pending confirmation"  
}
```

- API-Einheiten finden Sie unter [Abonnieren](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

So fügen Sie einem Thema ein Tag hinzu

Das folgende `tag-resource` Beispiel fügt dem angegebenen SNS Amazon-Thema ein Metadaten-Tag hinzu.

```
aws sns tag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tags Key=Team,Value=Alpha
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

unsubscribe

Das folgende Codebeispiel zeigt die Verwendung `unsubscribe`.

AWS CLI

So melden Sie sich von einem Thema ab

Im folgenden `unsubscribe`-Beispiel wird das angegebene Abonnement aus einem Thema gelöscht.

```
aws sns unsubscribe \  
  --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [Unsubscribe](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um ein Tag aus einem Thema zu entfernen

Im folgenden `untag-resource` Beispiel werden alle Tags mit den angegebenen Schlüsseln aus dem angegebenen SNS Amazon-Thema entfernt.

```
aws sns untag-resource \  
  --resource-arn arn:aws:sns:us-west-2:0123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

```
--resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
--tag-keys Team
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

Szenarien

Erstellen eines Plattformendpunkts für Push-Benachrichtigungen

Das folgende Codebeispiel zeigt, wie ein Plattformendpunkt für SNS Amazon-Push-Benachrichtigungen erstellt wird.

AWS CLI

So erstellen Sie ein Plattformanwendungsendpunkt

Im folgenden `create-platform-endpoint`-Beispiel wird mithilfe des angegebenen Tokens ein Endpunkt für die angegebene Plattformanwendung erstellt.

```
aws sns create-platform-endpoint \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/  
MyApplication \  
  --token EXAMPLE12345...
```

Ausgabe:

```
{  
  "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234"  
}
```

SQS Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie AWS Command Line Interface mit Amazon verwenden SQS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-permission

Das folgende Codebeispiel zeigt die Verwendung `add-permission`.

AWS CLI

Um einer Warteschlange eine Berechtigung hinzuzufügen

In diesem Beispiel kann das angegebene AWS Konto Nachrichten an die angegebene Warteschlange senden.

Befehl:

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue --aws-account-ids 12345EXAMPLE --actions SendMessage
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [AddPermission](#) in der AWS CLI Befehlsreferenz.

cancel-message-move-task

Das folgende Codebeispiel zeigt, wie man es benutzt `cancel-message-move-task`.

AWS CLI

Um eine Aufgabe zum Verschieben einer Nachricht abubrechen

Im folgenden `cancel-message-move-task` Beispiel wird die angegebene Aufgabe zum Verschieben von Nachrichten abgebrochen.

```
aws sqs cancel-message-move-task \  
  --task-handle AQEB6nR4...HzLvZQ==
```

Ausgabe:

```
{  
  "ApproximateNumberOfMessagesMoved": 102  
}
```

Weitere Informationen finden Sie unter [SQSAPIAmazon-Berechtigungen: Aktionen und Ressourcenreferenz](#) im Entwicklerhandbuch.

- API-Einheiten finden Sie [CancelMessageMoveTask](#) unter AWS CLI Befehlsreferenz.

change-message-visibility-batch

Das folgende Codebeispiel zeigt, wie man es benutzt `change-message-visibility-batch`.

AWS CLI

Um die Timeout-Sichtbarkeit mehrerer Nachrichten als Batch zu ändern

In diesem Beispiel wird die Timeout-Sichtbarkeit der beiden angegebenen Nachrichten auf 10 Stunden (10 Stunden x 60 Minuten x 60 Sekunden) geändert.

Befehl:

```
aws sqs change-message-visibility-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://change-message-visibility-batch.json
```

Eingabedatei (.json): `change-message-visibility-batch`

```
[  
  {  
    "Id": "FirstMessage",  
    "ReceiptHandle": "AQEBhz2q...Jf3kaw==",  
    "VisibilityTimeout": 36000  
  },  
  {  
    "Id": "SecondMessage",
```

```
    "ReceiptHandle": "AQEBkTUH...HifSnw==",
    "VisibilityTimeout": 36000
  }
]
```

Ausgabe:

```
{
  "Successful": [
    {
      "Id": "SecondMessage"
    },
    {
      "Id": "FirstMessage"
    }
  ]
}
```

- API-Einheiten finden Sie [ChangeMessageVisibilityBatch](#) in der AWS CLI Befehlsreferenz.

change-message-visibility

Das folgende Codebeispiel zeigt, wie man es benutzt `change-message-visibility`.

AWS CLI

Um die Timeout-Sichtbarkeit einer Nachricht zu ändern

In diesem Beispiel wird die Timeout-Sichtbarkeit der angegebenen Nachricht auf 10 Stunden (10 Stunden x 60 Minuten x 60 Sekunden) geändert.

Befehl:

```
aws sqs change-message-visibility --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --visibility-timeout 36000
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie unter [ChangeMessageVisibility AWS CLI](#) Befehlsreferenz.

create-queue

Das folgende Codebeispiel zeigt, wie man es benutzt `create-queue`.

AWS CLI

So erstellen Sie eine Warteschlange

In diesem Beispiel wird eine Warteschlange mit dem angegebenen Namen erstellt, die Aufbewahrungsdauer für Nachrichten auf 3 Tage (3 Tage * 24 Stunden * 60 Minuten * 60 Sekunden) festgelegt und die Warteschlange für unzustellbare Nachrichten der Warteschlange auf die angegebene Warteschlange mit einer maximalen Empfangszahl von 1.000 Nachrichten festgelegt.

Befehl:

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

Eingabedatei (create-queue.json):

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\", \"maxReceiveCount\": \"1000\"}\",
  "MessageRetentionPeriod": "259200"
}
```

Ausgabe:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- API-Einheiten finden Sie in der Befehlsreferenz [CreateQueue](#).AWS CLI

delete-message-batch

Das folgende Codebeispiel zeigt die Verwendung `delete-message-batch`.

AWS CLI

Um mehrere Nachrichten als Stapel zu löschen

In diesem Beispiel werden die angegebenen Nachrichten gelöscht.

Befehl:

```
aws sqs delete-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

Eingabedatei (delete-message-batch.json):

```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEB1mg1...Z4GuLw=="
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

Ausgabe:

```
{
  "Successful": [
    {
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

- API-Einheiten finden Sie [DeleteMessageBatch](#) in der AWS CLI Befehlsreferenz.

delete-message

Das folgende Codebeispiel zeigt die Verwendung `delete-message`.

AWS CLI

Um eine Nachricht zu löschen

In diesem Beispiel wird die angegebene Nachricht gelöscht.

Befehl:

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [DeleteMessage](#) in der AWS CLI Befehlsreferenz.

delete-queue

Das folgende Codebeispiel zeigt die Verwendung `delete-queue`.

AWS CLI

So löschen Sie eine Warteschlange

In diesem Beispiel wird die angegebene Warteschlange gelöscht.

Befehl:

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [DeleteQueue](#) in der AWS CLI Befehlsreferenz.

get-queue-attributes

Das folgende Codebeispiel zeigt die Verwendung `get-queue-attributes`.

AWS CLI

Um die Attribute einer Warteschlange abzurufen

In diesem Beispiel werden alle Attribute der angegebenen Warteschlange abgerufen.

Befehl:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All
```

Ausgabe:

```
{
  "Attributes": {
    "ApproximateNumberOfMessagesNotVisible": "0",
    "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\", \"maxReceiveCount\":1000}\",
    "MessageRetentionPeriod": "345600",
    "ApproximateNumberOfMessagesDelayed": "0",
    "MaximumMessageSize": "262144",
    "CreatedTimestamp": "1442426968",
    "ApproximateNumberOfMessages": "0",
    "ReceiveMessageWaitTimeSeconds": "0",
    "DelaySeconds": "0",
    "VisibilityTimeout": "30",
    "LastModifiedTimestamp": "1442426968",
    "QueueArn": "arn:aws:sqs:us-east-1:80398EXAMPLE:MyNewQueue"
  }
}
```

In diesem Beispiel werden nur die maximalen Nachrichtengrößen- und Sichtbarkeits-Timeout-Attribute der angegebenen Warteschlange abgerufen.

Befehl:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-names MaximumMessageSize VisibilityTimeout
```

Ausgabe:

```
{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}
```

- API-Einheiten finden Sie unter [GetQueueAttributes AWS CLI](#) Befehlsreferenz.

get-queue-url

Das folgende Codebeispiel zeigt die Verwendung `get-queue-url`.

AWS CLI

Um eine Warteschlange zu erhalten URL

In diesem Beispiel werden die angegebenen Warteschlangen abgerufen URL.

Befehl:

```
aws sqs get-queue-url --queue-name MyQueue
```

Ausgabe:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- API-Einheiten finden Sie [GetQueueUrl](#) in der AWS CLI Befehlsreferenz.

list-dead-letter-source-queues

Das folgende Codebeispiel zeigt die Verwendung `list-dead-letter-source-queues`.

AWS CLI

Um Warteschlangen mit unerlaubten Briefen aufzulisten

In diesem Beispiel werden die Warteschlangen aufgeführt, die der angegebenen Warteschlange für unzustellbare Nachrichten zugeordnet sind.

Befehl:

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

Ausgabe:

```
{
  "queueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- API-Einheiten finden Sie unter [ListDeadLetterSourceQueues AWS CLI-Befehlsreferenz](#).

list-message-move-tasks

Das folgende Codebeispiel zeigt die Verwendung `list-message-move-tasks`.

AWS CLI

Um die Nachricht aufzulisten, verschieben Sie Aufgaben

Das folgende `list-message-move-tasks` Beispiel listet die beiden letzten Aufgaben zum Verschieben von Nachrichten in der angegebenen Warteschlange auf.

```
aws sqs list-message-move-tasks \
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \
  --max-results 2
```

Ausgabe:

```
{
  "Results": [
    {
      "TaskHandle": "AQEB6nR4...HzlvZQ==",
      "Status": "RUNNING",
      "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",
      "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",
      "MaxNumberOfMessagesPerSecond": 50,
    }
  ]
}
```

```

    "ApproximateNumberOfMessagesMoved": 203,
    "ApproximateNumberOfMessagesToMove": 30,
    "StartedTimestamp": 1442428276921
  },
  {
    "Status": "COMPLETED",
    "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",
    "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",
    "ApproximateNumberOfMessagesMoved": 29,
    "ApproximateNumberOfMessagesToMove": 0,
    "StartedTimestamp": 1342428272093
  }
]
}

```

Weitere Informationen finden Sie unter [SQSAPIAmazon-Berechtigungen: Aktionen und Ressourcenreferenz](#) im Entwicklerhandbuch.

- API-Einheiten finden Sie [ListMessageMoveTasks](#) unter AWS CLI Befehlsreferenz.

list-queue-tags

Das folgende Codebeispiel zeigt die Verwendung `list-queue-tags`.

AWS CLI

Um alle Kostenzuweisungs-Tags für eine Warteschlange aufzulisten

Im folgenden `list-queue-tags` Beispiel werden alle Kostenzuweisungs-Tags angezeigt, die der angegebenen Warteschlange zugeordnet sind.

```

aws sqs list-queue-tags \
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue

```

Ausgabe:

```

{
  "Tags": {
    "Team": "Alpha"
  }
}

```

Weitere Informationen finden Sie unter [Listing Cost Allocation Tags](#) im Amazon Simple Queue Service Developer Guide.

- API-Einheiten finden Sie [ListQueueTags](#) unter AWS CLI Befehlsreferenz.

list-queues

Das folgende Codebeispiel zeigt die Verwendung `list-queues`.

AWS CLI

Um Warteschlangen aufzulisten

In diesem Beispiel werden alle Warteschlangen aufgelistet.

Befehl:

```
aws sqs list-queues
```

Ausgabe:

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/TestQueue1",
    "https://queue.amazonaws.com/80398EXAMPLE/TestQueue2"
  ]
}
```

In diesem Beispiel werden nur Warteschlangen aufgeführt, die mit „My“ beginnen.

Befehl:

```
aws sqs list-queues --queue-name-prefix My
```

Ausgabe:

```
{
  "QueueUrls": [
```

```
"https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",  
"https://queue.amazonaws.com/80398EXAMPLE/MyQueue",  
"https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"  
]  
}
```

- API-Einheiten finden Sie [ListQueues](#) in der AWS CLI Befehlsreferenz.

purge-queue

Das folgende Codebeispiel zeigt die Verwendung `purge-queue`.

AWS CLI

Um eine Warteschlange zu löschen

In diesem Beispiel werden alle Nachrichten in der angegebenen Warteschlange gelöscht.

Befehl:

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [PurgeQueue](#) in der AWS CLI Befehlsreferenz.

receive-message

Das folgende Codebeispiel zeigt die Verwendung `receive-message`.

AWS CLI

Um eine Nachricht zu erhalten

In diesem Beispiel werden bis zu 10 verfügbare Nachrichten empfangen und alle verfügbaren Attribute zurückgegeben.

Befehl:


```
aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All --message-attribute-names All --max-number-of-messages 10
```

Ausgabe:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIKMSNQ7EXAMPLE",
        "ApproximateReceiveCount": "5",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        },
        "City": {
          "DataType": "String",
          "StringValue": "Any City"
        }
      }
    }
  ]
}
```

In diesem Beispiel wird die nächste verfügbare Nachricht empfangen und nur die `SentTimestamp` Attribute `SenderId` und sowie das `PostalCode` Nachrichtenattribut zurückgegeben.

Befehl:

```
aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp --message-attribute-names PostalCode
```

Ausgabe:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIASZKMSNQ7EXAMPLE",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        }
      }
    }
  ]
}
```

- API-Einheiten finden Sie [ReceiveMessage](#) unter AWS CLI Befehlsreferenz.

remove-permission

Das folgende Codebeispiel zeigt die Verwendung `remove-permission`.

AWS CLI

Um eine Erlaubnis zu entfernen

In diesem Beispiel wird die Berechtigung mit der angegebenen Bezeichnung aus der angegebenen Warteschlange entfernt.

Befehl:

```
aws sqs remove-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue
```

Ausgabe:

None.

- API-Einheiten finden Sie [RemovePermission](#) in der AWS CLI Befehlsreferenz.

send-message-batch

Das folgende Codebeispiel zeigt die Verwendung von `send-message-batch`.

AWS CLI

Um mehrere Nachrichten als Batch zu senden

In diesem Beispiel werden 2 Nachrichten mit den angegebenen Nachrichtentexten, Verzögerungszeiten und Nachrichtenattributen an die angegebene Warteschlange gesendet.

Befehl:

```
aws sqs send-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

Eingabedatei (send-message-batch.json):

```
[
  {
    "Id": "FuelReport-0001-2015-09-16T140731Z",
    "MessageBody": "Fuel report for account 0001 on 2015-09-16 at 02:07:31 PM.",
    "DelaySeconds": 10,
    "MessageAttributes": {
      "SellerName": {
        "DataType": "String",
        "StringValue": "Example Store"
      },
      "City": {
        "DataType": "String",
        "StringValue": "Any City"
      },
      "Region": {
        "DataType": "String",
        "StringValue": "WA"
      }
    }
  },
]
```

```

        "PostalCode": {
            "DataType": "String",
            "StringValue": "99065"
        },
        "PricePerGallon": {
            "DataType": "Number",
            "StringValue": "1.99"
        }
    },
    {
        "Id": "FuelReport-0002-2015-09-16T140930Z",
        "MessageBody": "Fuel report for account 0002 on 2015-09-16 at 02:09:30 PM.",
        "DelaySeconds": 10,
        "MessageAttributes": {
            "SellerName": {
                "DataType": "String",
                "StringValue": "Example Fuels"
            },
            "City": {
                "DataType": "String",
                "StringValue": "North Town"
            },
            "Region": {
                "DataType": "String",
                "StringValue": "WA"
            },
            "PostalCode": {
                "DataType": "String",
                "StringValue": "99123"
            },
            "PricePerGallon": {
                "DataType": "Number",
                "StringValue": "1.87"
            }
        }
    }
]

```

Ausgabe:

```

{
  "Successful": [

```

```
{
  "MD50fMessageBody": "203c4a38...7943237e",
  "MD50fMessageAttributes": "10809b55...baf283ef",
  "Id": "FuelReport-0001-2015-09-16T140731Z",
  "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
},
{
  "MD50fMessageBody": "2cf0159a...c1980595",
  "MD50fMessageAttributes": "55623928...ae354a25",
  "Id": "FuelReport-0002-2015-09-16T140930Z",
  "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
}
]
```

- API-Einheiten finden Sie [SendMessageBatch](#) in der AWS CLI Befehlsreferenz.

send-message

Das folgende Codebeispiel zeigt die Verwendung von `send-message`.

AWS CLI

So senden Sie eine Nachricht

In diesem Beispiel wird eine Nachricht mit dem angegebenen Nachrichtentext, der angegebenen Verzögerungszeit und den Nachrichtenattributen an die angegebene Warteschlange gesendet.

Befehl:

```
aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --message-body "Information about the largest city in Any Region." --delay-seconds 10 --message-attributes file://send-message.json
```

Eingabedatei (send-message.json):

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
```

```
"DataType": "Binary",
  "BinaryValue": "Hello, World!"
},
"Population": {
  "DataType": "Number",
  "StringValue": "1250800"
}
}
```

Ausgabe:

```
{
  "MD5ofMessageBody": "51b0a325...39163aa0",
  "MD5ofMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```

- API-Einheiten finden Sie in der Befehlsreferenz. [SendMessage](#) AWS CLI

set-queue-attributes

Das folgende Codebeispiel zeigt die Verwendung `set-queue-attributes`.

AWS CLI

Um Warteschlangenattribute festzulegen

In diesem Beispiel wird für die angegebene Warteschlange eine Zustellungsverzögerung von 10 Sekunden, eine maximale Nachrichtengröße von 128 KB (128 KB x 1.024 Byte), eine Aufbewahrungszeit für Nachrichten von 3 Tagen (3 Tage x 24 Stunden * 60 Minuten x 60 Sekunden), eine Wartezeit für den Empfang von Nachrichten von 20 Sekunden und ein standardmäßiges Sichtbarkeits-Timeout von 60 Sekunden festgelegt. In diesem Beispiel wird der angegebenen Warteschlange für unzustellbare Nachrichten außerdem eine maximale Empfangszahl von 1.000 Nachrichten zugewiesen.

Befehl:

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

Eingabedatei (set-queue-attributes.json):

```
{
  "DelaySeconds": "10",
  "MaximumMessageSize": "131072",
  "MessageRetentionPeriod": "259200",
  "ReceiveMessageWaitTimeSeconds": "20",
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "VisibilityTimeout": "60"
}
```

Ausgabe:

```
None.
```

- API-Einheiten finden Sie [SetQueueAttributes](#) in der AWS CLI Befehlsreferenz.

start-message-move-task

Das folgende Codebeispiel zeigt die Verwendung `start-message-move-task`.

AWS CLI

Beispiel 1: *Um eine Aufgabe zum Verschieben von Nachrichten zu starten*

Im folgenden `start-message-move-task` Beispiel wird eine Aufgabe zum Verschieben von Nachrichten gestartet, um Nachrichten aus der angegebenen Warteschlange für unzustellbare Nachrichten in die Quellwarteschlange weiterzuleiten.

```
aws sqs start-message-move-task \
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

Ausgabe:

```
{
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="
}
```

Weitere Informationen finden Sie unter [Das ist der Thementitel](#) im Namen Ihres Leitfadens.

Beispiel 2: *Um eine Aufgabe zum Verschieben von Nachrichten mit einer maximalen Rate zu starten*

Im folgenden `start-message-move-task` Beispiel wird eine Aufgabe zum Verschieben von Nachrichten gestartet, um Nachrichten mit einer maximalen Geschwindigkeit von 50 Nachrichten pro Sekunde aus der angegebenen Warteschlange für unzustellbare Nachrichten an die angegebene Zielwarteschlange weiterzuleiten.

```
aws sqs start-message-move-task \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1 \  
  --destination-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2 \  
  --max-number-of-messages-per-second 50
```

Ausgabe:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```

Weitere Informationen finden Sie unter [SQSAPIAmazon-Berechtigungen: Aktionen und Ressourcenreferenz](#) im Entwicklerhandbuch.

- API-Einheiten finden Sie [StartMessageMoveTask](#) unter AWS CLI Befehlsreferenz.

tag-queue

Das folgende Codebeispiel zeigt die Verwendung `tag-queue`.

AWS CLI

Um einer Warteschlange Tags für die Kostenzuweisung hinzuzufügen

Im folgenden `tag-queue` Beispiel wird der angegebenen SQS Amazon-Warteschlange ein Kostenzuweisungs-Tag hinzugefügt.

```
aws sqs tag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tags Priority=Highest
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen von Kostenzuweisungs-Tags](#) im Amazon Simple Queue Service Developer Guide.

- API-Einheiten finden Sie [TagQueue](#) unter AWS CLI Befehlsreferenz.

untag-queue

Das folgende Codebeispiel zeigt die Verwendung `untag-queue`.

AWS CLI

Um Kostenverrechnungs-Tags aus einer Warteschlange zu entfernen

Im folgenden `untag-queue` Beispiel wird ein Kostenzuweisungs-Tag aus der angegebenen SQS Amazon-Warteschlange entfernt.

```
aws sqs untag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tag-keys "Priority"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Hinzufügen von Kostenzuweisungs-Tags](#) im Amazon Simple Queue Service Developer Guide.

- API-Einheiten finden Sie [UntagQueue](#) unter AWS CLI Befehlsreferenz.

Storage Gateway Gateway-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Storage Gateway Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

describe-gateway-information

Das folgende Codebeispiel zeigt die Verwendung `describe-gateway-information`.

AWS CLI

Um ein Gateway zu beschreiben

Der folgende `describe-gateway-information` Befehl gibt Metadaten über das angegebene Gateway zurück. Um anzugeben, welches Gateway beschrieben werden soll, verwenden Sie den Amazon-Ressourcennamen (ARN) des Gateways im Befehl.

Dieses Beispiel spezifiziert ein Gateway mit der ID `sgw-12A3456B` im Konto `123456789012`:

```
aws storagegateway describe-gateway-information --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Dieser Befehl gibt einen JSON Block aus, der Metadaten über das Gateway enthält, z. B. seinen Namen, seine Netzwerkschnittstellen, die konfigurierte Zeitzone und den Status (ob das Gateway läuft oder nicht).

- API-Einheiten finden Sie [DescribeGatewayInformation](#) unter AWS CLI Befehlsreferenz.

list-file-shares

Das folgende Codebeispiel zeigt die Verwendung `list-file-shares`.

AWS CLI

Um Dateifreigaben aufzulisten

Das folgende `command-name` Beispiel listet die verfügbaren Widgets in Ihrem AWS Konto auf.

```
aws storagegateway list-file-shares \--gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

Ausgabe:

```
{  
  "FileShareInfoList": [  
    ...  
  ]  
}
```

```
{
  "FileShareType": "NFS",
  "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345",
  "FileShareId": "share-2FA12345",
  "FileShareStatus": "AVAILABLE",
  "GatewayARN": "arn:aws:storagegateway:us-east-1:111122223333:gateway/
sgw-FB0AAAAA"
},
"Marker": null
}
```

Weitere Informationen finden Sie [ListFileShares](#) in der AWS Storage Gateway Service API Reference.

- API-Einheiten finden Sie [ListFileShares](#) unter AWS CLI Befehlsreferenz.

list-gateways

Das folgende Codebeispiel zeigt die Verwendung `list-gateways`.

AWS CLI

Um Gateways für ein Konto aufzulisten

Der folgende `list-gateways` Befehl listet alle für ein Konto definierten Gateways auf:

```
aws storagegateway list-gateways
```

Dieser Befehl gibt einen JSON-Block aus, der eine Liste von Gateway-Amazon-Ressourcennamen (ARNs) enthält.

- API-Einheiten finden Sie [ListGateways](#) in der AWS CLI Befehlsreferenz.

list-volumes

Das folgende Codebeispiel zeigt die Verwendung `list-volumes`.

AWS CLI

Um die für ein Gateway konfigurierten Volumes aufzulisten

Der folgende `list-volumes` Befehl gibt eine Liste der für das angegebene Gateway konfigurierten Volumes zurück. Um anzugeben, welches Gateway beschrieben werden soll, verwenden Sie den Amazon-Ressourcennamen (ARN) des Gateways im Befehl.

Dieses Beispiel spezifiziert ein Gateway mit der ID `sgw-12A3456B` im Konto `123456789012`:

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Dieser Befehl gibt einen JSON Block aus, der eine Liste von Volumes enthält, die den Typ und ARN für jedes Volume enthält.

- API-Einheiten finden Sie [ListVolumes](#) in der AWS CLI Befehlsreferenz.

refresh-cache

Das folgende Codebeispiel zeigt die Verwendung `refresh-cache`.

AWS CLI

Um den Fileshare-Cache zu aktualisieren

Im folgenden `refresh-cache` Beispiel wird der Cache für die angegebene Dateifreigabe aktualisiert.

```
aws storagegateway refresh-cache \
  --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/share-2FA12345
```

Ausgabe:

```
{
  "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/share-2FA12345",
  "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"
}
```

Weitere Informationen finden Sie [ListFileShares](#) in der AWS Storage Gateway Service API Reference.

- API-Einheiten finden Sie [RefreshCache](#) unter AWS CLI Befehlsreferenz.

AWS STS Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS STS.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

assume-role-with-saml

Das folgende Codebeispiel zeigt die Verwendung `assume-role-with-saml`.

AWS CLI

Um kurzfristige Anmeldeinformationen für eine Rolle zu erhalten, die authentifiziert wurde mit SAML

Der folgende `assume-role-with-saml` Befehl ruft eine Reihe von kurzfristigen Anmeldeinformationen für die IAM Rolle ab. `TestSaml` Die Anforderung in diesem Beispiel wird mithilfe der SAML Assertion authentifiziert, die Ihr Identitätsanbieter bei der Authentifizierung angegeben hat.

```
aws sts assume-role-with-saml \
  --role-arn arn:aws:iam::123456789012:role/TestSaml \
  --principal-arn arn:aws:iam::123456789012:saml-provider/SAML-test \
  --saml-assertion "VERYLONGENCODEDASSERTIONEXAMPLExzYW1s0kF1ZG1LbmNlPmJsYW5rPC9zYW1s0kF1ZG1LbmNlPjwv
+PHNhbw6TmFtZULEIEZvcm1hdD0idXJu0m9hc2lz0m5hbWVz0nRj0LNBTUw6Mi4w0m5hbWVpZC1mb3JtYXQ6dHJhbnM
+PHNhbw6U3ViamVjdENvbMzpcm1hdGlvbiBNZXRob2Q9InVybJpYXNpczpuYW1lc3p0YzptQU1MOjIuMDpjbTpiZWV"
```

Ausgabe:

```
{
  "Issuer": "https://integ.example.com/idp/shibboleth</Issuer",
  "AssumedRoleUser": {
    "Arn": "arn:aws:sts::123456789012:assumed-role/TestSaml",
    "AssumedRoleId": "AR0456EXAMPLE789:TestSaml"
  },
  "Credentials": {
    "AccessKeyId": "ASIAV3ZUEFP6EXAMPLE",
    "SecretAccessKey": "8P+SQvWIuLnKhh8d++jpw0nNmQRBZvNEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjE0z//////////
wEXAMPLEtMSJHMEUCIDoKK3JH9uGQE1z0sINr5M4jk
+Na8KHDcCYRVjJCZEv0AiEA30vJGtw1EcVi0leS2vhs8VdCKFJQWPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fP0Ln9FX9KSYvKTr4rvx3iSILtJabIQwj2ICCR/oLxBA==",
    "Expiration": "2019-11-01T20:26:47Z"
  },
  "Audience": "https://signin.aws.amazon.com/saml",
  "SubjectType": "transient",
  "PackedPolicySize": "6",
  "NameQualifier": "SbdG0nUkh1i4+EXAMPLExL/jEvs=",
  "Subject": "SamlExample"
}
```

Weitere Informationen finden Sie im Benutzerhandbuch unter [Temporäre Sicherheitsanmeldedaten anfordern](#).AWS IAM

- APIEinzelheiten finden Sie [AssumeRoleWithSaml](#) in der AWS CLI Befehlsreferenz.

assume-role-with-web-identity

Das folgende Codebeispiel zeigt die Verwendung `assume-role-with-web-identity`.

AWS CLI

Um kurzfristige Anmeldeinformationen für eine mit Web Identity authentifizierte Rolle zu erhalten (OAuth2."0)

Der folgende `assume-role-with-web-identity` Befehl ruft eine Reihe von kurzfristigen Anmeldeinformationen für die Rolle ab. IAM app1 Die Anforderung wird mithilfe des Webidentitäts-Tokens authentifziert, das vom angegebenen Web-Identity-Anbieter bereitgestellt

wird. Zwei zusätzliche Richtlinien werden auf die Sitzung angewendet, um die Möglichkeiten des Benutzers weiter einzuschränken. Die zurückgegebenen Anmeldeinformationen laufen eine Stunde nach ihrer Generierung ab.

```
aws sts assume-role-with-web-identity \
  --duration-seconds 3600 \
  --role-session-name "app1" \
  --provider-id "www.amazon.com" \
  --policy-arns "arn:aws:iam::123456789012:policy/
q=webidentitydemopolicy1","arn:aws:iam::123456789012:policy/webidentitydemopolicy2"
  \
  --role-arn arn:aws:iam::123456789012:role/FederatedWebIdentityRole \
  --web-identity-token "Atza
%7CIQEBljAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXXJnrulxKDHwy87oGKPznh0D6bEQZTSCzyoC
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wL7WTI7jn-Pcb6M-
buCgHhF0zTQxod27L9Cqn0Lio7N3gZAGpsp6n1-
AJB0CJckcyXe2c6uD0sr0JeZLKUm2eTDVMf8IehDVI0r1Q0nTV6KzzAI30Y87Vd_cVMQ"
```

Ausgabe:

```
{
  "SubjectFromWebIdentityToken": "amzn1.account.AF6RH07KZU5XRVQJGXX6HB56KR2A"
  "Audience": "client.5498841531868486423.1548@apps.example.com",
  "AssumedRoleUser": {
    "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/
app1",
    "AssumedRoleId": "AROACLKWSQRAOEXAMPLE:app1"
  }
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYzEXAMPLEKEY",
    "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4B1CFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mR1/+0tkIKG07fAE",
    "Expiration": "2020-05-19T18:06:10+00:00"
  },
  "Provider": "www.amazon.com"
}
```

Weitere Informationen finden Sie im AWS IAMBenutzerhandbuch unter [Temporäre Sicherheitsanmeldedaten anfordern](#).

- API-Einheiten finden Sie [AssumeRoleWithWebIdentity](#) in der AWS CLI Befehlsreferenz.

assume-role

Das folgende Codebeispiel zeigt die Verwendung `assume-role`.

AWS CLI

So übernehmen Sie eine Rolle

Der folgende `assume-role` Befehl ruft eine Reihe von kurzfristigen Anmeldeinformationen für die IAM Rolle `s3-access-example` ab.

```
aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
  --role-session-name s3-access-example
```

Ausgabe:

```
{
  "AssumedRoleUser": {
    "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
    "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-example"
  },
  "Credentials": {
    "SecretAccessKey": "9drTJvcXLB89EXAMPLEL8923FB892xMFI",
    "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIhfi4citgqZTgco40Yqr41Ilo4V2b2Dyauk0eYFNebHtY1FVgAUj
+7Indz3LU0aTWk1WKIjHmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6D19zR0tXoybnlrZIwMLlMi1Kcgo50ytwU=",
    "Expiration": "2016-03-15T00:05:07Z",
    "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
  }
}
```

Die Ausgabe des Befehls enthält einen Zugriffsschlüssel, einen geheimen Schlüssel und ein Sitzungs-Token, die Sie zur Authentifizierung bei AWS verwenden können.

Zur AWS CLI Verwendung können Sie ein benanntes Profil einrichten, das einer Rolle zugeordnet ist. Wenn Sie das Profil verwenden, ruft AWS CLI es `assume-role` auf und verwaltet die Anmeldeinformationen für Sie. Weitere Informationen finden Sie unter [Verwenden einer IAM Rolle AWS CLI im AWS CLIBenutzerhandbuch](#).

- API-Einheiten finden Sie [AssumeRole](#) in der AWS CLI Befehlsreferenz.

decode-authorization-message

Das folgende Codebeispiel zeigt die Verwendung `decode-authorization-message`.

AWS CLI

Um eine codierte Autorisierungsnachricht zu dekodieren, die als Antwort auf eine Anfrage zurückgegeben wird

Das folgende `decode-authorization-message` Beispiel dekodiert zusätzliche Informationen über den Autorisierungsstatus einer Anfrage aus einer codierten Nachricht, die als Antwort auf eine Amazon Web Services Services-Anfrage zurückgegeben wurde.

```
aws sts decode-authorization-message \
  --encoded-message EXAMPLEWodyRNrtLQARDip-
eTA6i6DrLUhHhPQrLWB_1Ab15pAKx19mPDLexYcGBreyIKQC1BGBIpbKkr3dFDkwqe07e2NMk5j_hmzAiChJN-8oy3Ewi
Ojau7BMj0TWw0tHPHv_Zaz87yENDipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK44E
p0u3FZjwYStfvTb3GHs3-6rLribG09jZ0ktkfE6vqx1FzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVWPxioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHLI-
tIiNd7tgNTYzDuPQYD2FMDBnp82V9eVmYGtPp5NIeSpuf3f0HanFuBZgENxZQZ2dLH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmV0iuQdNQKuyk0p067P0FrZECLjj0tNPB0ZCcuEKEXAM
```

Ausgabe:

```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements\
\":{\n\"items\":[\n{\n\"statementId\":\n\"VisualEditor0\", \"effect\":\n\"DENY\", \"principals\
\":{\n\"items\":[\n{\n\"value\":\n\"AROA123456789EXAMPLE\"}],\n\"principalGroups\
\":{\n\"items\":[\n]\n}],\n\"actions\":{\n\"items\":[\n{\n\"value\":\n\"ec2:RunInstances\
\"}],\n\"resources\":{\n\"items\":[\n{\n\"value\":\n\"*\"}],\n\"conditions\":{\n\"items\
\":[\n]\n]}\n}],\n\"failures\":{\n\"items\":[\n]\n}],\n\"context\":{\n\"principal\":{\n\"id\":\
\n\"AROA123456789EXAMPLE:Ana\", \"arn\":\n\"arn:aws:sts::111122223333:assumed-role/\
Developer/Ana\", \"action\":\n\"RunInstances\", \"resource\":\n\"arn:aws:ec2:us-
east-1:111122223333:instance/*\", \"conditions\":{\n\"items\":[\n{\n\"key\":
```

```

\ec2:MetadataHttpPutResponseHopLimit\", \"values\": {\"items\": [{\"value\":
\"2\"}]}}}, {\"key\": \"ec2:InstanceMarketType\", \"values\": {\"items\": [{\"value
\": \"on-demand\"}]}}}, {\"key\": \"aws:Resource\", \"values\": {\"items\": [{\"value
\": \"instance/*\"}]}}}, {\"key\": \"aws:Account\", \"values\": {\"items\": [{\"value
\": \"111122223333\"}]}}}, {\"key\": \"ec2:AvailabilityZone\", \"values\": {\"items\":
[\"value\": \"us-east-1f\"}]}}}, {\"key\": \"ec2:ecsOptimized\", \"values\": {\"items
\": [{\"value\": \"false\"}]}}}, {\"key\": \"ec2:IsLaunchTemplateResource\", \"values
\": {\"items\": [{\"value\": \"false\"}]}}}, {\"key\": \"ec2:InstanceType\", \"values\":
{\"items\": [{\"value\": \"t2.micro\"}]}}}, {\"key\": \"ec2:RootDeviceType\", \"values
\": {\"items\": [{\"value\": \"efs\"}]}}}, {\"key\": \"aws:Region\", \"values\": {\"items
\": [{\"value\": \"us-east-1\"}]}}}, {\"key\": \"ec2:MetadataHttpEndpoint\", \"values
\": {\"items\": [{\"value\": \"enabled\"}]}}}, {\"key\": \"aws:Service\", \"values\":
{\"items\": [{\"value\": \"ec2\"}]}}}, {\"key\": \"ec2:InstanceID\", \"values\": {\"items
\": [{\"value\": \"*\"}]}}}, {\"key\": \"ec2:MetadataHttpTokens\", \"values\": {\"items
\": [{\"value\": \"required\"}]}}}, {\"key\": \"aws:Type\", \"values\": {\"items\":
[\"value\": \"instance\"}]}}}, {\"key\": \"ec2:Tenancy\", \"values\": {\"items\":
[\"value\": \"default\"}]}}}, {\"key\": \"ec2:Region\", \"values\": {\"items\": [{\"value
\": \"us-east-1\"}]}}}, {\"key\": \"aws:ARN\", \"values\": {\"items\": [{\"value\":
\"arn:aws:ec2:us-east-1:111122223333:instance/*\"}]}}}}}}}"
}

```

Weitere Informationen finden Sie unter [Bewertungslogik für Richtlinien](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [DecodeAuthorizationMessage](#) in der AWS CLI Befehlsreferenz.

get-caller-identity

Das folgende Codebeispiel zeigt die Verwendung `get-caller-identity`.

AWS CLI

Um Details zur aktuellen IAM Identität zu erhalten

Der folgende `get-caller-identity` Befehl zeigt Informationen über die IAM Identität an, die zur Authentifizierung der Anfrage verwendet wurde. Der Anrufer ist ein IAM Benutzer.

```
aws sts get-caller-identity
```

Ausgabe:

```
{
```

```
"UserId": "AIDASAMPLEUSERID",
"Account": "123456789012",
"Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- API-Einheiten finden Sie [GetCallerIdentity](#) in der AWS CLI Befehlsreferenz.

get-federation-token

Das folgende Codebeispiel zeigt die Verwendung `get-federation-token`.

AWS CLI

Um einen Satz temporärer Sicherheitsanmeldeinformationen mithilfe von IAM Benutzerzugriffsschlüsselanmeldedaten zurückzugeben

Im folgenden `get-federation-token` Beispiel wird ein Satz temporärer Sicherheitsanmeldeinformationen (bestehend aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheitstoken) für einen Benutzer zurückgegeben. Sie müssen den `GetFederationToken` Vorgang mit den langfristigen Sicherheitsanmeldeinformationen eines IAM Benutzers aufrufen.

```
aws sts get-federation-token \
  --name Bob \
  --policy file://myfile.json \
  --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
  --duration-seconds 900
```

Inhalt von `myfile.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe*",

```

```

    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "cloudwatch:ListMetrics",
      "cloudwatch:GetMetricStatistics",
      "cloudwatch:Describe*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "autoscaling:Describe*",
    "Resource": "*"
  }
]
}

```

Ausgabe:

```

{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "SessionToken": "EXAMPLEpZ21uX2VjEGoaCXVzLXd1c3Q0tMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42Q0unWMTfKq0DCOP//////////
wEQAxoMNDUy0TI1MTcwNTA3Igxuy3A0puuoLsk3MJwqgQPg8Q0d9HuoC1Uxq26wnc/nm
+eZLjHDyGf2KUAHK2DuaS/nrGSEXAMPLE",
    "Expiration": "2023-12-20T02:06:07+00:00"
  },
  "FederatedUser": {
    "FederatedUserId": "111122223333:Bob",
    "Arn": "arn:aws:sts::111122223333:federated-user/Bob"
  },
  "PackedPolicySize": 36
}

```

Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldedaten anfordern](#) im AWS IAMBenutzerhandbuch.

- APIEinzelheiten finden Sie [GetFederationToken](#) in der AWS CLI Befehlsreferenz.

get-session-token

Das folgende Codebeispiel zeigt die Verwendung `get-session-token`.

AWS CLI

Um eine Reihe von kurzfristigen Anmeldeinformationen für eine IAM Identität zu erhalten

Der folgende `get-session-token` Befehl ruft eine Reihe von kurzfristigen Anmeldeinformationen für die IAM Identität ab, die den Anruf getätigt hat. Die resultierenden Anmeldeinformationen können für Anfragen verwendet werden, bei denen die Richtlinie eine mehrstufige Authentifizierung (MFA) vorschreibt. Die Anmeldeinformationen verfallen 15 Minuten nach ihrer Generierung.

```
aws sts get-session-token \  
  --duration-seconds 900 \  
  --serial-number "YourMFADeviceSerialNumber" \  
  --token-code 123456
```

Ausgabe:

```
{  
  "Credentials": {  
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYzEXAMPLEKEY",  
    "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4B1CFFxWNE1OPTgk5TthT  
+FvwqnKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/  
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgkBN9bkUDNCJiBeb/  
AX1zBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mR1/+0tkIKG07fAE",  
    "Expiration": "2020-05-19T18:06:10+00:00"  
  }  
}
```

Weitere Informationen finden Sie im AWS IAM Benutzerhandbuch unter [Temporäre Sicherheitsanmeldedaten anfordern](#).

- API-Einheiten finden Sie [GetSessionToken](#) in der AWS CLI Befehlsreferenz.

AWS Support Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS Support.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-attachments-to-set

Das folgende Codebeispiel zeigt die Verwendung `add-attachments-to-set`.

AWS CLI

Um einen Anhang zu einem Set hinzuzufügen

Im folgenden `add-attachments-to-set` Beispiel wird einem Set ein Bild hinzugefügt, das Sie dann für einen Support-Fall in Ihrem AWS Konto angeben können.

```
aws support add-attachments-to-set \  
  --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQ1Br70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE" \  
  --attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

Ausgabe:

```
{  
  "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQ1Br70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE",  
  "expiryTime": "2020-05-14T17:04:40.790+0000"
```

```
}
```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [AddAttachmentsToSet](#) in der AWS CLI Befehlsreferenz.

add-communication-to-case

Das folgende Codebeispiel zeigt die Verwendung `add-communication-to-case`.

AWS CLI

Um einem Fall Kommunikation hinzuzufügen

Im folgenden `add-communication-to-case` Beispiel werden Mitteilungen zu einem Supportfall in Ihrem AWS Konto hinzugefügt.

```
aws support add-communication-to-case \  
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \  
  --communication-body "I'm attaching a set of images to this case." \  
  --cc-email-addresses "myemail@example.com" \  
  --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQ1Br70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE"
```

Ausgabe:

```
{  
  "result": true  
}
```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [AddCommunicationToCase](#) in der AWS CLI Befehlsreferenz.

create-case

Das folgende Codebeispiel zeigt die Verwendung `create-case`.

AWS CLI

Um einen Fall zu erstellen

Im folgenden `create-case` Beispiel wird ein Support-Fall für Ihr AWS Konto erstellt.

```
aws support create-case \  
  --category-code "using-aws" \  
  --cc-email-addresses "myemail@example.com" \  
  --communication-body "I want to learn more about an AWS service." \  
  --issue-type "technical" \  
  --language "en" \  
  --service-code "general-info" \  
  --severity-code "low" \  
  --subject "Question about my account"
```

Ausgabe:

```
{  
  "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"  
}
```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateCase](#) in der AWS CLI Befehlsreferenz.

describe-attachment

Das folgende Codebeispiel zeigt die Verwendung `describe-attachment`.

AWS CLI

Um einen Anhang zu beschreiben

Das folgende `describe-attachment` Beispiel gibt Informationen über den Anhang mit der angegebenen ID zurück.

```
aws support describe-attachment \  
  --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-  
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakqlc60-  
iJjL5HqyYGiT1FG8EXAMPLE"
```

Ausgabe:

```
{
```



```
"attachment": {
  "fileName": "troubleshoot-screenshot.png",
  "data": "base64-blob"
}
```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAttachment](#) in der AWS CLI Befehlsreferenz.

describe-cases

Das folgende Codebeispiel zeigt die Verwendung `describe-cases`.

AWS CLI

Um einen Fall zu beschreiben

Das folgende `describe-cases` Beispiel gibt Informationen über den angegebenen Supportfall in Ihrem AWS Konto zurück.

```
aws support describe-cases \
  --display-id "1234567890" \
  --after-time "2020-03-23T21:31:47.774Z" \
  --include-resolved-cases \
  --language "en" \
  --no-include-communications \
  --max-item 1
```

Ausgabe:

```
{
  "cases": [
    {
      "status": "resolved",
      "ccEmailAddresses": [],
      "timeCreated": "2020-03-23T21:31:47.774Z",
      "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
      "severityCode": "low",
      "language": "en",
      "categoryCode": "using-aws",
      "serviceCode": "general-info",
    }
  ]
}
```

```

        "submittedBy": "myemail@example.com",
        "displayId": "1234567890",
        "subject": "Question about my account"
    }
]
}

```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCases](#) in der AWS CLI Befehlsreferenz.

describe-communications

Das folgende Codebeispiel zeigt die Verwendung `describe-communications`.

AWS CLI

Um die neueste Mitteilung für einen Fall zu beschreiben

Das folgende `describe-communications` Beispiel gibt die neueste Kommunikation für den angegebenen Supportfall in Ihrem AWS Konto zurück.

```

aws support describe-communications \
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
  --after-time "2020-03-23T21:31:47.774Z" \
  --max-item 1

```

Ausgabe:

```

{
  "communications": [
    {
      "body": "I want to learn more about an AWS service.",
      "attachmentSet": [],
      "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
      "timeCreated": "2020-05-12T23:12:35.000Z",
      "submittedBy": "Amazon Web Services"
    }
  ],
  "NextToken": "eyJ1ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="
}

```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeCommunications](#) in der AWS CLI Befehlsreferenz.

describe-services

Das folgende Codebeispiel zeigt die Verwendung `describe-services`.

AWS CLI

Um AWS Dienste und Servicekategorien aufzulisten

Das folgende `describe-services` Beispiel listet die verfügbaren Dienstkategorien für die Anforderung allgemeiner Informationen auf.

```
aws support describe-services \  
--service-code-list "general-info"
```

Ausgabe:

```
{  
  "services": [  
    {  
      "code": "general-info",  
      "name": "General Info and Getting Started",  
      "categories": [  
        {  
          "code": "charges",  
          "name": "How Will I Be Charged?"  
        },  
        {  
          "code": "gdpr-queries",  
          "name": "Data Privacy Query"  
        },  
        {  
          "code": "reserved-instances",  
          "name": "Reserved Instances"  
        },  
        {  
          "code": "resource",  
          "name": "Where is my Resource?"  
        }  
      ]  
    }  
  ]  
}
```

```
{
  {
    "code": "using-aws",
    "name": "Using AWS & Services"
  },
  {
    "code": "free-tier",
    "name": "Free Tier"
  },
  {
    "code": "security-and-compliance",
    "name": "Security & Compliance"
  },
  {
    "code": "account-structure",
    "name": "Account Structure"
  }
]
}
```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeServices](#) in der AWS CLI Befehlsreferenz.

describe-severity-levels

Das folgende Codebeispiel zeigt die Verwendung `describe-severity-levels`.

AWS CLI

Um die verfügbaren Schweregrade aufzulisten

Im folgenden `describe-severity-levels` Beispiel werden die verfügbaren Schweregrade für einen Supportfall aufgeführt.

```
aws support describe-severity-levels
```

Ausgabe:

```
{
  "severityLevels": [
```

```
{
  "code": "low",
  "name": "Low"
},
{
  "code": "normal",
  "name": "Normal"
},
{
  "code": "high",
  "name": "High"
},
{
  "code": "urgent",
  "name": "Urgent"
},
{
  "code": "critical",
  "name": "Critical"
}
]
```

Weitere Informationen finden Sie unter [Auswählen eines Schweregrads](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSeverityLevels](#) in der AWS CLI Befehlsreferenz.

describe-trusted-advisor-check-refresh-statuses

Das folgende Codebeispiel zeigt die Verwendung `describe-trusted-advisor-check-refresh-statuses`.

AWS CLI

Um den Aktualisierungsstatus von AWS Trusted Advisor Advisor-Prüfungen aufzulisten

Im folgenden `describe-trusted-advisor-check-refresh-statuses` Beispiel werden die Aktualisierungsstatus für zwei Trusted Advisor Advisor-Prüfungen aufgeführt: Amazon S3 Bucket Permissions und IAM Use.

```
aws support describe-trusted-advisor-check-refresh-statuses \
```

```
--check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

Ausgabe:

```
{
  "statuses": [
    {
      "checkId": "Pfx0RwqBli",
      "status": "none",
      "millisUntilNextRefreshable": 0
    },
    {
      "checkId": "zXCkfM1nI3",
      "status": "none",
      "millisUntilNextRefreshable": 0
    }
  ]
}
```

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTrustedAdvisorCheckRefreshStatuses](#) in der AWS CLI Befehlsreferenz.

describe-trusted-advisor-check-result

Das folgende Codebeispiel zeigt die Verwendung `describe-trusted-advisor-check-result`.

AWS CLI

Um die Ergebnisse einer AWS Trusted Advisor-Prüfung aufzulisten

Das folgende `describe-trusted-advisor-check-result` Beispiel listet die Ergebnisse der IAM Benutzerprüfung auf.

```
aws support describe-trusted-advisor-check-result \
  --check-id "zXCkfM1nI3"
```

Ausgabe:

```
{
```

```
"result": {
  "checkId": "zXCkfM1nI3",
  "timestamp": "2020-05-13T21:38:05Z",
  "status": "ok",
  "resourcesSummary": {
    "resourcesProcessed": 1,
    "resourcesFlagged": 0,
    "resourcesIgnored": 0,
    "resourcesSuppressed": 0
  },
  "categorySpecificSummary": {
    "costOptimizing": {
      "estimatedMonthlySavings": 0.0,
      "estimatedPercentMonthlySavings": 0.0
    }
  },
  "flaggedResources": [
    {
      "status": "ok",
      "resourceId": "47DEQpj8HBSa-_TImW-5JCeuQeRkm5NMpJWZEXAMPLE",
      "isSuppressed": false
    }
  ]
}
```

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTrustedAdvisorCheckResult](#) in der AWS CLI Befehlsreferenz.

describe-trusted-advisor-check-summaries

Das folgende Codebeispiel zeigt die Verwendung `describe-trusted-advisor-check-summaries`.

AWS CLI

Um die Zusammenfassungen der AWS Trusted Advisor Advisor-Prüfungen aufzulisten

Das folgende `describe-trusted-advisor-check-summaries` Beispiel listet die Ergebnisse von zwei Trusted Advisor Advisor-Prüfungen auf: Amazon S3 Bucket Permissions und IAM Use.

```
aws support describe-trusted-advisor-check-summaries \  
--check-ids "Pfx0RwqBli" "zXCkfM1nI3"
```

Ausgabe:

```
{  
  "summaries": [  
    {  
      "checkId": "Pfx0RwqBli",  
      "timestamp": "2020-05-13T21:38:12Z",  
      "status": "ok",  
      "hasFlaggedResources": true,  
      "resourcesSummary": {  
        "resourcesProcessed": 44,  
        "resourcesFlagged": 0,  
        "resourcesIgnored": 0,  
        "resourcesSuppressed": 0  
      },  
      "categorySpecificSummary": {  
        "costOptimizing": {  
          "estimatedMonthlySavings": 0.0,  
          "estimatedPercentMonthlySavings": 0.0  
        }  
      }  
    },  
    {  
      "checkId": "zXCkfM1nI3",  
      "timestamp": "2020-05-13T21:38:05Z",  
      "status": "ok",  
      "hasFlaggedResources": true,  
      "resourcesSummary": {  
        "resourcesProcessed": 1,  
        "resourcesFlagged": 0,  
        "resourcesIgnored": 0,  
        "resourcesSuppressed": 0  
      },  
      "categorySpecificSummary": {  
        "costOptimizing": {  
          "estimatedMonthlySavings": 0.0,  
          "estimatedPercentMonthlySavings": 0.0  
        }  
      }  
    }  
  ]  
}
```



```
]
}
```

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTrustedAdvisorCheckSummaries](#) in der AWS CLI Befehlsreferenz.

describe-trusted-advisor-checks

Das folgende Codebeispiel zeigt die Verwendung `describe-trusted-advisor-checks`.

AWS CLI

Um die verfügbaren AWS Trusted Advisor Advisor-Checks aufzulisten

Das folgende `describe-trusted-advisor-checks` Beispiel listet die verfügbaren Trusted Advisor Advisor-Checks in Ihrem AWS Konto auf. Zu diesen Informationen gehören der Name, die ID, die Beschreibung, die Kategorie und die Metadaten des Checks. Beachten Sie, dass die Ausgabe aus Gründen der Lesbarkeit gekürzt ist.

```
aws support describe-trusted-advisor-checks \
  --language "en"
```

Ausgabe:

```
{
  "checks": [
    {
      "id": "zXCkfM1nI3",
      "name": "IAM Use",
      "description": "Checks for your use of AWS Identity and Access
Management (IAM). You can use IAM to create users, groups, and roles in AWS, and
you can use permissions to control access to AWS resources. \n<br>\n<br>\n<b>Alert
Criteria</b><br>\nYellow: No IAM users have been created for this account.\n<br>
\n<br>\n<b>Recommended Action</b><br>\nCreate one or more IAM users and groups in
your account. You can then create additional users whose permissions are limited
to perform specific tasks in your AWS environment. For more information, see <a
href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAMGettingStarted.html\"
target=\"_blank\">Getting Started</a>. \n<br><br>\n<b>Additional Resources</b><br>
```

```
\n<a href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAM_Introduction.html\"  
  target=\"_blank\">What Is IAM?</a>\",  
    \"category\": \"security\",  
    \"metadata\": []  
  }  
]  
}
```

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeTrustedAdvisorChecks](#) in der AWS CLI Befehlsreferenz.

refresh-trusted-advisor-check

Das folgende Codebeispiel zeigt die Verwendung `refresh-trusted-advisor-check`.

AWS CLI

So aktualisieren Sie eine AWS Trusted Advisor Advisor-Prüfung

Im folgenden `refresh-trusted-advisor-check` Beispiel wird der Trusted Advisor Advisor-Check für Amazon S3 Bucket Permissions in Ihrem AWS Konto aktualisiert.

```
aws support refresh-trusted-advisor-check \  
  --check-id "Pfx0RwqBli"
```

Ausgabe:

```
{  
  \"status\": {  
    \"checkId\": \"Pfx0RwqBli\",  
    \"status\": \"enqueued\",  
    \"millisUntilNextRefreshable\": 3599992  
  }  
}
```

Weitere Informationen finden Sie unter [AWS Trusted Advisor](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [RefreshTrustedAdvisorCheck](#) in der AWS CLI Befehlsreferenz.

resolve-case

Das folgende Codebeispiel zeigt die Verwendung `resolve-case`.

AWS CLI

Um einen Support-Fall zu lösen

Das folgende `resolve-case` Beispiel löst einen Supportfall in Ihrem AWS Konto.

```
aws support resolve-case \  
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

Ausgabe:

```
{  
  "finalCaseStatus": "resolved",  
  "initialCaseStatus": "work-in-progress"  
}
```

Weitere Informationen finden Sie unter [Fallmanagement](#) im AWS Support-Benutzerhandbuch.

- API-Einheiten finden Sie [ResolveCase](#) in der AWS CLI Befehlsreferenz.

SWF Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren SWF.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

count-closed-workflow-executions

Das folgende Codebeispiel zeigt die Verwendung `count-closed-workflow-executions`.

AWS CLI

Zählung abgeschlossener Workflow-Ausführungen

Sie können `swf count-closed-workflow-executions` damit die Anzahl der geschlossenen Workflow-Ausführungen für eine bestimmte Domäne abrufen. Sie können Filter angeben, um bestimmte Ausführungsklassen zu zählen.

Die `--start-time-filter` Argumente `--domain` und entweder `--close-time-filter` oder sind erforderlich. Alle anderen Argumente sind optional.

```
aws swf count-closed-workflow-executions \
  --domain DataFrobtzz \
  --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
1370044800 }"
```

Ausgabe:

```
{
  "count": 2,
  "truncated": false
}
```

Wenn „gekürzt“ ist `true`, steht „count“ für die maximale Anzahl, die von Amazon SWF zurückgegeben werden kann. Alle weiteren Ergebnisse werden gekürzt.

Um die Anzahl der zurückgegebenen Ergebnisse zu reduzieren, können Sie:

ändern Sie die `--start-time-filter` Werte `--close-time-filter` oder, um den durchsuchten Zeitraum einzugrenzen. Beide schließen sich gegenseitig aus: Sie können in einer Anforderung nur einen dieser Werte angeben. Verwenden Sie die `--type-filter` Argumente, `--tag-filter` oder `--close-status-filter` `--execution-filter`, um die Ergebnisse weiter zu filtern. Diese Argumente schließen sich jedoch auch gegenseitig aus.

Siehe auch [CountClosedWorkflowExecutions](#) in der Amazon Simple Workflow API Service-Referenz

- API-Einheiten finden Sie [CountClosedWorkflowExecutions](#) in der AWS CLI Befehlsreferenz.

count-open-workflow-executions

Das folgende Codebeispiel zeigt die Verwendung `count-open-workflow-executions`.

AWS CLI

Zählung offener Workflow-Ausführungen

Sie können `swf count-open-workflow-executions` damit die Anzahl der offenen Workflow-Ausführungen für eine bestimmte Domäne abrufen. Sie können Filter angeben, um bestimmte Ausführungsklassen zu zählen.

Die `--start-time-filter` Argumente `--domain` und `--start-time-filter` sind erforderlich. Alle anderen Argumente sind optional.

```
aws swf count-open-workflow-executions \
  --domain DataFrobtzz \
  --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
1370044800 }"
```

Ausgabe:

```
{
  "count": 4,
  "truncated": false
}
```

Wenn „gekürzt“ ist `true`, steht „count“ für die maximale Anzahl, die von Amazon SWF zurückgegeben werden kann. Alle weiteren Ergebnisse werden gekürzt.

Um die Anzahl der zurückgegebenen Ergebnisse zu reduzieren, können Sie:

ändern Sie die `--start-time-filter` Werte, um den durchsuchten Zeitraum einzugrenzen. Verwenden Sie die `--type-filter` Argumente, `--tag-filter` oder `--close-status-filter` `--execution-filter`, um die Ergebnisse weiter zu filtern. Beide schließen sich gegenseitig aus: Sie können in einer Anfrage nur eines davon angeben.

Weitere Informationen finden Sie `CountOpenWorkflowExecutions` in der Amazon Simple Workflow Service API Reference

- API-Einheiten finden Sie [CountOpenWorkflowExecutions](#) unter AWS CLI Befehlsreferenz.

deprecate-domain

Das folgende Codebeispiel zeigt die Verwendung `deprecate-domain`.

AWS CLI

Eine Domain als veraltet kennzeichnen

Verwenden Sie `swf deprecate-domain`, um eine Domäne als veraltet zu kennzeichnen. (Sie können die Domäne dann noch sehen, aber keine neuen Workflow-Ausführungen erstellen oder Typen für die Domäne registrieren.) Der einzige erforderliche Parameter, `--name`, akzeptiert den Namen der Domäne, die als veraltet gekennzeichnet werden soll.

```
aws swf deprecate-domain \  
  --name MyNeatNewDomain ""
```

Wie bei `register-domain` wird keine Ausgabe zurückgegeben. Wenn Sie die Option `list-domains` zur Anzeige der registrierten Domains verwenden, werden Sie jedoch feststellen, dass die Domain veraltet ist und nicht mehr in den zurückgegebenen Daten erscheint.

```
aws swf list-domains \  
  --registration-status REGISTERED \  
  {  
    "domainInfos": [  
      {  
        "status": "REGISTERED",  
        "name": "DataFrobotz"  
      },  
      {  
        "status": "REGISTERED",  
        "name": "erontest"  
      }  
    ]  
  }
```

Wenn Sie `--registration-status DEPRECATED` mit `list-domains` verwenden, wird Ihre veraltete Domain angezeigt.

```
aws swf list-domains \  
  --registration-status DEPRECATED
```

```

--registration-status DEPRECATED
  {
    "domainInfos": [
      {
        "status": "DEPRECATED",
        "name": "MyNeatNewDomain"
      }
    ]
  }

```

Sie können es weiterhin verwendend `describe-domain`, um Informationen über eine veraltete Domain abzurufen.

```

aws swf describe-domain \
  --name MyNeatNewDomain
  {
    "domainInfo": {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    },
    "configuration": {
      "workflowExecutionRetentionPeriodInDays": "0"
    }
  }

```

Siehe auch [DeprecateDomain](#) in der Amazon Simple Workflow API Service-Referenz

- API-Einheiten finden Sie [DeprecateDomain](#) in der AWS CLI Befehlsreferenz.

describe-domain

Das folgende Codebeispiel zeigt die Verwendung `describe-domain`.

AWS CLI

Informationen über eine Domain abrufen

Verwenden Sie den `swf describe-domain` Befehl, um detaillierte Informationen zu einer bestimmten Domäne zu erhalten. Es gibt einen erforderlichen Parameter: `--name`, der den Namen der Domäne akzeptiert, über die Sie Informationen abrufen möchten.

```
aws swf describe-domain \
```

```
--name DataFrobotz
{
  "domainInfo": {
    "status": "REGISTERED",
    "name": "DataFrobotz"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "1"
  }
}
```

Sie können ihn auch verwendend `describe-domain`, um Informationen über veraltete Domänen abzurufen.

```
aws swf describe-domain \
  --name MyNeatNewDomain
{
  "domainInfo": {
    "status": "DEPRECATED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}
```

Siehe auch [DescribeDomain](#) in der Amazon Simple Workflow API Service-Referenz

- API-Einheiten finden Sie [DescribeDomain](#) in der AWS CLI Befehlsreferenz.

list-activity-types

Das folgende Codebeispiel zeigt die Verwendung `list-activity-types`.

AWS CLI

Aktivitätstypen auflisten

Um eine Liste der Aktivitätstypen für eine Domain zu erhalten, verwenden Sie `swf list-activity-types`. Die `--registration-status` Argumente `--domain` und sind erforderlich.

```
aws swf list-activity-types \
```



```
--domain DataFrobtzz \  
--registration-status REGISTERED
```

Ausgabe:

```
{  
  "typeInfos": [  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.451,  
      "activityType": {  
        "version": "1",  
        "name": "confirm-user-email"  
      },  
      "description": "subscribe confirm-user-email activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.709,  
      "activityType": {  
        "version": "1",  
        "name": "confirm-user-phone"  
      },  
      "description": "subscribe confirm-user-phone activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454149.871,  
      "activityType": {  
        "version": "1",  
        "name": "get-subscription-info"  
      },  
      "description": "subscribe get-subscription-info activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.909,  
      "activityType": {  
        "version": "1",  
        "name": "send-subscription-success"  
      },  
      "description": "subscribe send-subscription-success activity"  
    }  
  ],  
}
```

```

    {
      "status": "REGISTERED",
      "creationDate": 1371454150.085,
      "activityType": {
        "version": "1",
        "name": "subscribe-user-sns"
      },
      "description": "subscribe subscribe-user-sns activity"
    }
  ]
}

```

Sie können das `--name` Argument verwenden, um nur Aktivitätstypen mit einem bestimmten Namen auszuwählen:

```

aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --name "send-subscription-success"

```

Ausgabe:

```

{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.909,
      "activityType": {
        "version": "1",
        "name": "send-subscription-success"
      },
      "description": "subscribe send-subscription-success activity"
    }
  ]
}

```

Um Ergebnisse in Seiten abzurufen, können Sie das `--maximum-page-size` Argument festlegen. Wenn mehr Ergebnisse zurückgegeben werden, als auf eine Ergebnisseite passen, wird in der Ergebnismenge ein `nextPageToken` "" zurückgegeben:

```

aws swf list-activity-types \

```

```
--domain DataFrobtzz \  
--registration-status REGISTERED \  
--maximum-page-size 2
```

Ausgabe:

```
{  
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAAA1Gp1BelJq  
+PmHvAnDxJYbup8+0R4LVtbXLD17QNY7C30pHo9Sz06D/GuFz10yC73umBQ1t0PJ/gC/  
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/  
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarki qpSY1ZVveBasBv1vyU  
WGAaqehiDz7/JzLT/wNNUM0d+Nhe",  
  "typeInfos": [  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.451,  
      "activityType": {  
        "version": "1",  
        "name": "confirm-user-email"  
      },  
      "description": "subscribe confirm-user-email activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.709,  
      "activityType": {  
        "version": "1",  
        "name": "confirm-user-phone"  
      },  
      "description": "subscribe confirm-user-phone activity"  
    }  
  ]  
}
```

Sie können den nextPageToken Wert an den nächsten Aufruf des `--next-page-token` Arguments übergeben, wodurch die nächste Ergebnisseite abgerufen wird: `list-activity-types`

```
aws swf list-activity-types \  
--domain DataFrobtzz \  
--registration-status REGISTERED \  
--maximum-page-size 2 \  

```

```
--next-page-token "AAAAKgAAAAEAAAAAAAAAAAAA1Gp1Be1Jq
+PmHvAnDxJYbup8+0R4LVtbXLDL7QNY7C30pHo9Sz06D/GuFz10yC73umBQ1t0PJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGyqIYUMLtarki qpSY1ZVveBasBvLvU
WGAaqehiDz7/JzLT/wWNNUM0d+Nhe"
```

Ausgabe:

```
{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCc1gCuq3J+h/
m3+vOfFqtkcjLwV5cc40jNAzTCuq/
XcylPumGwkjbajtqpZpbq0cVnFjFxGoi0LB20lbvv0krbUISBvlpFPmSWpDSZJsxg5UxCcweteS1Fn1PNSZ/
MoinBZo80TkjMuzcsTuK0zH9wCaR8ITcALJ3SaquHU3pyIRS5hPmFA30LIc8zaAepj1aujo6hntNSCruB4"
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.871,
      "activityType": {
        "version": "1",
        "name": "get-subscription-info"
      },
      "description": "subscribe get-subscription-info activity"
    },
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.909,
      "activityType": {
        "version": "1",
        "name": "send-subscription-success"
      },
      "description": "subscribe send-subscription-success activity"
    }
  ]
}
```

Wenn noch mehr Ergebnisse zurückgegeben werden müssen, wird "nextPageToken" zusammen mit den Ergebnissen zurückgegeben. Wenn es keine Ergebnisseiten mehr gibt, die zurückgegeben werden können, wird nextPageToken "" nicht in der Ergebnismenge zurückgegeben.

Sie können das `--reverse-order` Argument verwenden, um die Reihenfolge der zurückgegebenen Ergebnisse umzukehren. Dies wirkt sich auch auf Seitenergebnisse aus.

```
aws swf list-activity-types \  
  --domain DataFrobtzz \  
  --registration-status REGISTERED \  
  --maximum-page-size 2 \  
  --reverse-order
```

Ausgabe:

```
{  
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAwXcpu5ePSyQkrC  
+8WMbmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCsfgkEn7mk  
+yMCgVc1JxDWmB0EH46bhcmclmYNQihMDmUwocpr7To6/R7CLu0St1gkFayx0idJXErQW0zdNfQaIWAnF/  
cwioBbXlkz1fQzmDeU3M5oYGMPQIrUqkPq7pMEW0q0lK5eDN97NzFYdZZ/rlcLDWPZhUjY",  
  "typeInfos": [  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.085,  
      "activityType": {  
        "version": "1",  
        "name": "subscribe-user-sns"  
      },  
      "description": "subscribe subscribe-user-sns activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.909,  
      "activityType": {  
        "version": "1",  
        "name": "send-subscription-success"  
      },  
      "description": "subscribe send-subscription-success activity"  
    }  
  ]  
}
```

Siehe auch [ListActivityTypes](#) in der Amazon Simple Workflow API Service-Referenz

- API-Einheiten finden Sie [ListActivityTypes](#) in der AWS CLI Befehlsreferenz.

list-domains

Das folgende Codebeispiel zeigt die Verwendung `list-domains`.

AWS CLI

Beispiel 1: Um Ihre registrierten Domains aufzulisten

Das folgende `list-domains` Befehlsbeispiel listet die REGISTERED SWF Domains auf, die Sie für Ihr Konto registriert haben.

```
aws swf list-domains \  
  --registration-status REGISTERED
```

Ausgabe:

```
{  
  "domainInfos": [  
    {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    },  
    {  
      "status": "REGISTERED",  
      "name": "erontest"  
    }  
  ]  
}
```

Weitere Informationen finden Sie [ListDomains](#) in der Amazon Simple Workflow Service API Reference

Beispiel 2: Um Ihre veralteten Domains aufzulisten

Das folgende `list-domains` Befehlsbeispiel listet die DEPRECATED SWF Domains auf, die Sie für Ihr Konto registriert haben. Veraltete Domänen sind Domänen, die keine neuen Workflows oder Aktivitäten registrieren können, die aber trotzdem abgefragt werden können.

```
aws swf list-domains \  
  --registration-status DEPRECATED
```

Ausgabe:

```
{  
  "domainInfos": [  
    {
```

```

    "status": "DEPRECATED",
    "name": "MyNeatNewDomain"
  }
]
}

```

Weitere Informationen finden Sie [ListDomains](#) in der Amazon Simple Workflow Service API Reference

Beispiel 3: Um die erste Seite registrierter Domains aufzulisten

Das folgende `list-domains` Befehlsbeispiel listet die REGISTERED SWF Domains auf der ersten Seite auf, die Sie mit der `--maximum-page-size` Option für Ihr Konto registriert haben.

```

aws swf list-domains \
  --registration-status REGISTERED \
  --maximum-page-size 1

```

Ausgabe:

```

{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    }
  ],
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKE0Pm3k2r40jAGaFsG0uVbrK1jv1a7wdU7FYH301kNCP8b7PBj9SBkUyGoiAg
}

```

Weitere Informationen finden Sie [ListDomains](#) in der Amazon Simple Workflow Service API Reference

Beispiel 4: Um die angegebene Einzelseite registrierter Domains aufzulisten

Das folgende `list-domains` Befehlsbeispiel listet die REGISTERED SWF Domains auf der ersten Seite auf, die Sie mit der `--maximum-page-size` Option für Ihr Konto registriert haben.

Wenn Sie den Aufruf erneut durchführen und diesmal den Wert von `nextPageToken` im `--next-page-token` Argument angeben, erhalten Sie eine weitere Ergebnisseite.

```
aws swf list-domains \
  --registration-status REGISTERED \
  --maximum-page-size 1 \
  --next-page-token "AAAAKgAAAAEAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrsLMGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r40jAGaFsG0uVbrKljvLa7wdU7FYH30LkNCP8b7PBj9SBkUyGoiAg
```

Ausgabe:

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

Wenn keine zusätzlichen Ergebnisseiten vorhanden sind, wird nextPageToken nicht in den Ergebnissen zurückgegeben.

Weitere Informationen finden Sie [ListDomains](#) in der Amazon Simple Workflow Service API Reference

- API-Einheiten finden Sie [ListDomains](#) unter AWS CLI Befehlsreferenz.

list-workflow-types

Das folgende Codebeispiel zeigt die Verwendung `list-workflow-types`.

AWS CLI

Workflow-Typen auflisten

Um eine Liste der Workflowtypen für eine Domäne zu erhalten, verwenden Sie `swf list-workflow-types`. Die `--registration-status` Argumente `--domain` und `--registration-status` sind erforderlich. Hier ist ein einfaches Beispiel.

```
aws swf list-workflow-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED
```


Ausgabe:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.598,
      "description": "DataFrobtzz subscribe workflow",
      "workflowType": {
        "version": "v3",
        "name": "subscribe"
      }
    }
  ]
}
```

Wie bei können Sie das `--name` Argument verwenden `list-activity-types`, um nur Workflowtypen mit einem bestimmten Namen auszuwählen und das `--maximum-page-size` Argument in Abstimmung mit den Ergebnissen auf zwei Seiten `--next-page-token` zu verwenden. Um die Reihenfolge umzukehren, in der Ergebnisse zurückgegeben werden, verwenden Sie `--reverse-order`.

Siehe auch [ListWorkflowTypes](#) in der Amazon Simple Workflow API Service-Referenz

- API-Einheiten finden Sie [ListWorkflowTypes](#) in der AWS CLI Befehlsreferenz.

register-domain

Das folgende Codebeispiel zeigt die Verwendung `register-domain`.

AWS CLI**Registrierung einer Domain**

Sie können das verwenden AWS CLI, um neue Domains zu registrieren. Verwenden Sie den `swf register-domain`-Befehl. Es gibt zwei erforderliche Parameter `--name`, nämlich den Domainnamen und eine Ganzzahl `--workflow-execution-retention-period-in-days`, um die Anzahl der Tage anzugeben, für die Workflow-Ausführungsdaten in dieser Domäne aufbewahrt werden sollen, bis zu einem Höchstzeitraum von 90 Tagen (weitere Informationen finden Sie unter SWF FAQ < https://aws.amazon.com/swf/faqs/#retain_limit >). Daten zur Workflow-Ausführung werden nach Ablauf der angegebenen Anzahl von Tagen nicht aufbewahrt.

```
aws swf register-domain \
  --name MyNeatNewDomain \
  --workflow-execution-retention-period-in-days 0
""
```

Wenn Sie eine Domäne registrieren, wird nichts zurückgegeben (""). Sie können aber `swf list-domains` oder `swf describe-domain` verwenden, um die neue Domäne zu sehen.

```
aws swf list-domains \
  --registration-status REGISTERED
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    },
    {
      "status": "REGISTERED",
      "name": "MyNeatNewDomain"
    },
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

Verwenden von `swf describe-domain`:

```
aws swf describe-domain --
name MyNeatNewDomain
{
  "domainInfo": {
    "status": "REGISTERED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}
```

Siehe auch [RegisterDomain](#) in der Amazon Simple Workflow API Service-Referenz

- API-Einheiten finden Sie [RegisterDomain](#) in der AWS CLI Befehlsreferenz.

register-workflow-type

Das folgende Codebeispiel zeigt die Verwendung `register-workflow-type`.

AWS CLI

Einen Workflow-Typ registrieren

Verwenden Sie den `swf register-workflow-type` Befehl AWS CLI, um einen Workflow-Typ bei der zu registrieren.

```
aws swf register-workflow-type \  
  --domain DataFrobtzz \  
  --name "MySimpleWorkflow" \  
  --workflow-version "v1"
```

Bei Erfolg erzeugt der Befehl keine Ausgabe.

Bei einem Fehler (wenn Sie beispielsweise versuchen, denselben Workflow zweimal zu registrieren oder eine Domäne anzugeben, die nicht existiert), erhalten Sie eine Antwort in JSON

```
{  
  "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",  
  "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"  
}
```

Die `--domain`, `--name` und `swf` sind erforderlich. `--workflow-version` Sie können auch die Workflow-Beschreibung, Timeouts und die Richtlinie für untergeordnete Workflows festlegen.

Weitere Informationen finden Sie [RegisterWorkflowType](#) in der Amazon Simple Workflow Service API Reference

- API-Einheiten finden Sie [RegisterWorkflowType](#) unter AWS CLI Befehlsreferenz.

Systems Manager Manager-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Systems Manager Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

add-tags-to-resource

Das folgende Codebeispiel zeigt die Verwendung `add-tags-to-resource`.

AWS CLI

Beispiel 1: Um einem Wartungsfenster Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel wird dem angegebenen Wartungsfenster ein Tag hinzugefügt.

```
aws ssm add-tags-to-resource \  
  --resource-type "MaintenanceWindow" \  
  --resource-id "mw-03eb9db428EXAMPLE" \  
  --tags "Key=Stack,Value=Production"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einem Parameter Tags hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel werden dem angegebenen Parameter zwei Tags hinzugefügt.

```
aws ssm add-tags-to-resource \  
  --resource-type "Parameter" \  
  --resource-id "My-Parameter" \  
  --tags '[{"Key":"Region","Value":"East"}, {"Key":"Environment",  
  "Value":"Production"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 3: Um Tags zu einem SSM Dokument hinzuzufügen

Im folgenden `add-tags-to-resource` Beispiel wird dem angegebenen Dokument ein Tag hinzugefügt.

```
aws ssm add-tags-to-resource \  
  --resource-type "Document" \  
  --resource-id "My-Document" \  
  --tags "Key=Quarter, Value=Q322"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Systems Manager Manager-Ressourcen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [AddTagsToResource](#) in der AWS CLI Befehlsreferenz.

associate-ops-item-related-item

Das folgende Codebeispiel zeigt die Verwendung `associate-ops-item-related-item`.

AWS CLI

Um einen verwandten Artikel zuzuordnen

Im folgenden `associate-ops-item-related-item` Beispiel wird ein verwandtes Element dem zugeordnet OpsItem.

```
aws ssm associate-ops-item-related-item \  
  --ops-item-id "oi-649fExample" \  
  --association-type "RelatesTo" \  
  --resource-type "AWS::SSMIncidents::IncidentRecord" \  
  --resource-uri "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/c2bde883-f7d5-343a-b13a-bf5fe9ea689f"
```

Ausgabe:

```
{  
  "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"
```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Incident Manager-Vorfällen OpsCenter im AWS Systems Manager Manager-Benutzerhandbuch](#).

- API-Einheiten finden Sie [AssociateOpsItemRelatedItem](#) in der AWS CLI Befehlsreferenz.

cancel-command

Das folgende Codebeispiel zeigt die Verwendung `cancel-command`.

AWS CLI

Beispiel 1: Um einen Befehl für alle Instanzen abzubrechen

Im folgenden `cancel-command` Beispiel wird versucht, den angegebenen Befehl abzubrechen, der bereits für alle Instanzen ausgeführt wird.

```
aws ssm cancel-command \  
  --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: Um einen Befehl für bestimmte Instanzen abzubrechen

Im folgenden `cancel-command` Beispiel wird versucht, einen Befehl nur für die angegebene Instanz abzubrechen.

```
aws ssm cancel-command \  
  --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE" \  
  --instance-ids "i-02573cafEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Tagging Systems Manager Manager-Parameter](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CancelCommand](#) in der AWS CLI Befehlsreferenz.

cancel-maintenance-window-execution

Das folgende Codebeispiel zeigt die Verwendung `cancel-maintenance-window-execution`.

AWS CLI

Um die Ausführung eines Wartungsfensters abubrechen

In diesem `cancel-maintenance-window-execution` Beispiel wird die Ausführung des angegebenen Wartungsfensters gestoppt, die bereits ausgeführt wird.

```
aws ssm cancel-maintenance-window-execution \  
  --window-execution-id j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

Ausgabe:

```
{  
  "WindowExecutionId": "j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Systems Manager Maintenance Windows Tutorials \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CancelMaintenanceWindowExecution](#) in der AWS CLI Befehlsreferenz.

create-activation

Das folgende Codebeispiel zeigt die Verwendung `create-activation`.

AWS CLI

Um eine verwaltete Instanzaktivierung zu erstellen

Im folgenden `create-activation` Beispiel wird eine verwaltete Instanzaktivierung erstellt.

```
aws ssm create-activation \  
  --default-instance-name "HybridWebServers" \  
  --iam-role "HybridWebServersRole" \  
  --registration-limit 5
```

Ausgabe:

```
{  
  "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",  
  "ActivationCode": "dRmgnYaFv567vEXAMPLE"
```

```
}
```

Weitere Informationen finden Sie unter [Schritt 4: Aktivierung einer verwalteten Instanz für eine Hybridumgebung erstellen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateActivation](#) in der AWS CLI Befehlsreferenz.

create-association-batch

Das folgende Codebeispiel zeigt die Verwendung `create-association-batch`.

AWS CLI

Um mehrere Assoziationen zu erstellen

In diesem Beispiel wird ein Konfigurationsdokument mehreren Instanzen zugeordnet. Die Ausgabe gibt gegebenenfalls eine Liste mit erfolgreichen und fehlgeschlagenen Vorgängen zurück.

Befehl:

```
aws ssm create-association-batch --entries "Name=AWS-UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

Ausgabe:

```
{
  "Successful": [
    {
      "Name": "AWS-UpdateSSMAgent",
      "InstanceId": "i-1234567890abcdef0",
      "AssociationVersion": "1",
      "Date": 1550504725.007,
      "LastUpdateAssociationDate": 1550504725.007,
      "Status": {
        "Date": 1550504725.007,
        "Name": "Associated",
        "Message": "Associated with AWS-UpdateSSMAgent"
      },
      "Overview": {
        "Status": "Pending",

```



```

        "DetailedStatus": "Creating"
    },
    "DocumentVersion": "$DEFAULT",
    "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-1234567890abcdef0"
            ]
        }
    ]
},
{
    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-9876543210abcdef0",
    "AssociationVersion": "1",
    "Date": 1550504725.057,
    "LastUpdateAssociationDate": 1550504725.057,
    "Status": {
        "Date": 1550504725.057,
        "Name": "Associated",
        "Message": "Associated with AWS-UpdateSSMAgent"
    },
    "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
    },
    "DocumentVersion": "$DEFAULT",
    "AssociationId": "9c9f7f20-5154-4fed-a83e-0123456789ab",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-9876543210abcdef0"
            ]
        }
    ]
}
],
"Failed": []
}

```

- API-Einheiten finden Sie [CreateAssociationBatch](#) in der AWS CLI Befehlsreferenz.

create-association

Das folgende Codebeispiel zeigt die Verwendung `create-association`.

AWS CLI

Beispiel 1: Um ein Dokument mithilfe einer Instanz zuzuordnen IDs

In diesem Beispiel wird mithilfe von `instance` ein Konfigurationsdokument einer Instanz zugeordnet IDs.

```
aws ssm create-association \  
  --instance-id "i-0cb2b964d3e14fd9f" \  
  --name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
{  
  "AssociationDescription": {  
    "Status": {  
      "Date": 1487875500.33,  
      "Message": "Associated with AWS-UpdateSSMAgent",  
      "Name": "Associated"  
    },  
    "Name": "AWS-UpdateSSMAgent",  
    "InstanceId": "i-0cb2b964d3e14fd9f",  
    "Overview": {  
      "Status": "Pending",  
      "DetailedStatus": "Creating"  
    },  
    "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",  
    "DocumentVersion": "$DEFAULT",  
    "LastUpdateAssociationDate": 1487875500.33,  
    "Date": 1487875500.33,  
    "Targets": [  
      {  
        "Values": [  
          "i-0cb2b964d3e14fd9f"  
        ],  
        "Key": "InstanceIds"  
      }  
    ]  
  }  
}
```

```
}
```

Weitere Informationen finden Sie [CreateAssociation](#) in der AWS Systems Manager API Manager-Referenz.

Beispiel 2: So verknüpfen Sie ein Dokument mithilfe von Zielen

In diesem Beispiel wird mithilfe von Zielen ein Konfigurationsdokument mit einer Instanz verknüpft.

```
aws ssm create-association \  
  --name "AWS-UpdateSSMAgent" \  
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

Ausgabe:

```
{  
  "AssociationDescription": {  
    "Status": {  
      "Date": 1487875500.33,  
      "Message": "Associated with AWS-UpdateSSMAgent",  
      "Name": "Associated"  
    },  
    "Name": "AWS-UpdateSSMAgent",  
    "InstanceId": "i-0cb2b964d3e14fd9f",  
    "Overview": {  
      "Status": "Pending",  
      "DetailedStatus": "Creating"  
    },  
    "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",  
    "DocumentVersion": "$DEFAULT",  
    "LastUpdateAssociationDate": 1487875500.33,  
    "Date": 1487875500.33,  
    "Targets": [  
      {  
        "Values": [  
          "i-0cb2b964d3e14fd9f"  
        ],  
        "Key": "InstanceIds"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie [CreateAssociation](#) in der AWS Systems Manager API Manager-Referenz.

Beispiel 3: Um eine Assoziation zu erstellen, die nur einmal ausgeführt wird

In diesem Beispiel wird eine neue Assoziation erstellt, die nur einmal am angegebenen Datum und zu der angegebenen Uhrzeit ausgeführt wird. Verknüpfungen, die mit einem Datum in der Vergangenheit oder Gegenwart erstellt wurden (zum Zeitpunkt der Verarbeitung liegt das Datum in der Vergangenheit), werden sofort ausgeführt.

```
aws ssm create-association \  
  --name "AWS-UpdateSSMAgent" \  
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \  
  --schedule-expression "at(2020-05-14T15:55:00)" \  
  --apply-only-at-cron-interval
```

Ausgabe:

```
{  
  "AssociationDescription": {  
    "Status": {  
      "Date": 1487875500.33,  
      "Message": "Associated with AWS-UpdateSSMAgent",  
      "Name": "Associated"  
    },  
    "Name": "AWS-UpdateSSMAgent",  
    "InstanceId": "i-0cb2b964d3e14fd9f",  
    "Overview": {  
      "Status": "Pending",  
      "DetailedStatus": "Creating"  
    },  
    "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",  
    "DocumentVersion": "$DEFAULT",  
    "LastUpdateAssociationDate": 1487875500.33,  
    "Date": 1487875500.33,  
    "Targets": [  
      {  
        "Values": [  
          "i-0cb2b964d3e14fd9f"  
        ],  
        "Key": "InstanceIds"  
      }  
    ]  
  }  
}
```

```

    ]
  }
}

```

Weitere Informationen finden Sie [CreateAssociation](#) in der AWS Systems API Manager-Referenz oder Referenz: [Cron- und Rate-Ausdrücke für Systems Manager](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateAssociation AWS CLI](#) Befehlsreferenz.

create-document

Das folgende Codebeispiel zeigt die Verwendung `create-document`.

AWS CLI

Um ein Dokument zu erstellen

Im folgenden `create-document` Beispiel wird ein Systems Manager Manager-Dokument erstellt.

```

aws ssm create-document \
  --content file://exampleDocument.yml \
  --name "Example" \
  --document-type "Automation" \
  --document-format YAML

```

Ausgabe:

```

{
  "DocumentDescription": {
    "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
    "HashType": "Sha256",
    "Name": "Example",
    "Owner": "29884EXAMPLE",
    "CreateDate": 1583256349.452,
    "Status": "Creating",
    "DocumentVersion": "1",
    "Description": "Document Example",
    "Parameters": [
      {
        "Name": "AutomationAssumeRole",
        "Type": "String",

```

```
        "Description": "(Required) The ARN of the role that allows
Automation to perform the actions on your behalf. If no role is specified, Systems
Manager Automation uses your IAM permissions to execute this document.",
        "DefaultValue": ""
    },
    {
        "Name": "InstanceId",
        "Type": "String",
        "Description": "(Required) The ID of the Amazon EC2 instance.",
        "DefaultValue": ""
    }
],
"PlatformTypes": [
    "Windows",
    "Linux"
],
"DocumentType": "Automation",
"SchemaVersion": "0.3",
"LatestVersion": "1",
"DefaultVersion": "1",
"DocumentFormat": "YAML",
"Tags": []
}
}
```

Weitere Informationen finden Sie unter [Erstellen von Systems Manager Manager-Dokumenten](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateDocument](#) in der AWS CLI Befehlsreferenz.

create-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `create-maintenance-window`.

AWS CLI

Beispiel 1: Um ein Wartungsfenster zu erstellen

Im folgenden `create-maintenance-window` Beispiel wird ein neues Wartungsfenster erstellt, das alle fünf Minuten für bis zu zwei Stunden (je nach Bedarf) den Start neuer Aufgaben innerhalb einer Stunde nach Ende der Ausführung des Wartungsfensters verhindert, nicht zugeordnete Ziele (Instanzen, die Sie nicht für das Wartungsfenster registriert haben) zulässt und durch die

Verwendung benutzerdefinierter Tags darauf hinweist, dass der Ersteller beabsichtigt, es in einem Tutorial zu verwenden.

```
aws ssm create-maintenance-window \
  --name "My-Tutorial-Maintenance-Window" \
  --schedule "rate(5 minutes)" \
  --duration 2 --cutoff 1 \
  --allow-unassociated-targets \
  --tags "Key=Purpose, Value=Tutorial"
```

Ausgabe:

```
{
  "WindowId": "mw-0c50858d01EXAMPLE"
}
```

Beispiel 2: Um ein Wartungsfenster zu erstellen, das nur einmal ausgeführt wird

Im folgenden `create-maintenance-window` Beispiel wird ein neues Wartungsfenster erstellt, das nur einmal am angegebenen Datum und zur angegebenen Uhrzeit ausgeführt wird.

```
aws ssm create-maintenance-window \
  --name My-One-Time-Maintenance-Window \
  --schedule "at(2020-05-14T15:55:00)" \
  --duration 5 \
  --cutoff 2 \
  --allow-unassociated-targets \
  --tags "Key=Environment, Value=Production"
```

Ausgabe:

```
{
  "WindowId": "mw-01234567890abcdef"
}
```

Weitere Informationen finden Sie unter [Maintenance Windows](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

create-ops-item

Das folgende Codebeispiel zeigt die Verwendung `create-ops-item`.

AWS CLI

Um ein zu erstellen OpsItems

Im folgenden `create-ops-item` Beispiel wird the `/aws/resources` key in verwendet `OperationalData` , um eine OpsItem mit Amazon DynamoDB verknüpfte Ressource zu erstellen.

```
aws ssm create-ops-item \
  --title "EC2 instance disk full" \
  --description "Log clean up may have failed which caused the disk to be full" \
  --priority 2 \
  --source ec2 \
  --operational-data '{"aws/resources":{"Value":[{"arn": "arn:aws:dynamodb:us-west-2:12345678:table/OpsItems"}, {"arn": "arn:aws:dynamodb:us-west-2:12345678:table/OpsItems"}]}, "Type": "SearchableString"}' \
  --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Ausgabe:

```
{
  "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

Weitere Informationen finden Sie unter [Erstellen OpsItems](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateOpsItem](#) in der AWS CLI Befehlsreferenz.

create-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `create-patch-baseline`.

AWS CLI

Beispiel 1: So erstellen Sie eine Patch-Baseline mit automatischer Genehmigung

Im folgenden `create-patch-baseline` Beispiel wird eine Patch-Baseline für Windows Server erstellt, die Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung durch Microsoft genehmigt.


```
aws ssm create-patch-baseline \
  --name "Windows-Production-Baseline-AutoApproval" \
  --operating-system "WINDOWS" \
  --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
\
  --description "Baseline containing all updates approved for Windows Server
production systems"
```

Ausgabe:

```
{
  "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Beispiel 2: So erstellen Sie eine Patch-Baseline mit einem Stichtag für die Genehmigung

Im folgenden create-patch-baseline Beispiel wird eine Patch-Baseline für Windows Server erstellt, die alle Patches für eine Produktionsumgebung genehmigt, die am oder vor dem 7. Juli 2020 veröffentlicht wurden.

```
aws ssm create-patch-baseline \
  --name "Windows-Production-Baseline-AutoApproval" \
  --operating-system "WINDOWS" \
  --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
\
  --description "Baseline containing all updates approved for Windows Server
production systems"
```

Ausgabe:

```
{
  "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Beispiel 3: So erstellen Sie eine Patch-Baseline mit in einer JSON Datei gespeicherten Genehmigungsregeln

Im folgenden `create-patch-baseline` Beispiel wird eine Patch-Baseline für Amazon Linux 2017.09 erstellt, die Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung genehmigt, Genehmigungsregeln für die Patch-Baseline festlegt und ein benutzerdefiniertes Repository für Patches festlegt.

```
aws ssm create-patch-baseline \  
  --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Inhalt von `my-amazon-linux-approval-rules-and-repo.json`:

```
{  
  "Name": "Amazon-Linux-2017.09-Production-Baseline",  
  "Description": "My approval rules patch baseline for Amazon Linux 2017.09  
instances",  
  "OperatingSystem": "AMAZON_LINUX",  
  "Tags": [  
    {  
      "Key": "Environment",  
      "Value": "Production"  
    }  
  ],  
  "ApprovalRules": {  
    "PatchRules": [  
      {  
        "ApproveAfterDays": 7,  
        "EnableNonSecurity": true,  
        "PatchFilterGroup": {  
          "PatchFilters": [  
            {  
              "Key": "SEVERITY",  
              "Values": [  
                "Important",  
                "Critical"  
              ]  
            },  
            {  
              "Key": "CLASSIFICATION",  
              "Values": [  
                "Security",  
                "Bugfix"  
              ]  
            }  
          ]  
        }  
      }  
    ],  
  }  
}
```

```

        {
            "Key": "PRODUCT",
            "Values": [
                "AmazonLinux2017.09"
            ]
        }
    ]
}
},
"Sources": [
    {
        "Name": "My-AL2017.09",
        "Products": [
            "AmazonLinux2017.09"
        ],
        "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/mirror.list //
\nmirrorlist_expire=300//\nmetadata_expire=300 \npriority=10 \nfailovermethod=priority
\nfastestmirror_enabled=0 \ngpgcheck=1 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-
KEY-amazon-ga \nenabled=1 \nretries=3 \ntimeout=5\nreport_instanceid=yes"
    }
]
}

```

Beispiel 4: Um eine Patch-Baseline zu erstellen, die genehmigte und abgelehnte Patches angibt

Im folgenden `create-patch-baseline` Beispiel werden Patches, die genehmigt und abgelehnt werden sollen, ausdrücklich als Ausnahme von den Standard-Genehmigungsregeln angegeben.

```

aws ssm create-patch-baseline \
  --name "Amazon-Linux-2017.09-Alpha-Baseline" \
  --description "My custom approve/reject patch baseline for Amazon Linux 2017.09 instances" \
  --operating-system "AMAZON_LINUX" \
  --approved-patches "CVE-2018-1234567,example-pkg-EE-2018*.amzn1.noarch" \
  --approved-patches-compliance-level "HIGH" \
  --approved-patches-enable-non-security \
  --tags "Key=Environment,Value=Alpha"

```

Weitere Informationen finden Sie unter [Erstellen einer benutzerdefinierten Patch-Baseline](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [CreatePatchBaseline](#) unter AWS CLI Befehlsreferenz.

create-resource-data-sync

Das folgende Codebeispiel zeigt die Verwendung `create-resource-data-sync`.

AWS CLI

Um eine Ressourcendatensynchronisierung zu erstellen

In diesem Beispiel wird eine Ressourcendatensynchronisierung erstellt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl:

```
aws ssm create-resource-data-sync --sync-name "ssm-resource-data-sync" --s3-destination "BucketName=ssm-bucket,Prefix=inventory,SyncFormat=JsonSerDe,Region=us-east-1"
```

- API-Einheiten finden Sie [CreateResourceDataSync](#) in der AWS CLI Befehlsreferenz.

delete-activation

Das folgende Codebeispiel zeigt die Verwendung `delete-activation`.

AWS CLI

Um eine verwaltete Instanzaktivierung zu löschen

Im folgenden `delete-activation` Beispiel wird eine verwaltete Instanzaktivierung gelöscht.

```
aws ssm delete-activation \
  --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Setting Up AWS Systems Manager for Hybrid Environments](#) im AWS Systems Manager Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteActivation](#) in der AWS CLI Befehlsreferenz.

delete-association

Das folgende Codebeispiel zeigt die Verwendung `delete-association`.

AWS CLI

Beispiel 1: Um eine Assoziation mithilfe der Zuordnungs-ID zu löschen

Im folgenden `delete-association` Beispiel wird die Assoziation für die angegebene Zuordnungs-ID gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm delete-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Bearbeiten und Erstellen einer neuen Version einer Zuordnung](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So löschen Sie eine Zuordnung

Im folgenden `delete-association` Beispiel wird die Verknüpfung zwischen einer Instanz und einem Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm delete-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Zuordnungen in Systems Manager](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAssociation](#) in der AWS CLI Befehlsreferenz.

delete-document

Das folgende Codebeispiel zeigt die Verwendung `delete-document`.

AWS CLI

Um ein Dokument zu löschen

Im folgenden `delete-document` Beispiel wird ein Systems Manager Manager-Dokument gelöscht.

```
aws ssm delete-document \  
  --name "Example"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erstellen von Systems Manager Manager-Dokumenten](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteDocument](#) in der AWS CLI Befehlsreferenz.

delete-inventory

Das folgende Codebeispiel zeigt die Verwendung `delete-inventory`.

AWS CLI

Um einen benutzerdefinierten Inventartyp zu löschen

In diesem Beispiel wird ein benutzerdefiniertes Inventarschema gelöscht.

Befehl:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DeleteSchema"
```

Ausgabe:

```
{  
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",  
  "TypeName": "Custom:RackInfo",  
  "DeletionSummary": {  
    "TotalCount": 1,  
    "RemainingCount": 1,  
    "SummaryItems": [  
      {  
        "Version": "1.0",  
        "Count": 1,  
        "RemainingCount": 1
```

```
    }  
  ]  
}  
}
```

Um einen benutzerdefinierten Inventartyp zu deaktivieren

In diesem Beispiel wird ein benutzerdefiniertes Inventarschema deaktiviert.

Befehl:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DisableSchema"
```

Ausgabe:

```
{  
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",  
  "TypeName": "Custom:RackInformation",  
  "DeletionSummary": {  
    "TotalCount": 0,  
    "RemainingCount": 0,  
    "SummaryItems": []  
  }  
}
```

- API-Einheiten finden Sie [DeleteInventory](#) in der AWS CLI Befehlsreferenz.

delete-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `delete-maintenance-window`.

AWS CLI

Um ein Wartungsfenster zu löschen

In diesem `delete-maintenance-window` Beispiel wird das angegebene Wartungsfenster entfernt.

```
aws ssm delete-maintenance-window \
```

```
--window-id "mw-1a2b3c4d5e6f7g8h9"
```

Ausgabe:

```
{  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9"  
}
```

Weitere Informationen finden Sie unter [Löschen eines Wartungsfensters \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

delete-parameter

Das folgende Codebeispiel zeigt die Verwendung `delete-parameter`.

AWS CLI

Um einen Parameter zu löschen

Im folgenden `delete-parameter` Beispiel wird der angegebene Einzelparameter gelöscht.

```
aws ssm delete-parameter \  
  --name "MyParameter"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit dem Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteParameter](#) in der AWS CLI Befehlsreferenz.

delete-parameters

Das folgende Codebeispiel zeigt die Verwendung `delete-parameters`.

AWS CLI

Um eine Liste von Parametern zu löschen

Im folgenden `delete-parameters` Beispiel werden die angegebenen Parameter gelöscht.

```
aws ssm delete-parameters \  
  --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

Ausgabe:

```
{  
  "DeletedParameters": [  
    "MyFirstParameter",  
    "MySecondParameter"  
  ],  
  "InvalidParameters": [  
    "MyInvalidParameterName"  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteParameters](#) in der AWS CLI Befehlsreferenz.

delete-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `delete-patch-baseline`.

AWS CLI

Um eine Patch-Baseline zu löschen

Im folgenden `delete-patch-baseline` Beispiel wird die angegebene Patch-Baseline gelöscht.

```
aws ssm delete-patch-baseline \  
  --baseline-id "pb-045f10b4f382baeda"
```

Ausgabe:

```
{  
  "BaselineId": "pb-045f10b4f382baeda"  
}
```

Weitere Informationen finden Sie unter [Aktualisieren oder Löschen einer Patch-Baseline \(Konsole\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeletePatchBaseline](#) in der AWS CLI Befehlsreferenz.

delete-resource-data-sync

Das folgende Codebeispiel zeigt die Verwendung `delete-resource-data-sync`.

AWS CLI

Um eine Ressourcendatensynchronisierung zu löschen

In diesem Beispiel wird eine Ressourcendatensynchronisierung gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl:

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- API-Einheiten finden Sie [DeleteResourceDataSync](#) in der AWS CLI Befehlsreferenz.

deregister-managed-instance

Das folgende Codebeispiel zeigt die Verwendung `deregister-managed-instance`.

AWS CLI

Um die Registrierung einer verwalteten Instanz aufzuheben

Im folgenden `deregister-managed-instance` Beispiel wird die Registrierung der angegebenen verwalteten Instanz aufgehoben.

```
aws ssm deregister-managed-instance  
--instance-id "mi-08ab247cdfEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Deregistering Managed Instances in a Hybrid Environment](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterManagedInstance](#) in der AWS CLI Befehlsreferenz.

deregister-patch-baseline-for-patch-group

Das folgende Codebeispiel zeigt die Verwendung `deregister-patch-baseline-for-patch-group`.

AWS CLI

Um eine Patchgruppe von einer Patch-Baseline abzumelden

Im folgenden `deregister-patch-baseline-for-patch-group` Beispiel wird die Registrierung der angegebenen Patchgruppe von der angegebenen Patch-Baseline aufgehoben.

```
aws ssm deregister-patch-baseline-for-patch-group \  
  --patch-group "Production" \  
  --baseline-id "pb-0ca44a362fEXAMPLE"
```

Ausgabe:

```
{  
  "PatchGroup": "Production",  
  "BaselineId": "pb-0ca44a362fEXAMPLE"  
}
```

Weitere Informationen finden [Sie unter Hinzufügen einer Patchgruppe zu einer Patch-Baseline](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterPatchBaselineForPatchGroup](#) unter AWS CLI Befehlsreferenz.

deregister-target-from-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `deregister-target-from-maintenance-window`.

AWS CLI

Um ein Ziel aus einem Wartungsfenster zu entfernen

Im folgenden `deregister-target-from-maintenance-window` Beispiel wird das angegebene Ziel aus dem angegebenen Wartungsfenster entfernt.

```
aws ssm deregister-target-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Ausgabe:

```
{  
  "WindowId": "mw-ab12cd34ef56gh78",  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Weitere Informationen finden Sie unter [Aktualisieren eines Wartungsfensters \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterTargetFromMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

deregister-task-from-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `deregister-task-from-maintenance-window`.

AWS CLI

Um eine Aufgabe aus einem Wartungsfenster zu entfernen

Im folgenden `deregister-task-from-maintenance-window` Beispiel wird die angegebene Aufgabe aus dem angegebenen Wartungsfenster entfernt.

```
aws ssm deregister-task-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Ausgabe:

```
{  
  "WindowTaskId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",  
  "WindowId": "mw-ab12cd34ef56gh78"  
}
```

Weitere Informationen finden Sie unter [Systems Manager Maintenance Windows Tutorials \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterTaskFromMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

describe-activations

Das folgende Codebeispiel zeigt die Verwendung `describe-activations`.

AWS CLI

Um Aktivierungen zu beschreiben

Das folgende `describe-activations` Beispiel listet Details zu den Aktivierungen in Ihrem AWS Konto auf.

```
aws ssm describe-activations
```

Ausgabe:

```
{
  "ActivationList": [
    {
      "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
      "Description": "Example1",
      "IamRole": "HybridWebServersRole",
      "RegistrationLimit": 5,
      "RegistrationsCount": 5,
      "ExpirationDate": 1584316800.0,
      "Expired": false,
      "CreateDate": 1581954699.792
    },
    {
      "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
      "Description": "Example2",
      "IamRole": "HybridDatabaseServersRole",
      "RegistrationLimit": 5,
      "RegistrationsCount": 5,
      "ExpirationDate": 1580515200.0,
      "Expired": true,
      "CreateDate": 1578064132.002
    },
  ],
}
```

```
]
}
```

Weitere Informationen finden Sie unter [Schritt 4: Aktivierung einer verwalteten Instanz für eine Hybridumgebung erstellen](#) im AWS Systems Manager Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeActivations](#) in der AWS CLI Befehlsreferenz.

describe-association-execution-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-association-execution-targets`.

AWS CLI

Um Details zur Ausführung einer Assoziation abzurufen

Das folgende `describe-association-execution-targets` Beispiel beschreibt die angegebene Assoziationsausführung.

```
aws ssm describe-association-execution-targets \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

Ausgabe:

```
{
  "AssociationExecutionTargets": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
      "ResourceId": "i-1234567890abcdef0",
      "ResourceType": "ManagedInstance",
      "Status": "Success",
      "DetailedStatus": "Success",
      "LastExecutionDate": 1550505538.497,
      "OutputSource": {
        "OutputSourceId": "97fff367-fc5a-4299-aed8-0123456789ab",
        "OutputSourceType": "RunCommand"
      }
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Anzeigen von Zuordnungsverläufen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAssociationExecutionTargets](#) in der AWS CLI Befehlsreferenz.

describe-association-executions

Das folgende Codebeispiel zeigt die Verwendung `describe-association-executions`.

AWS CLI

Beispiel 1: Um Details zu allen Ausführungen für eine Assoziation abzurufen

Das folgende `describe-association-executions` Beispiel beschreibt alle Ausführungen der angegebenen Assoziation.

```
aws ssm describe-association-executions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
  "AssociationExecutions": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
      "Status": "Success",
      "DetailedStatus": "Success",
      "CreatedTime": 1550505827.119,
      "ResourceCountByStatus": "{Success=1}"
    },
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
      "Status": "Success",
      "DetailedStatus": "Success",
      "CreatedTime": 1550505536.843,
```

```

        "ResourceCountByStatus": "{Success=1}"
    },
    ...
]
}

```

Weitere Informationen finden Sie unter [Anzeigen von Zuordnungsverläufen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: Um Details zu allen Ausführungen für eine Zuordnung nach einem bestimmten Datum und einer bestimmten Uhrzeit abzurufen

Das folgende `describe-association-executions` Beispiel beschreibt alle Ausführungen einer Assoziation nach dem angegebenen Datum und der angegebenen Uhrzeit.

```

aws ssm describe-association-executions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"

```

Ausgabe:

```

{
  "AssociationExecutions": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
      "Status": "Success",
      "DetailedStatus": "Success",
      "CreatedTime": 1550505827.119,
      "ResourceCountByStatus": "{Success=1}"
    },
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
      "Status": "Success",
      "DetailedStatus": "Success",
      "CreatedTime": 1550505536.843,
      "ResourceCountByStatus": "{Success=1}"
    },
    ...
  ]
}

```



```
}
```

Weitere Informationen finden Sie unter [Anzeigen von Zuordnungsverläufen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAssociationExecutions](#) in der AWS CLI Befehlsreferenz.

describe-association

Das folgende Codebeispiel zeigt die Verwendung `describe-association`.

AWS CLI

Beispiel 1: Um Details zu einer Assoziation abzurufen

Das folgende `describe-association` Beispiel beschreibt die Assoziation für die angegebene Zuordnungs-ID.

```
aws ssm describe-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
  "AssociationDescription": {
    "Name": "AWS-GatherSoftwareInventory",
    "AssociationVersion": "1",
    "Date": 1534864780.995,
    "LastUpdateAssociationDate": 1543235759.81,
    "Overview": {
      "Status": "Success",
      "AssociationStatusAggregatedCount": {
        "Success": 2
      }
    }
  },
  "DocumentVersion": "$DEFAULT",
  "Parameters": {
    "applications": [
      "Enabled"
    ],
    "awsComponents": [
      "Enabled"
    ]
  },
}
```

```
    "customInventory": [
      "Enabled"
    ],
    "files": [
      ""
    ],
    "instanceDetailedInformation": [
      "Enabled"
    ],
    "networkConfig": [
      "Enabled"
    ],
    "services": [
      "Enabled"
    ],
    "windowsRegistry": [
      ""
    ],
    "windowsRoles": [
      "Enabled"
    ],
    "windowsUpdates": [
      "Enabled"
    ]
  },
  "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
  "Targets": [
    {
      "Key": "InstanceIds",
      "Values": [
        "*"
      ]
    }
  ],
  "ScheduleExpression": "rate(24 hours)",
  "LastExecutionDate": 1550501886.0,
  "LastSuccessfulExecutionDate": 1550501886.0,
  "AssociationName": "Inventory-Association"
}
```

Weitere Informationen finden Sie unter [Bearbeiten und Erstellen einer neuen Version einer Zuordnung](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So rufen Sie Details zu einer Zuordnung für eine bestimmte Instanz und ein bestimmtes Dokument ab

Das folgende describe-association Beispiel beschreibt die Zuordnung zwischen einer Instanz und einem Dokument.

```
aws ssm describe-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
{  
  "AssociationDescription": {  
    "Status": {  
      "Date": 1487876122.564,  
      "Message": "Associated with AWS-UpdateSSMAgent",  
      "Name": "Associated"  
    },  
    "Name": "AWS-UpdateSSMAgent",  
    "InstanceId": "i-1234567890abcdef0",  
    "Overview": {  
      "Status": "Pending",  
      "DetailedStatus": "Associated",  
      "AssociationStatusAggregatedCount": {  
        "Pending": 1  
      }  
    },  
    "AssociationId": "d8617c07-2079-4c18-9847-1234567890ab",  
    "DocumentVersion": "$DEFAULT",  
    "LastUpdateAssociationDate": 1487876122.564,  
    "Date": 1487876122.564,  
    "Targets": [  
      {  
        "Values": [  
          "i-1234567890abcdef0"  
        ],  
        "Key": "InstanceIds"  
      }  
    ]  
  }  
}
```

Weitere Informationen finden Sie unter [Bearbeiten und Erstellen einer neuen Version einer Zuordnung](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAssociation](#) unter AWS CLI Befehlsreferenz.

describe-automation-executions

Das folgende Codebeispiel zeigt die Verwendung `describe-automation-executions`.

AWS CLI

Um eine Automatisierungsausführung zu beschreiben

Im folgenden `describe-automation-executions` Beispiel werden Details zu einer Automatisierungsausführung angezeigt.

```
aws ssm describe-automation-executions \  
  --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{  
  "AutomationExecutionMetadataList": [  
    {  
      "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",  
      "DocumentName": "AWS-StartEC2Instance",  
      "DocumentVersion": "1",  
      "AutomationExecutionStatus": "Success",  
      "ExecutionStartTime": 1583737233.748,  
      "ExecutionEndTime": 1583737234.719,  
      "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/  
OrchestrationService",  
      "LogFile": "",  
      "Outputs": {},  
      "Mode": "Auto",  
      "Targets": [],  
      "ResolvedTargets": {  
        "ParameterValues": [],  
        "Truncated": false  
      },  
      "AutomationType": "Local"  
    }  
  ]  
}
```

```
}
```

Weitere Informationen finden Sie unter [Running a Simple Automation Workflow](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAutomationExecutions](#) unter AWS CLI Befehlsreferenz.

describe-automation-step-executions

Das folgende Codebeispiel zeigt die Verwendung `describe-automation-step-executions`.

AWS CLI

Beispiel 1: Um alle Schritte für eine Automatisierungsausführung zu beschreiben

Das folgende `describe-automation-step-executions` Beispiel zeigt Details zu den Schritten einer Automatisierungsausführung.

```
aws ssm describe-automation-step-executions \  
--automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{  
  "StepExecutions": [  
    {  
      "StepName": "startInstances",  
      "Action": "aws:changeInstanceState",  
      "ExecutionStartTime": 1583737234.134,  
      "ExecutionEndTime": 1583737234.672,  
      "StepStatus": "Success",  
      "Inputs": {  
        "DesiredState": "\"running\"",  
        "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"  
      },  
      "Outputs": {  
        "InstanceStates": [  
          "running"  
        ]  
      },  
      "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",  
      "OverriddenParameters": {}  
    }  
  ]  
}
```

```

    }
  ]
}

```

Beispiel 2: Um einen bestimmten Schritt für eine Automatisierungsausführung zu beschreiben

Das folgende `describe-automation-step-executions` Beispiel zeigt Details zu einem bestimmten Schritt einer Automatisierungsausführung.

```

aws ssm describe-automation-step-executions \
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
  --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE

```

Weitere Informationen finden Sie unter [Schrittweises Ausführen eines Automatisierungswflows \(Befehlszeile\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- APIEinzelheiten finden Sie [DescribeAutomationStepExecutions](#) unter AWS CLI Befehlsreferenz.

`describe-available-patches`

Das folgende Codebeispiel zeigt die Verwendung `describe-available-patches`.

AWS CLI

Um verfügbare Patches zu erhalten

Im folgenden `describe-available-patches` Beispiel werden Details zu allen verfügbaren Patches für Windows Server 2019 abgerufen, die den MSRC Schweregrad Kritisch haben.

```

aws ssm describe-available-patches \
  --
  filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"

```

Ausgabe:

```

{
  "Patches": [
    {
      "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
      "ReleaseDate": 1544047205.0,
      "Title": "2018-11 Update for Windows Server 2019 for x64-based Systems (KB4470788)",

```

```

        "Description": "Install this update to resolve issues in Windows. For a
complete listing of the issues that are included in this update, see the associated
Microsoft Knowledge Base article for more information. After you install this item,
you may have to restart your computer.",
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
        "Vendor": "Microsoft",
        "ProductFamily": "Windows",
        "Product": "WindowsServer2019",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Critical",
        "KbNumber": "KB4470788",
        "MsrcNumber": "",
        "Language": "All"
    },
    {
        "Id": "c96115e1-5587-4115-b851-22baa46a3f11",
        "ReleaseDate": 1549994410.0,
        "Title": "2019-02 Security Update for Adobe Flash Player for Windows
Server 2019 for x64-based Systems (KB4487038)",
        "Description": "A security issue has been identified in a Microsoft
software product that could affect your system. You can help protect your system
by installing this update from Microsoft. For a complete listing of the issues that
are included in this update, see the associated Microsoft Knowledge Base article.
After you install this update, you may have to restart your system.",
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4487038",
        "Vendor": "Microsoft",
        "ProductFamily": "Windows",
        "Product": "WindowsServer2019",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Critical",
        "KbNumber": "KB4487038",
        "MsrcNumber": "",
        "Language": "All"
    },
    ...
]
}

```

Um Details zu einem bestimmten Patch abzurufen

Im folgenden `describe-available-patches` Beispiel werden Details zum angegebenen Patch abgerufen.

```
aws ssm describe-available-patches \  
  --filters "Key=PATCH_ID,Values=KB4480979"
```

Ausgabe:

```
{  
  "Patches": [  
    {  
      "Id": "680861e3-fb75-432e-818e-d72e5f2be719",  
      "ReleaseDate": 1546970408.0,  
      "Title": "2019-01 Security Update for Adobe Flash Player for Windows  
Server 2016 for x64-based Systems (KB4480979)",  
      "Description": "A security issue has been identified in a Microsoft  
software product that could affect your system. You can help protect your system  
by installing this update from Microsoft. For a complete listing of the issues that  
are included in this update, see the associated Microsoft Knowledge Base article.  
After you install this update, you may have to restart your system.",  
      "ContentUrl": "https://support.microsoft.com/en-us/kb/4480979",  
      "Vendor": "Microsoft",  
      "ProductFamily": "Windows",  
      "Product": "WindowsServer2016",  
      "Classification": "SecurityUpdates",  
      "MsrcSeverity": "Critical",  
      "KbNumber": "KB4480979",  
      "MsrcNumber": "",  
      "Language": "All"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [So funktionieren Patch Manager-Operationen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeAvailablePatches](#) unter AWS CLI Befehlsreferenz.

describe-document-permission

Das folgende Codebeispiel zeigt die Verwendung `describe-document-permission`.

AWS CLI

Um die Berechtigungen für Dokumente zu beschreiben

Im folgenden `describe-document-permission` Beispiel werden Berechtigungsdetails zu einem Systems Manager Manager-Dokument angezeigt, das öffentlich geteilt wird.

```
aws ssm describe-document-permission \  
  --name "Example" \  
  --permission-type "Share"
```

Ausgabe:

```
{  
  "AccountIds": [  
    "all"  
  ],  
  "AccountSharingInfoList": [  
    {  
      "AccountId": "all",  
      "SharedDocumentVersion": "$DEFAULT"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Freigeben eines Systems Manager Manager-Dokuments](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDocumentPermission](#) in der AWS CLI Befehlsreferenz.

describe-document

Das folgende Codebeispiel zeigt die Verwendung `describe-document`.

AWS CLI

Um Details eines Dokuments anzuzeigen

Im folgenden `describe-document` Beispiel werden Details zu einem Systems Manager Manager-Dokument in Ihrem AWS Konto angezeigt.

```
aws ssm describe-document \  
  --name "Example"
```

Ausgabe:

```

{
  "Document": {
    "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
    "HashType": "Sha256",
    "Name": "Example",
    "Owner": "29884EXAMPLE",
    "CreateDate": 1583257938.266,
    "Status": "Active",
    "DocumentVersion": "1",
    "Description": "Document Example",
    "Parameters": [
      {
        "Name": "AutomationAssumeRole",
        "Type": "String",
        "Description": "(Required) The ARN of the role that allows
Automation to perform the actions on your behalf. If no role is specified, Systems
Manager Automation uses your IAM permissions to execute this document.",
        "DefaultValue": ""
      },
      {
        "Name": "InstanceId",
        "Type": "String",
        "Description": "(Required) The ID of the Amazon EC2 instance.",
        "DefaultValue": ""
      }
    ],
    "PlatformTypes": [
      "Windows",
      "Linux"
    ],
    "DocumentType": "Automation",
    "SchemaVersion": "0.3",
    "LatestVersion": "1",
    "DefaultVersion": "1",
    "DocumentFormat": "YAML",
    "Tags": []
  }
}

```

Weitere Informationen finden Sie unter [Erstellen von Systems Manager Manager-Dokumenten](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeDocument](#) in der AWS CLI Befehlsreferenz.

describe-effective-instance-associations

Das folgende Codebeispiel zeigt die Verwendung `describe-effective-instance-associations`.

AWS CLI

Um Details zu den effektiven Verknüpfungen für eine Instanz abzurufen

Im folgenden `describe-effective-instance-associations` Beispiel werden Details zu den effektiven Verknüpfungen für eine Instanz abgerufen.

Befehl:

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
  "Associations": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "InstanceId": "i-1234567890abcdef0",
      "Content": "{\n  \"schemaVersion\": \"1.2\",\n  \"description\":\n  \"Update the Amazon SSM Agent to the latest version or specified version.\",\n  \"parameters\": {\n    \"version\": {\n      \"default\": \"\",\n      \"description\": \"(Optional) A specific version of the Amazon SSM Agent\n  to install. If not specified, the agent will be updated to the latest version.\",\n      \"type\": \"String\",\n      \"allowDowngrade\": {\n        \"default\": \"false\",\n        \"description\": \"(Optional)\n  Allow the Amazon SSM Agent service to be downgraded to an earlier version. If\n  set to false, the service can be upgraded to newer versions only (default). If\n  set to true, specify the earlier version.\",\n        \"type\": \"String\",\n        \"allowedValues\": [\n          \"true\",\n          \"false\"\n        ]\n      },\n      \"runtimeConfig\": {\n        \"aws:updateSsmAgent\": {\n          \"properties\": [\n            {\n              \"agentName\": \"amazon-ssm-agent\",\n              \"source\":\n              \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-manifest.json\",\n              \"allowDowngrade\": \"{{ allowDowngrade }}\",\n              \"targetVersion\": \"{{ version }}\"\n            }\n          ]\n        }\n      }\n    }\n  }\n  \"AssociationVersion\": \"1\"
    }
  ]
}
```

```
]
}
```

- API-Einheiten finden Sie unter [DescribeEffectiveInstanceAssociations AWS CLI-Befehlsreferenz](#).

describe-effective-patches-for-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `describe-effective-patches-for-patch-baseline`.

AWS CLI

Beispiel 1: Um alle Patches abzurufen, die durch eine benutzerdefinierte Patch-Baseline definiert sind

Im folgenden `describe-effective-patches-for-patch-baseline` Beispiel werden die durch eine benutzerdefinierte Patch-Baseline definierten Patches im aktuellen AWS Konto zurückgegeben. Beachten Sie, dass für eine benutzerdefinierte Baseline nur die ID erforderlich ist `--baseline-id`.

```
aws ssm describe-effective-patches-for-patch-baseline \
  --baseline-id "pb-08b654cf9b9681f04"
```

Ausgabe:

```
{
  "EffectivePatches": [
    {
      "Patch": {
        "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
        "ReleaseDate": 1544047205.0,
        "Title": "2018-11 Update for Windows Server 2019 for x64-based Systems (KB4470788)",
        "Description": "Install this update to resolve issues in Windows. For a complete listing of the issues that are included in this update, see the associated Microsoft Knowledge Base article for more information. After you install this item, you may have to restart your computer.",
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
        "Vendor": "Microsoft",
        "ProductFamily": "Windows",
```

```

        "Product": "WindowsServer2019",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Critical",
        "KbNumber": "KB4470788",
        "MsrcNumber": "",
        "Language": "All"
    },
    "PatchStatus": {
        "DeploymentStatus": "APPROVED",
        "ComplianceLevel": "CRITICAL",
        "ApprovalDate": 1544047205.0
    }
},
{
    "Patch": {
        "Id": "915a6b1a-f556-4d83-8f50-b2e75a9a7e58",
        "ReleaseDate": 1549994400.0,
        "Title": "2019-02 Cumulative Update for .NET Framework 3.5 and 4.7.2
for Windows Server 2019 for x64 (KB4483452)",
        "Description": "A security issue has been identified in a Microsoft
software product that could affect your system. You can help protect your system by
installing this update from Microsoft. For a complete listing of the issues that
are included in this update, see the associated Microsoft Knowledge Base article.
After you install this update, you may have to restart your system.",
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4483452",
        "Vendor": "Microsoft",
        "ProductFamily": "Windows",
        "Product": "WindowsServer2019",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Important",
        "KbNumber": "KB4483452",
        "MsrcNumber": "",
        "Language": "All"
    },
    "PatchStatus": {
        "DeploymentStatus": "APPROVED",
        "ComplianceLevel": "CRITICAL",
        "ApprovalDate": 1549994400.0
    }
},
...
],
"NextToken": "--token string truncated--"

```

```
}
```

Beispiel 2: Um alle Patches abzurufen, die durch eine AWS verwaltete Patch-Baseline definiert sind

Im folgenden `describe-effective-patches-for-patch-baseline` Beispiel werden die durch eine AWS verwaltete Patch-Baseline definierten Patches zurückgegeben. Beachten Sie, dass für eine AWS verwaltete Baseline die vollständige Baseline erforderlich ARN ist für `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
  pb-020d361a05defe4ed"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [So werden Sicherheitspatches ausgewählt](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeEffectivePatchesForPatchBaseline](#) in der AWS CLI Befehlsreferenz.

`describe-instance-associations-status`

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-associations-status`.

AWS CLI

Um den Status der Zuordnungen einer Instanz zu beschreiben

Dieses Beispiel zeigt Details zu den Zuordnungen für eine Instanz.

Befehl:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
  "InstanceAssociationStatusInfos": [
```

```

    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "Name": "AWS-GatherSoftwareInventory",
      "DocumentVersion": "1",
      "AssociationVersion": "1",
      "InstanceId": "i-1234567890abcdef0",
      "ExecutionDate": 1550501886.0,
      "Status": "Success",
      "ExecutionSummary": "1 out of 1 plugin processed, 1 success, 0 failed, 0
      timedout, 0 skipped. ",
      "AssociationName": "Inventory-Association"
    },
    {
      "AssociationId": "5c5a31f6-6dae-46f9-944c-0123456789ab",
      "Name": "AWS-UpdateSSMAgent",
      "DocumentVersion": "1",
      "AssociationVersion": "1",
      "InstanceId": "i-1234567890abcdef0",
      "ExecutionDate": 1550505828.548,
      "Status": "Success",
      "DetailedStatus": "Success",
      "AssociationName": "UpdateSSMAgent"
    }
  ]
}

```

- API-Einheiten finden Sie [DescribeInstanceAssociationsStatus](#) unter AWS CLI Befehlsreferenz.

describe-instance-information

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-information`.

AWS CLI

Beispiel 1: Um Informationen zu verwalteten Instanzen zu beschreiben

Im folgenden `describe-instance-information` Beispiel werden Details zu jeder Ihrer verwalteten Instanzen abgerufen.

```
aws ssm describe-instance-information
```

Beispiel 2: Um Informationen über eine bestimmte verwaltete Instanz zu beschreiben

Das folgende `describe-instance-information` Beispiel zeigt Details der verwalteten Instanz `i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \  
  --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Beispiel 3: Um Informationen über verwaltete Instanzen mit einem bestimmten Tag-Schlüssel zu beschreiben

Das folgende `describe-instance-information` Beispiel zeigt Details für verwaltete Instanzen, die über den Tag-Schlüssel verfügen `DEV`.

```
aws ssm describe-instance-information \  
  --filters "Key=tag-key,Values=DEV"
```

Ausgabe:

```
{  
  "InstanceInformationList": [  
    {  
      "InstanceId": "i-028ea792daEXAMPLE",  
      "PingStatus": "Online",  
      "LastPingDateTime": 1582221233.421,  
      "AgentVersion": "2.3.842.0",  
      "IsLatestVersion": true,  
      "PlatformType": "Linux",  
      "PlatformName": "SLES",  
      "PlatformVersion": "15.1",  
      "ResourceType": "EC2Instance",  
      "IPAddress": "192.0.2.0",  
      "ComputerName": "ip-198.51.100.0.us-east-2.compute.internal",  
      "AssociationStatus": "Success",  
      "LastAssociationExecutionDate": 1582220806.0,  
      "LastSuccessfulAssociationExecutionDate": 1582220806.0,  
      "AssociationOverview": {  
        "DetailedStatus": "Success",  
        "InstanceAssociationStatusAggregatedCount": {  
          "Success": 2  
        }  
      }  
    }  
  ]  
}
```



```
]
}
```

Weitere Informationen finden Sie unter [Verwaltete Instanzen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstanceInformation](#) in der AWS CLI Befehlsreferenz.

describe-instance-patch-states-for-patch-group

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-patch-states-for-patch-group`.

AWS CLI

Beispiel 1: Um den Instanzstatus für eine Patchgruppe abzurufen

Im folgenden `describe-instance-patch-states-for-patch-group` Beispiel werden Details zu den Status der Patchzusammenfassung pro Instanz für die angegebene Patchgruppe abgerufen.

```
aws ssm describe-instance-patch-states-for-patch-group \
  --patch-group "Production"
```

Ausgabe:

```
{
  "InstancePatchStates": [
    {
      "InstanceId": "i-02573cafcfEXAMPLE",
      "PatchGroup": "Production",
      "BaselineId": "pb-0c10e65780EXAMPLE",
      "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
      "OwnerInformation": "",
      "InstalledCount": 32,
      "InstalledOtherCount": 1,
      "InstalledPendingRebootCount": 0,
      "InstalledRejectedCount": 0,
      "MissingCount": 2,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 2671,
    }
  ]
}
```

```

    "NotApplicableCount": 400,
    "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
    "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
    "Operation": "Scan",
    "RebootOption": "NoReboot",
    "CriticalNonCompliantCount": 0,
    "SecurityNonCompliantCount": 1,
    "OtherNonCompliantCount": 0
  },
  {
    "InstanceId": "i-0471e04240EXAMPLE",
    "PatchGroup": "Production",
    "BaselineId": "pb-09ca3fb51fEXAMPLE",
    "SnapshotId": "05d8ffb0-1bbe-4812-ba2d-d9b7bEXAMPLE",
    "OwnerInformation": "",
    "InstalledCount": 32,
    "InstalledOtherCount": 1,
    "InstalledPendingRebootCount": 0,
    "InstalledRejectedCount": 0,
    "MissingCount": 2,
    "FailedCount": 0,
    "UnreportedNotApplicableCount": 2671,
    "NotApplicableCount": 400,
    "OperationStartTime": "2021-08-04T22:06:20.340000-07:00",
    "OperationEndTime": "2021-08-04T22:07:11.220000-07:00",
    "Operation": "Scan",
    "RebootOption": "NoReboot",
    "CriticalNonCompliantCount": 0,
    "SecurityNonCompliantCount": 1,
    "OtherNonCompliantCount": 0
  }
]
}

```

Beispiel 2: Um den Instanzstatus für eine Patch-Gruppe mit mehr als fünf fehlenden Patches abzurufen

Im folgenden `describe-instance-patch-states-for-patch-group` Beispiel werden Details zum Status der Patch-Zusammenfassung für die angegebene Patchgruppe für Instances mit mehr als fünf fehlenden Patches abgerufen.

```

aws ssm describe-instance-patch-states-for-patch-group \
  --filters Key=MissingCount,Type=GreaterThan,Values=5 \

```

```
--patch-group "Production"
```

Ausgabe:

```
{
  "InstancePatchStates": [
    {
      "InstanceId": "i-02573cafcfEXAMPLE",
      "PatchGroup": "Production",
      "BaselineId": "pb-0c10e65780EXAMPLE",
      "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
      "OwnerInformation": "",
      "InstalledCount": 46,
      "InstalledOtherCount": 4,
      "InstalledPendingRebootCount": 1,
      "InstalledRejectedCount": 1,
      "MissingCount": 7,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 232,
      "NotApplicableCount": 654,
      "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
      "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
      "Operation": "Scan",
      "RebootOption": "NoReboot",
      "CriticalNonCompliantCount": 0,
      "SecurityNonCompliantCount": 1,
      "OtherNonCompliantCount": 1
    }
  ]
}
```

Beispiel 3: Um den Instanzstatus für eine Patchgruppe mit weniger als zehn Instanzen abzurufen, für die ein Neustart erforderlich ist

Im folgenden `describe-instance-patch-states-for-patch-group` Beispiel werden Details zum Status der Patch-Zusammenfassung für die angegebene Patchgruppe für Instances mit weniger als zehn Instanzen abgerufen, die einen Neustart erfordern.

```
aws ssm describe-instance-patch-states-for-patch-group \
  --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
  --patch-group "Production"
```

Ausgabe:

```
{
  "InstancePatchStates": [
    {
      "InstanceId": "i-02573cafcfEXAMPLE",
      "BaselineId": "pb-0c10e65780EXAMPLE",
      "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
      "PatchGroup": "Production",
      "OwnerInformation": "",
      "InstalledCount": 32,
      "InstalledOtherCount": 1,
      "InstalledPendingRebootCount": 4,
      "InstalledRejectedCount": 0,
      "MissingCount": 2,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 846,
      "NotApplicableCount": 212,
      "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
      "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
      "Operation": "Scan",
      "RebootOption": "NoReboot",
      "CriticalNonCompliantCount": 0,
      "SecurityNonCompliantCount": 1,
      "OtherNonCompliantCount": 0
    }
  ]
}
```

Weitere Informationen finden Sie unter [Grundlegendes zu den Werten für den Patch-Kompatibilitätsstatus](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstancePatchStatesForPatchGroup](#) in der AWS CLI Befehlsreferenz.

describe-instance-patch-states

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-patch-states`.

AWS CLI

Um die Statusübersicht der Patches für Instanzen abzurufen

In diesem `describe-instance-patch-states` Beispiel werden die Status der Patch-Zusammenfassung für eine Instanz abgerufen.

```
aws ssm describe-instance-patch-states \
  --instance-ids "i-1234567890abcdef0"
```

Ausgabe:

```
{
  "InstancePatchStates": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "PatchGroup": "my-patch-group",
      "BaselineId": "pb-0713accee01234567",
      "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
      "CriticalNonCompliantCount": 2,
      "SecurityNonCompliantCount": 2,
      "OtherNonCompliantCount": 1,
      "InstalledCount": 123,
      "InstalledOtherCount": 334,
      "InstalledPendingRebootCount": 0,
      "InstalledRejectedCount": 0,
      "MissingCount": 1,
      "FailedCount": 2,
      "UnreportedNotApplicableCount": 11,
      "NotApplicableCount": 2063,
      "OperationStartTime": "2021-05-03T11:00:56-07:00",
      "OperationEndTime": "2021-05-03T11:01:09-07:00",
      "Operation": "Scan",
      "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
      "RebootOption": "RebootIfNeeded"
    }
  ]
}
```

Weitere Informationen finden Sie unter [About Patch Compliance](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeInstancePatchStates](#) in der AWS CLI Befehlsreferenz.

describe-instance-patches

Das folgende Codebeispiel zeigt die Verwendung `describe-instance-patches`.

AWS CLI

Beispiel 1: Um die Details zum Patch-Status für eine Instanz abzurufen

Im folgenden `describe-instance-patches` Beispiel werden Details zu den Patches für die angegebene Instanz abgerufen.

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{  
  "Patches": [  
    {  
      "Title": "2019-01 Security Update for Adobe Flash Player for Windows  
Server 2016 for x64-based Systems (KB4480979)",  
      "KBId": "KB4480979",  
      "Classification": "SecurityUpdates",  
      "Severity": "Critical",  
      "State": "Installed",  
      "InstalledTime": "2019-01-09T00:00:00+00:00"  
    },  
    {  
      "Title": "",  
      "KBId": "KB4481031",  
      "Classification": "",  
      "Severity": "",  
      "State": "InstalledOther",  
      "InstalledTime": "2019-02-08T00:00:00+00:00"  
    },  
    ...  
  ],  
  "NextToken": "--token string truncated--"  
}
```

Beispiel 2: Um eine Liste von Patches mit dem Status Missing für eine Instanz abzurufen

Im folgenden `describe-instance-patches` Beispiel werden Informationen über Patches abgerufen, die sich für die angegebene Instanz im Status `Missing` befinden.

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0" \  
  --filters Key=State,Values=Missing
```

Ausgabe:

```
{  
  "Patches": [  
    {  
      "Title": "Windows Malicious Software Removal Tool x64 - February 2019  
(KB890830)",  
      "KBId": "KB890830",  
      "Classification": "UpdateRollups",  
      "Severity": "Unspecified",  
      "State": "Missing",  
      "InstalledTime": "1970-01-01T00:00:00+00:00"  
    },  
    ...  
  ],  
  "NextToken": "--token string truncated--"  
}
```

Weitere Informationen finden Sie unter [About Patch Compliance States](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: Um eine Liste der Patches abzurufen, die seit einer bestimmten `InstalledTime` Instanz installiert wurden

Im folgenden `describe-instance-patches` Beispiel werden Informationen über Patches abgerufen, die seit einem bestimmten Zeitpunkt für die angegebene Instanz installiert wurden, indem die Verwendung von `--filters` und `--query` kombiniert wird.

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0" \  
  --filters Key=State,Values=Installed \  
  --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

Ausgabe:

```
{
  "Patches": [
    {
      "Title": "2023-03 Cumulative Update for Windows Server 2019 (1809) for
x64-based Systems (KB5023702)",
      "KBId": "KB5023702",
      "Classification": "SecurityUpdates",
      "Severity": "Critical",
      "State": "Installed",
      "InstalledTime": "2023-03-16T11:00:00+00:00"
    },
    ...
  ],
  "NextToken": "--token string truncated--"
}
```

- API-Einheiten finden Sie unter [DescribeInstancePatches AWS CLI-Befehlsreferenz](#).

describe-inventory-deletions

Das folgende Codebeispiel zeigt die Verwendung `describe-inventory-deletions`.

AWS CLI

Um Inventar zu löschen

In diesem Beispiel werden Details zu Vorgängen zum Löschen von Inventar abgerufen.

Befehl:

```
aws ssm describe-inventory-deletions
```

Ausgabe:

```
{
  "InventoryDeletions": [
    {
      "DeletionId": "6961492a-8163-44ec-aa1e-01234567850",
      "TypeName": "Custom:RackInformation",
      "DeletionStartTime": 1550254911.0,
      "LastStatus": "InProgress",
    }
  ]
}
```



```

    "LastStatusMessage": "The Delete is in progress",
    "DeletionSummary": {
      "TotalCount": 0,
      "RemainingCount": 0,
      "SummaryItems": []
    },
    "LastStatusUpdateTime": 1550254911.0
  },
  {
    "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
    "TypeName": "Custom:RackInfo",
    "DeletionStartTime": 1550254859.0,
    "LastStatus": "InProgress",
    "LastStatusMessage": "The Delete is in progress",
    "DeletionSummary": {
      "TotalCount": 1,
      "RemainingCount": 1,
      "SummaryItems": [
        {
          "Version": "1.0",
          "Count": 1,
          "RemainingCount": 1
        }
      ]
    },
    "LastStatusUpdateTime": 1550254859.0
  }
]
}

```

Um Einzelheiten zu einer bestimmten Inventarlöschung abzurufen

In diesem Beispiel werden Details für einen bestimmten Vorgang zum Löschen von Inventar abgerufen.

Befehl:

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-b1c6-987654321c4d"
```

Ausgabe:

```
{
```

```

"InventoryDeletions": [
  {
    "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
    "TypeName": "Custom:RackInfo",
    "DeletionStartTime": 1550254859.0,
    "LastStatus": "InProgress",
    "LastStatusMessage": "The Delete is in progress",
    "DeletionSummary": {
      "TotalCount": 1,
      "RemainingCount": 1,
      "SummaryItems": [
        {
          "Version": "1.0",
          "Count": 1,
          "RemainingCount": 1
        }
      ]
    },
    "LastStatusUpdateTime": 1550254859.0
  }
]
}

```

- API-Einheiten finden Sie [DescribeInventoryDeletions](#) in der AWS CLI Befehlsreferenz.

describe-maintenance-window-execution-task-invocations

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-window-execution-task-invocations`.

AWS CLI

Um die spezifischen Aufgabenaufrufe für die Ausführung einer Aufgabe in einem Wartungsfenster auszuführen

Im folgenden `describe-maintenance-window-execution-task-invocations` Beispiel werden die Aufrufe für die angegebene Aufgabe aufgeführt, die im Rahmen der Ausführung des angegebenen Wartungsfensters ausgeführt wurden.

```

aws ssm describe-maintenance-window-execution-task-invocations \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
  --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"

```

Ausgabe:

```
{
  "WindowExecutionTaskInvocationIdentities": [
    {
      "Status": "SUCCESS",
      "Parameters": "{\"documentName\": \"AWS-RunShellScript\", \"instanceIds\": [\"i-0000293ffd8c57862\"], \"parameters\": {\"commands\": [\"df\"]}, \"maxConcurrency\": \"1\", \"maxErrors\": \"1\"}\",
      "InvocationId": "e274b6e1-fe56-4e32-bd2a-8073c6381d8b",
      "StartTime": 1487692834.723,
      "EndTime": 1487692834.871,
      "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2a638355",
      "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d384503b6586"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Informationen zu Aufgaben und Aufgabenausführungen anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindowExecutionTaskInvocations](#) in der AWS CLI Befehlsreferenz.

describe-maintenance-window-execution-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-window-execution-tasks`.

AWS CLI

Um alle Aufgaben aufzulisten, die mit der Ausführung eines Wartungsfensters verbunden sind

Das folgende `ssm describe-maintenance-window-execution-tasks` Beispiel listet die Aufgaben auf, die mit der Ausführung des angegebenen Wartungsfensters verknüpft sind.

```
aws ssm describe-maintenance-window-execution-tasks \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Ausgabe:

```
{
```

```

    "WindowExecutionTaskIdentities": [
      {
        "Status": "SUCCESS",
        "TaskArn": "AWS-RunShellScript",
        "StartTime": 1487692834.684,
        "TaskType": "RUN_COMMAND",
        "EndTime": 1487692835.005,
        "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
        "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
      }
    ]
  }
}

```

Weitere Informationen finden Sie unter [Informationen zu Aufgaben und Aufgabenausführungen anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindowExecutionTasks](#) in der AWS CLI Befehlsreferenz.

describe-maintenance-window-executions

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-window-executions`.

AWS CLI

Beispiel 1: Um alle Ausführungen für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-executions` Beispiel listet alle Ausführungen für das angegebene Wartungsfenster auf.

```

aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE"

```

Ausgabe:

```

{
  "WindowExecutions": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
      "Status": "IN_PROGRESS",
      "StartTime": "2021-08-04T11:00:00.000000-07:00"
    }
  ]
}

```

```

    },
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "ff75b750-4834-4377-8f61-b3cadEXAMPLE",
      "Status": "SUCCESS",
      "StartTime": "2021-08-03T11:00:00.000000-07:00",
      "EndTime": "2021-08-03T11:37:21.450000-07:00"
    },
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
      "Status": "FAILED",
      "StatusDetails": "One or more tasks in the orchestration failed.",
      "StartTime": "2021-08-02T11:00:00.000000-07:00",
      "EndTime": "2021-08-02T11:22:36.190000-07:00"
    }
  ]
}

```

Beispiel 2: Um alle Ausführungen für ein Wartungsfenster vor einem bestimmten Datum aufzulisten

Im folgenden `describe-maintenance-window-executions` Beispiel werden alle Ausführungen für das angegebene Wartungsfenster vor dem angegebenen Datum aufgeführt.

```

aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"

```

Ausgabe:

```

{
  "WindowExecutions": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
      "Status": "FAILED",
      "StatusDetails": "One or more tasks in the orchestration failed.",
      "StartTime": "2021-08-02T11:00:00.000000-07:00",
      "EndTime": "2021-08-02T11:22:36.190000-07:00"
    }
  ]
}

```

```
]
}
```

Beispiel 3: Um alle Ausführungen für ein Wartungsfenster nach einem bestimmten Datum aufzulisten

Im folgenden `describe-maintenance-window-executions` Beispiel werden alle Ausführungen für das angegebene Wartungsfenster nach dem angegebenen Datum aufgeführt.

```
aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

Ausgabe:

```
{
  "WindowExecutions": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
      "Status": "IN_PROGRESS",
      "StartTime": "2021-08-04T11:00:00.000000-07:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Informationen zu Aufgaben und Aufgabenausführungen anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribeMaintenanceWindowExecutions AWS CLIBefehlsreferenz](#).

describe-maintenance-window-schedule

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-window-schedule`.

AWS CLI

Beispiel 1: Um bevorstehende Ausführungen für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-schedule` Beispiel listet alle bevorstehenden Ausführungen für das angegebene Wartungsfenster auf.

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-ab12cd34eEXAMPLE
```

Ausgabe:

```
{  
  "ScheduledWindowExecutions": [  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "Name": "My-First-Maintenance-Window",  
      "ExecutionTime": "2020-02-19T16:00Z"  
    },  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "Name": "My-First-Maintenance-Window",  
      "ExecutionTime": "2020-02-26T16:00Z"  
    },  
    ...  
  ]  
}
```

Beispiel 2: Um alle bevorstehenden Ausführungen für ein Wartungsfenster vor einem bestimmten Datum aufzulisten

Das folgende `describe-maintenance-window-schedule` Beispiel listet alle bevorstehenden Ausführungen für das angegebene Wartungsfenster auf, die vor dem angegebenen Datum stattfinden.

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-0ecb1226dd7b2e9a6 \  
  --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

Beispiel 3: Um alle anstehenden Ausführungen für ein Wartungsfenster nach einem bestimmten Datum aufzulisten

Das folgende `describe-maintenance-window-schedule` Beispiel listet alle bevorstehenden Ausführungen für das angegebene Wartungsfenster auf, die nach dem angegebenen Datum stattfinden.

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-0ecb1226dd7b2e9a6 \  
  --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

```
--filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindowSchedule](#) in der AWS CLI Befehlsreferenz.

describe-maintenance-window-targets

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-window-targets`.

AWS CLI

Beispiel 1: Um alle Ziele für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-targets` Beispiel listet alle Ziele für ein Wartungsfenster auf.

```
aws ssm describe-maintenance-window-targets \
  --window-id "mw-06cf17cbefEXAMPLE"
```

Ausgabe:

```
{
  "Targets": [
    {
      "ResourceType": "INSTANCE",
      "OwnerInformation": "Single instance",
      "WindowId": "mw-06cf17cbefEXAMPLE",
      "Targets": [
        {
          "Values": [
            "i-0000293ffdEXAMPLE"
          ],
          "Key": "InstanceIds"
        }
      ],
      "WindowTargetId": "350d44e6-28cc-44e2-951f-4b2c9EXAMPLE"
    },
    {
      "ResourceType": "INSTANCE",
```



```

    "OwnerInformation": "Two instances in a list",
    "WindowId": "mw-06cf17cbefEXAMPLE",
    "Targets": [
      {
        "Values": [
          "i-0000293ffdEXAMPLE",
          "i-0cb2b964d3EXAMPLE"
        ],
        "Key": "InstanceIds"
      }
    ],
    "WindowTargetId": "e078a987-2866-47be-bedd-d9cf4EXAMPLE"
  }
]
}

```

Beispiel 2: Um alle Ziele für ein Wartungsfenster aufzulisten, die einem bestimmten Besitzerinformationswert entsprechen

In diesem `describe-maintenance-window-targets` Beispiel werden alle Ziele für ein Wartungsfenster mit einem bestimmten Wert aufgeführt.

```

aws ssm describe-maintenance-window-targets \
  --window-id "mw-0ecb1226ddEXAMPLE" \
  --filters "Key=OwnerInformation,Values=CostCenter1"

```

Ausgabe:

```

{
  "Targets": [
    {
      "WindowId": "mw-0ecb1226ddEXAMPLE",
      "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
      "ResourceType": "INSTANCE",
      "Targets": [
        {
          "Key": "tag:Environment",
          "Values": [
            "Prod"
          ]
        }
      ],
      "OwnerInformation": "CostCenter1",
    }
  ]
}

```

```

        "Name": "ProdTarget1"
      }
    ]
  }

```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindowTargets](#) in der AWS CLI Befehlsreferenz.

describe-maintenance-window-tasks

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-window-tasks`.

AWS CLI

Beispiel 1: Um alle Aufgaben für ein Wartungsfenster aufzulisten

Das folgende `describe-maintenance-window-tasks` Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf.

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-06cf17cbefEXAMPLE"

```

Ausgabe:

```

{
  "Tasks": [
    {
      "WindowId": "mw-06cf17cbefEXAMPLE",
      "WindowTaskId": "018b31c3-2d77-4b9e-bd48-c91edEXAMPLE",
      "TaskArn": "AWS-RestartEC2Instance",
      "TaskParameters": {},
      "Type": "AUTOMATION",
      "Description": "Restarting EC2 Instance for maintenance",
      "MaxConcurrency": "1",
      "MaxErrors": "1",
      "Name": "My-Automation-Example-Task",
      "Priority": 0,
      "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
      "Targets": [

```

```

        {
            "Key": "WindowTargetIds",
            "Values": [
                "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
            ]
        }
    ]
},
{
    "WindowId": "mw-06cf17cbefEXAMPLE",
    "WindowTaskId": "1943dee0-0a17-4978-9bf4-3cc2fEXAMPLE",
    "TaskArn": "AWS-DisableS3BucketPublicReadWrite",
    "TaskParameters": {},
    "Type": "AUTOMATION",
    "Description": "Automation task to disable read/write access on public
S3 buckets",
    "MaxConcurrency": "10",
    "MaxErrors": "5",
    "Name": "My-Disable-S3-Public-Read-Write-Access-Automation-Task",
    "Priority": 0,
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
            ]
        }
    ]
}
]
}
}

```

Beispiel 2: Um alle Aufgaben für ein Wartungsfenster aufzulisten, das das RunPowerShellScript Befehlsdokument AWS- aufruft

Das folgende describe-maintenance-window-tasks Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf, das das AWS-RunPowerShellScript Befehlsdokument aufruft.

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \

```

```
--filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Ausgabe:

```
{
  "Tasks": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
      "TaskArn": "AWS-RunPowerShellScript",
      "Type": "RUN_COMMAND",
      "Targets": [
        {
          "Key": "WindowTargetIds",
          "Values": [
            "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
          ]
        }
      ],
      "TaskParameters": {},
      "Priority": 1,
      "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
      "MaxConcurrency": "1",
      "MaxErrors": "1",
      "Name": "MyTask"
    }
  ]
}
```

Beispiel 3: Um alle Aufgaben für ein Wartungsfenster aufzulisten, die eine Priorität von 3 haben

Das folgende `describe-maintenance-window-tasks` Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf, die den Wert `Priority` von `3` haben.

```
aws ssm describe-maintenance-window-tasks \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=Priority,Values=3"
```

Ausgabe:

```
{
```

```
"Tasks": [
  {
    "WindowId": "mw-ab12cd34eEXAMPLE",
    "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
    "TaskArn": "AWS-RunPowerShellScript",
    "Type": "RUN_COMMAND",
    "Targets": [
      {
        "Key": "WindowTargetIds",
        "Values": [
          "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
        ]
      }
    ],
    "TaskParameters": {},
    "Priority": 3,
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Name": "MyRunCommandTask"
  },
  {
    "WindowId": "mw-ab12cd34eEXAMPLE",
    "WindowTaskId": "ee45feff-ad65-4a6c-b478-5cab8EXAMPLE",
    "TaskArn": "AWS-RestartEC2Instance",
    "Type": "AUTOMATION",
    "Targets": [
      {
        "Key": "WindowTargetIds",
        "Values": [
          "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
        ]
      }
    ],
    "TaskParameters": {},
    "Priority": 3,
    "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "MaxConcurrency": "10",
    "MaxErrors": "5",
    "Name": "My-Automation-Task",
    "Description": "A description for my Automation task"
  }
]
```

```
]
}
```

Beispiel 4: Um alle Aufgaben für ein Wartungsfenster aufzulisten, die eine Priorität von 1 haben, und verwenden Sie Run Command

In diesem `describe-maintenance-window-tasks` Beispiel werden alle Aufgaben für das angegebene Wartungsfenster aufgeführt, die einen Wert `Priority` von 1 und einen Verwendungszweck haben `Run Command`.

```
aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Ausgabe:

```
{
  "Tasks": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
      "TaskArn": "AWS-RunPowerShellScript",
      "Type": "RUN_COMMAND",
      "Targets": [
        {
          "Key": "WindowTargetIds",
          "Values": [
            "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
          ]
        }
      ],
      "TaskParameters": {},
      "Priority": 1,
      "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
      "MaxConcurrency": "1",
      "MaxErrors": "1",
      "Name": "MyRunCommandTask"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindowTasks](#) unter AWS CLI Befehlsreferenz.

describe-maintenance-windows-for-target

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-windows-for-target`.

AWS CLI

Um alle Wartungsfenster aufzulisten, die einer bestimmten Instanz zugeordnet sind

Im folgenden `describe-maintenance-windows-for-target` Beispiel werden die Wartungsfenster aufgeführt, deren Ziele oder Aufgaben mit der angegebenen Instanz verknüpft sind.

```
aws ssm describe-maintenance-windows-for-target \
  --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \
  --resource-type INSTANCE
```

Ausgabe:

```
{
  "WindowIdentities": [
    {
      "WindowId": "mw-0c5ed765acEXAMPLE",
      "Name": "My-First-Maintenance-Window"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindowsForTarget](#) in der AWS CLI Befehlsreferenz.

describe-maintenance-windows

Das folgende Codebeispiel zeigt die Verwendung `describe-maintenance-windows`.

AWS CLI

Beispiel 1: Um alle Wartungsfenster aufzulisten

Das folgende `describe-maintenance-windows` Beispiel listet alle Wartungsfenster in Ihrem AWS Konto in der aktuellen Region auf.

```
aws ssm describe-maintenance-windows
```

Ausgabe:

```
{
  "WindowIdentities": [
    {
      "WindowId": "mw-0ecb1226ddEXAMPLE",
      "Name": "MyMaintenanceWindow-1",
      "Enabled": true,
      "Duration": 2,
      "Cutoff": 1,
      "Schedule": "rate(180 minutes)",
      "NextExecutionTime": "2020-02-12T23:19:20.596Z"
    },
    {
      "WindowId": "mw-03eb9db428EXAMPLE",
      "Name": "MyMaintenanceWindow-2",
      "Enabled": true,
      "Duration": 3,
      "Cutoff": 1,
      "Schedule": "rate(7 days)",
      "NextExecutionTime": "2020-02-17T23:22:00.956Z"
    }
  ]
}
```

Beispiel 2: Um alle aktivierten Wartungsfenster aufzulisten

Das folgende `describe-maintenance-windows` Beispiel listet alle aktivierten Wartungsfenster auf.

```
aws ssm describe-maintenance-windows \
  --filters "Key=Enabled,Values=true"
```


Beispiel 3: Um Wartungsfenster aufzulisten, die einem bestimmten Namen entsprechen

In diesem `describe-maintenance-windows` Beispiel werden alle Wartungsfenster mit dem angegebenen Namen aufgeführt.

```
aws ssm describe-maintenance-windows \  
  --filters "Key=Name,Values=MyMaintenanceWindow"
```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeMaintenanceWindows](#) in der AWS CLI Befehlsreferenz.

describe-ops-items

Das folgende Codebeispiel zeigt die Verwendung `describe-ops-items`.

AWS CLI

Um eine Reihe von aufzulisten OpsItems

Im folgenden `describe-ops-items` Beispiel wird eine Liste aller offenen Konten OpsItems in Ihrem AWS Konto angezeigt.

```
aws ssm describe-ops-items \  
  --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Ausgabe:

```
{  
  "OpsItemSummaries": [  
    {  
      "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/  
fbf77cbe264a33509569f23e4EXAMPLE",  
      "CreatedTime": "2020-03-14T17:02:46.375000-07:00",  
      "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-  
Role/fbf77cbe264a33509569f23e4EXAMPLE",  
      "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",  
      "Source": "SSM",  
      "Status": "Open",  
      "OpsItemId": "oi-7cfc5EXAMPLE",  
      "Title": "SSM Maintenance Window execution failed",
```

```

    "OperationalData": {
      "/aws/dedup": {
        "Value": "{\\"dedupString\\":\\"SSMOpsItems-SSM-maintenance-window-
execution-failed\\"}",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[{\\"arn\\":\\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\\"}]",
        "Type": "SearchableString"
      }
    },
    "Category": "Availability",
    "Severity": "3"
  },
  {
    "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
    "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
    "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
    "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
    "Source": "EC2",
    "Status": "Open",
    "OpsItemId": "oi-6f966EXAMPLE",
    "Title": "EC2 instance stopped",
    "OperationalData": {
      "/aws/automations": {
        "Value": "[ { \\"automationType\\": \\"AWS:SSM:Automation\\",
\\"automationId\\": \\"AWS-RestartEC2Instance\\" } ]",
        "Type": "SearchableString"
      },
      "/aws/dedup": {
        "Value": "{\\"dedupString\\":\\"SSMOpsItems-EC2-instance-stopped
\\"}",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[{\\"arn\\":\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\\"}]",
        "Type": "SearchableString"
      }
    },
    "Category": "Availability",

```

```

        "Severity": "3"
      }
    ]
  }

```

Weitere Informationen finden Sie unter [Arbeiten mit OpsItems](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeOpsItems](#) in der AWS CLI Befehlsreferenz.

describe-parameters

Das folgende Codebeispiel zeigt die Verwendung `describe-parameters`.

AWS CLI

Beispiel 1: Um alle Parameter aufzulisten

Das folgende `describe-parameters` Beispiel listet alle Parameter im AWS Girokonto und in der Region auf.

```
aws ssm describe-parameters
```

Ausgabe:

```

{
  "Parameters": [
    {
      "Name": "MySecureStringParameter",
      "Type": "SecureString",
      "KeyId": "alias/aws/ssm",
      "LastModifiedDate": 1582155479.205,
      "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/Admin/Richard-Roe-Managed",
      "Description": "This is a SecureString parameter",
      "Version": 2,
      "Tier": "Advanced",
      "Policies": [
        {
          "PolicyText": "{\"Type\":\"Expiration\",\"Version\":\"1.0\", \"Attributes\":{\"Timestamp\":\"2020-07-07T22:30:00Z\"}}",
          "PolicyType": "Expiration",
          "PolicyStatus": "Pending"
        }
      ]
    }
  ]
}

```

```

        },
        {
            "PolicyText": "{\"Type\":\"ExpirationNotification\",\"Version\":
\\\"1.0\\\",\\\"Attributes\\\":{\\\"Before\\\":\\\"12\\\",\\\"Unit\\\":\\\"Hours\\\"}}\",
            "PolicyType": "ExpirationNotification",
            "PolicyStatus": "Pending"
        }
    ]
},
{
    "Name": "MyStringListParameter",
    "Type": "StringList",
    "LastModifiedDate": 1582154764.222,
    "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
    "Description": "This is a StringList parameter",
    "Version": 1,
    "Tier": "Standard",
    "Policies": []
},
{
    "Name": "MyStringParameter",
    "Type": "String",
    "LastModifiedDate": 1582154711.976,
    "LastModifiedUser": "arn:aws:iam::111222333444:user/Alejandro-Rosalez",
    "Description": "This is a String parameter",
    "Version": 1,
    "Tier": "Standard",
    "Policies": []
},
{
    "Name": "latestAmi",
    "Type": "String",
    "LastModifiedDate": 1580862415.521,
    "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/lambda-ssm-
role/Automation-UpdateSSM-Param",
    "Version": 3,
    "Tier": "Standard",
    "Policies": []
}
]
}

```

Beispiel 2: Um alle Parameter aufzulisten, die bestimmten Metadaten entsprechen

In diesem `describe-parameters` Beispiel werden alle Parameter aufgeführt, die einem Filter entsprechen.

```
aws ssm describe-parameters --filters „Key=Type, Values=“ StringList
```

Ausgabe:

```
{
  "Parameters": [
    {
      "Name": "MyStringListParameter",
      "Type": "StringList",
      "LastModifiedDate": 1582154764.222,
      "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
      "Description": "This is a StringList parameter",
      "Version": 1,
      "Tier": "Standard",
      "Policies": []
    }
  ]
}
```

Weitere Informationen finden Sie unter [Suchen nach Systems Manager Manager-Parametern](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeParameters](#) in der AWS CLI Befehlsreferenz.

describe-patch-baselines

Das folgende Codebeispiel zeigt die Verwendung `describe-patch-baselines`.

AWS CLI

Beispiel 1: Um alle Patch-Baselines aufzulisten

Im folgenden `describe-patch-baselines` Beispiel werden Details für alle Patch-Baselines in Ihrem Konto in der aktuellen Region abgerufen.

```
aws ssm describe-patch-baselines
```

Ausgabe:

```
{
```

```

    "BaselineIdentities": [
      {
        "BaselineName": "AWS-SuseDefaultPatchBaseline",
        "DefaultBaseline": true,
        "BaselineDescription": "Default Patch Baseline for Suse Provided by
AWS.",
        "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0123fdb36e334a3b2",
        "OperatingSystem": "SUSE"
      },
      {
        "BaselineName": "AWS-DefaultPatchBaseline",
        "DefaultBaseline": false,
        "BaselineDescription": "Default Patch Baseline Provided by AWS.",
        "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed",
        "OperatingSystem": "WINDOWS"
      },
      ...
      {
        "BaselineName": "MyWindowsPatchBaseline",
        "DefaultBaseline": true,
        "BaselineDescription": "My patch baseline for EC2 instances for Windows
Server",
        "BaselineId": "pb-0ad00e0dd7EXAMPLE",
        "OperatingSystem": "WINDOWS"
      }
    ]
  }

```

Beispiel 2: Um alle Patch-Baselines aufzulisten, die bereitgestellt werden von AWS

Das folgende `describe-patch-baselines` Beispiel listet alle Patch-Baselines auf, die von bereitgestellt werden. AWS

```

aws ssm describe-patch-baselines \
  --filters "Key=OWNER,Values=[AWS]"

```

Beispiel 3: Um alle Patch-Baselines aufzulisten, die Ihnen gehören

Im folgenden `describe-patch-baselines` Beispiel werden alle benutzerdefinierten Patch-Baselines aufgeführt, die in Ihrem Konto in der aktuellen Region erstellt wurden.

```
aws ssm describe-patch-baselines \  
  --filters "Key=OWNER,Values=[Self]"
```

Weitere Informationen finden Sie unter [Über vordefinierte und benutzerdefinierte Patch-Baselines](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribePatchBaselines AWS CLI Befehlsreferenz](#).

describe-patch-group-state

Das folgende Codebeispiel zeigt die Verwendung `describe-patch-group-state`.

AWS CLI

Um den Status einer Patch-Gruppe abzurufen

Im folgenden `describe-patch-group-state` Beispiel wird die allgemeine Zusammenfassung der Patch-Konformität für eine Patchgruppe abgerufen.

```
aws ssm describe-patch-group-state \  
  --patch-group "Production"
```

Ausgabe:

```
{  
  "Instances": 21,  
  "InstancesWithCriticalNonCompliantPatches": 1,  
  "InstancesWithFailedPatches": 2,  
  "InstancesWithInstalledOtherPatches": 3,  
  "InstancesWithInstalledPatches": 21,  
  "InstancesWithInstalledPendingRebootPatches": 2,  
  "InstancesWithInstalledRejectedPatches": 1,  
  "InstancesWithMissingPatches": 3,  
  "InstancesWithNotApplicablePatches": 4,  
  "InstancesWithOtherNonCompliantPatches": 1,  
  "InstancesWithSecurityNonCompliantPatches": 1,  
  "InstancesWithUnreportedNotApplicablePatches": 2  
}
```

Weitere Informationen finden Sie unter About patch groups < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html> >__ und [Understanding Patch Compliance](#) State Values im Systems Manager Manager-Benutzerhandbuch.AWS

- APIAWS CLI Einzelheiten [DescribePatchGroupState](#)finden Sie in der Befehlsreferenz.

describe-patch-groups

Das folgende Codebeispiel zeigt die Verwendungdescribe-patch-groups.

AWS CLI

Um Patch-Gruppenregistrierungen anzuzeigen

Das folgende describe-patch-groups Beispiel listet die Patch-Gruppenregistrierungen auf.

```
aws ssm describe-patch-groups
```

Ausgabe:

```
{
  "Mappings": [
    {
      "PatchGroup": "Production",
      "BaselineIdentity": {
        "BaselineId": "pb-0123456789abcdef0",
        "BaselineName": "ProdPatching",
        "OperatingSystem": "WINDOWS",
        "BaselineDescription": "Patches for Production",
        "DefaultBaseline": false
      }
    },
    {
      "PatchGroup": "Development",
      "BaselineIdentity": {
        "BaselineId": "pb-0713accee01234567",
        "BaselineName": "DevPatching",
        "OperatingSystem": "WINDOWS",
        "BaselineDescription": "Patches for Development",
        "DefaultBaseline": true
      }
    }
  ],
}
```



```

    ...
  ]
}

```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>>__ und [Hinzufügen einer Patchgruppe zu einer Patch-Baseline](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribePatchGroups](#) in AWS CLI der Befehlsreferenz.

describe-patch-properties

Das folgende Codebeispiel zeigt die Verwendung `describe-patch-properties`.

AWS CLI

Um die Verfügbarkeit von Amazon Linux-Patches aufzulisten

Das folgende `describe-patch-properties` Beispiel zeigt eine Liste der Amazon Linux-Produkte, für die Patches in Ihrem AWS Konto verfügbar sind.

```

aws ssm describe-patch-properties \
  --operating-system AMAZON_LINUX \
  --property PRODUCT

```

Ausgabe:

```

{
  "Properties": [
    {
      "Name": "AmazonLinux2012.03"
    },
    {
      "Name": "AmazonLinux2012.09"
    },
    {
      "Name": "AmazonLinux2013.03"
    },
    {
      "Name": "AmazonLinux2013.09"
    }
  ]
}

```

```
    },
    {
      "Name": "AmazonLinux2014.03"
    },
    {
      "Name": "AmazonLinux2014.09"
    },
    {
      "Name": "AmazonLinux2015.03"
    },
    {
      "Name": "AmazonLinux2015.09"
    },
    {
      "Name": "AmazonLinux2016.03"
    },
    {
      "Name": "AmazonLinux2016.09"
    },
    {
      "Name": "AmazonLinux2017.03"
    },
    {
      "Name": "AmazonLinux2017.09"
    },
    {
      "Name": "AmazonLinux2018.03"
    }
  ]
}
```

Weitere Informationen finden Sie unter [About Patch Baselines](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DescribePatchProperties AWS CLI](#) Befehlsreferenz.

describe-sessions

Das folgende Codebeispiel zeigt die Verwendung `describe-sessions`.

AWS CLI

Beispiel 1: Um alle aktiven Session Manager-Sitzungen aufzulisten

In diesem `describe-sessions` Beispiel wird eine Liste der aktiven Sitzungen abgerufen, die zuletzt in den letzten 30 Tagen erstellt wurden (sowohl verbundene als auch getrennte Sitzungen), die vom angegebenen Benutzer gestartet wurden. Dieser Befehl gibt nur Ergebnisse für Verbindungen zu Zielen zurück, die mit Session Manager initiiert wurden. Verbindungen, die auf andere Weise hergestellt wurden, wie z. B. Remotedesktopverbindungen oder, werden nicht aufgeführtSSH.

```
aws ssm describe-sessions \
  --state "Active" \
  --filters "key=owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-Rodriguez"
```

Ausgabe:

```
{
  "Sessions": [
    {
      "SessionId": "John-07a16060613c408b5",
      "Target": "i-1234567890abcdef0",
      "Status": "Connected",
      "StartDate": 1550676938.352,
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-Rodriguez",
      "OutputUrl": {}
    },
    {
      "SessionId": "John-01edf534b8b56e8eb",
      "Target": "i-9876543210abcdef0",
      "Status": "Connected",
      "StartDate": 1550676842.194,
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-Rodriguez",
      "OutputUrl": {}
    }
  ]
}
```

Beispiel 2: Um alle beendeten Session Manager-Sitzungen aufzulisten

In diesem `describe-sessions` Beispiel wird für alle Benutzer eine Liste der zuletzt beendeten Sitzungen der letzten 30 Tage abgerufen.

```
aws ssm describe-sessions \  
  --state "History"
```

Ausgabe:

```
{  
  "Sessions": [  
    {  
      "SessionId": "Mary-Major-0022b1eb2b0d9e3bd",  
      "Target": "i-1234567890abcdef0",  
      "Status": "Terminated",  
      "StartDate": 1550520701.256,  
      "EndDate": 1550521931.563,  
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Mary-  
Major"  
    },  
    {  
      "SessionId": "Jane-Roe-0db53f487931ed9d4",  
      "Target": "i-9876543210abcdef0",  
      "Status": "Terminated",  
      "StartDate": 1550161369.149,  
      "EndDate": 1550162580.329,  
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Jane-Roe"  
    },  
    ...  
  ],  
  "NextToken": "--token string truncated--"  
}
```

Weitere Informationen finden Sie unter [Sitzungsverlauf anzeigen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [DescribeSessions](#) in der AWS CLI Befehlsreferenz.

disassociate-ops-item-related-item

Das folgende Codebeispiel zeigt die Verwendung `disassociate-ops-item-related-item`.

AWS CLI

Um eine zugehörige Artikelzuordnung zu löschen

Im folgenden `disassociate-ops-item-related-item` Beispiel wird die Verknüpfung zwischen dem OpsItem und einem verwandten Element gelöscht.

```
aws ssm disassociate-ops-item-related-item \  
  --ops-item-id "oi-f99f2EXAMPLE" \  
  --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Incident Manager-Vorfällen OpsCenter im AWS Systems Manager Manager-Benutzerhandbuch](#).

- API-Einheiten finden Sie [DisassociateOpsItemRelatedItem](#) in der AWS CLI Befehlsreferenz.

get-automation-execution

Das folgende Codebeispiel zeigt die Verwendung `get-automation-execution`.

AWS CLI

Um Details zu einer Automatisierungsausführung anzuzeigen

Im folgenden `get-automation-execution` Beispiel werden detaillierte Informationen zu einer Automatisierungsausführung angezeigt.

```
aws ssm get-automation-execution \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{  
  "AutomationExecution": {  
    "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",  
    "DocumentName": "AWS-StartEC2Instance",  
    "DocumentVersion": "1",  
    "ExecutionStartTime": 1583737233.748,  
    "ExecutionEndTime": 1583737234.719,  
    "AutomationExecutionStatus": "Success",  
    "StepExecutions": [  
      {  
        "StepName": "startInstances",
```

```

    "Action": "aws:changeInstanceState",
    "ExecutionStartTime": 1583737234.134,
    "ExecutionEndTime": 1583737234.672,
    "StepStatus": "Success",
    "Inputs": {
      "DesiredState": "\"running\"",
      "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
    },
    "Outputs": {
      "InstanceStates": [
        "running"
      ]
    },
    "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
    "OverriddenParameters": {}
  }
],
"StepExecutionsTruncated": false,
"Parameters": {
  "AutomationAssumeRole": [
    ""
  ],
  "InstanceId": [
    "i-0cb99161f6EXAMPLE"
  ]
},
"Outputs": {},
"Mode": "Auto",
"ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
"Targets": [],
"ResolvedTargets": {
  "ParameterValues": [],
  "Truncated": false
}
}
}

```

Weitere Informationen finden Sie unter [Exemplarische Vorgehensweise: Ein Linux patchen AMI \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetAutomationExecution](#) in der AWS CLI Befehlsreferenz.

get-calendar-state

Das folgende Codebeispiel zeigt die Verwendung `get-calendar-state`.

AWS CLI

Beispiel 1: Um den aktuellen Status eines Änderungskalenders abzurufen

In diesem `get-calendar-state` Beispiel wird der Status eines Kalenders zum aktuellen Zeitpunkt zurückgegeben. Da in dem Beispiel keine Uhrzeit angegeben ist, wird der aktuelle Status des Kalenders gemeldet.

```
aws ssm get-calendar-state \
  --calendar-names "MyCalendar"
```

Ausgabe:

```
{
  "State": "OPEN",
  "AtTime": "2020-02-19T22:28:51Z",
  "NextTransitionTime": "2020-02-24T21:15:19Z"
}
```

Beispiel 2: Um den Status eines Änderungskalenders zu einem bestimmten Zeitpunkt abzurufen

In diesem `get-calendar-state` Beispiel wird der Status eines Kalenders zum angegebenen Zeitpunkt zurückgegeben.

```
aws ssm get-calendar-state \
  --calendar-names "MyCalendar" \
  --at-time "2020-07-19T21:15:19Z"
```

Ausgabe:

```
{
  "State": "CLOSED",
  "AtTime": "2020-07-19T21:15:19Z"
}
```

Weitere Informationen finden [Sie unter Get the State of the Change Calendar](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetCalendarState](#) unter AWS CLI Befehlsreferenz.

get-command-invocation

Das folgende Codebeispiel zeigt die Verwendung `get-command-invocation`.

AWS CLI

Um die Details eines Befehlsaufrufs anzuzeigen

Das folgende `get-command-invocation` Beispiel listet alle Aufrufe des angegebenen Befehls auf der angegebenen Instanz auf.

```
aws ssm get-command-invocation \  
  --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \  
  --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{  
  "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",  
  "InstanceId": "i-1234567890abcdef0",  
  "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",  
  "DocumentName": "AWS-UpdateSSMAgent",  
  "DocumentVersion": "",  
  "PluginName": "aws:updateSsmAgent",  
  "ResponseCode": 0,  
  "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",  
  "ExecutionElapsedTime": "PT0.091S",  
  "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",  
  "Status": "Success",  
  "StatusDetails": "Success",  
  "StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest  
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/  
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,  
update skipped\n",  
  "StandardOutputUrl": "",  
  "StandardErrorContent": "",  
  "StandardErrorUrl": "",  
  "CloudWatchOutputConfig": {  
    "CloudWatchLogGroupName": "",
```



```
    "CloudWatchOutputEnabled": false
  }
}
```

Weitere Informationen finden Sie unter [Understanding Command Statuses](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetCommandInvocation AWS CLI](#) Befehlsreferenz.

get-connection-status

Das folgende Codebeispiel zeigt die Verwendung `get-connection-status`.

AWS CLI

Um den Verbindungsstatus einer verwalteten Instanz anzuzeigen

In diesem `get-connection-status` Beispiel wird der Verbindungsstatus der angegebenen verwalteten Instanz zurückgegeben.

```
aws ssm get-connection-status \
  --target i-1234567890abcdef0
```

Ausgabe:

```
{
  "Target": "i-1234567890abcdef0",
  "Status": "connected"
}
```

- API-Einheiten finden Sie [GetConnectionStatus](#) unter AWS CLI Befehlsreferenz.

get-default-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `get-default-patch-baseline`.

AWS CLI

Beispiel 1: Um die Standard-Windows-Patch-Baseline anzuzeigen

Im folgenden `get-default-patch-baseline` Beispiel werden Details für die Standard-Patch-Baseline für Windows Server abgerufen.

```
aws ssm get-default-patch-baseline
```

Ausgabe:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Beispiel 2: So zeigen Sie die Standard-Patch-Baseline für Amazon Linux an

Im folgenden `get-default-patch-baseline` Beispiel werden Details für die Standard-Patch-Baseline für Amazon Linux abgerufen.

```
aws ssm get-default-patch-baseline \
  --operating-system AMAZON_LINUX
```

Ausgabe:

```
{
  "BaselineId": "pb-047c6eb9c8fc12345",
  "OperatingSystem": "AMAZON_LINUX"
}
```

Weitere Informationen finden Sie unter [Über vordefinierte und benutzerdefinierte Patch-Baselines < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html) und [Eine bestehende Patch-Baseline als Standard festlegen](#) im Systems Manager Manager-Benutzerhandbuch.AWS

- APIAWS CLI Einzelheiten [GetDefaultPatchBaseline](#) finden Sie in der Befehlsreferenz.

get-deployable-patch-snapshot-for-instance

Das folgende Codebeispiel zeigt die Verwendung `get-deployable-patch-snapshot-for-instance`.

AWS CLI

Um den aktuellen Snapshot für die Patch-Baseline abzurufen, verwendet eine Instanz

Im folgenden `get-deployable-patch-snapshot-for-instance` Beispiel werden Details für den aktuellen Snapshot für die angegebene Patch-Baseline abgerufen, die von einer Instanz verwendet wird. Dieser Befehl muss von der Instanz aus mit den Anmeldeinformationen der Instanz ausgeführt werden. Um sicherzustellen, dass er die Instance-Anmeldeinformationen verwendet, führen Sie ihn aus `aws configure` und geben Sie nur die Region Ihrer Instance an. Lassen Sie die `Secret Key` Felder `Access Key` und leer.

Tipp: Verwenden Sie `uuidgen`, um eine zu generieren `snapshot-id`.

```
aws ssm get-deployable-patch-snapshot-for-instance \  
  --instance-id "i-1234567890abcdef0" \  
  --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Ausgabe:

```
{  
  "InstanceId": "i-1234567890abcdef0",  
  "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",  
  "Product": "AmazonLinux2018.03",  
  "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-  
east-1.s3.amazonaws.com/  
ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/  
AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-Algorithm=AWS4-HMAC-  
SHA256&X-Amz-Date=20190215T164031Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-  
Amz-Credential=AKIAJ5C56P35AEBRX2QQ%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-  
Amz-Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"  
}
```

Weitere Informationen finden Sie unter [Parametername: Snapshot-ID](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetDeployablePatchSnapshotForInstance](#) in der AWS CLI Befehlsreferenz.

get-document

Das folgende Codebeispiel zeigt die Verwendung `get-document`.

AWS CLI

Um den Inhalt des Dokuments abzurufen

Im folgenden `get-document` Beispiel wird der Inhalt eines Systems Manager Manager-Dokuments angezeigt.

```
aws ssm get-document \
  --name "AWS-RunShellScript"
```

Ausgabe:

```
{
  "Name": "AWS-RunShellScript",
  "DocumentVersion": "1",
  "Status": "Active",
  "Content": "{\n  \"schemaVersion\": \"1.2\",\n  \"description\": \"Run a\n  shell script or specify the commands to run.\",\n  \"parameters\": {\n\n  \"commands\": {\n    \"type\": \"StringList\",\n    \"description\n  \": \"(Required) Specify a shell script or a command to run.\",\n    \"minItems\": 1,\n    \"displayType\": \"textarea\"\n  },\n  \"workingDirectory\": {\n    \"type\": \"String\",\n    \"default\n  \": \"\",\n    \"description\": \"(Optional) The path to the working\n  directory on your instance.\",\n    \"maxChars\": 4096\n  },\n  \"executionTimeout\": {\n    \"type\": \"String\",\n    \"default\n  \": \"3600\",\n    \"description\": \"(Optional) The time in seconds for a\n  command to complete before it is considered to have failed. Default is 3600 (1\n  hour). Maximum is 172800 (48 hours).\",\n    \"allowedPattern\": \"([1-9]\n  [0-9]{0,4})|(1[0-6][0-9]{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n  }\n  },\n  \"runtimeConfig\": {\n    \"aws:runShellScript\": {\n      \"properties\": [\n        {\n          \"id\":\n        \": \"0.aws:runShellScript\",\n          \"runCommand\": \"{{ commands }}\",\n          \"workingDirectory\": \"{{ workingDirectory }}\",\n          \"timeoutSeconds\": \"{{ executionTimeout }}\"\n        }\n      ]\n    }\n  },\n  \"DocumentType\": \"Command\",\n  \"DocumentFormat\": \"JSON\"\n}
```

Weitere Informationen finden Sie unter [AWS Systems Manager Manager-Dokumente](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetDocument](#) in der AWS CLI Befehlsreferenz.

get-inventory-schema

Das folgende Codebeispiel zeigt die Verwendung `get-inventory-schema`.

AWS CLI

Um Ihr Inventarschema anzuzeigen

In diesem Beispiel wird eine Liste von Inventartypnamen für das Konto zurückgegeben.

Befehl:

```
aws ssm get-inventory-schema
```

Ausgabe:

```
{
  "Schemas": [
    {
      "TypeName": "AWS:AWSComponent",
      "Version": "1.0",
      "Attributes": [
        {
          "Name": "Name",
          "DataType": "STRING"
        },
        {
          "Name": "ApplicationType",
          "DataType": "STRING"
        },
        {
          "Name": "Publisher",
          "DataType": "STRING"
        },
        {
          "Name": "Version",
          "DataType": "STRING"
        },
        {
          "Name": "InstalledTime",
          "DataType": "STRING"
        },
        {
          "Name": "Architecture",
```

```

        "DataType": "STRING"
      },
      {
        "Name": "URL",
        "DataType": "STRING"
      }
    ]
  },
  ...
],
"NextToken": "--token string truncated--"
}

```

Um das Inventarschema für einen bestimmten Inventartyp anzuzeigen

In diesem Beispiel wird das Inventarschema für den AWS Inventartyp „AWS Komponente“ zurückgegeben.

Befehl:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- API-Einheiten finden Sie [GetInventorySchema](#) in der AWS CLI Befehlsreferenz.

get-inventory

Das folgende Codebeispiel zeigt die Verwendung `get-inventory`.

AWS CLI

Um Ihr Inventar einzusehen

In diesem Beispiel werden die benutzerdefinierten Metadaten für Ihr Inventar abgerufen.

Befehl:

```
aws ssm get-inventory
```

Ausgabe:

```
{
  "Entities": [
```

```

    {
      "Data": {
        "AWS:InstanceInformation": {
          "Content": [
            {
              "ComputerName": "ip-172-31-44-222.us-
west-2.compute.internal",
              "InstanceId": "i-0cb2b964d3e14fd9f",
              "IpAddress": "172.31.44.222",
              "AgentType": "amazon-ssm-agent",
              "ResourceType": "EC2Instance",
              "AgentVersion": "2.0.672.0",
              "PlatformVersion": "2016.09",
              "PlatformName": "Amazon Linux AMI",
              "PlatformType": "Linux"
            }
          ],
          "TypeName": "AWS:InstanceInformation",
          "SchemaVersion": "1.0",
          "CaptureTime": "2017-02-20T18:03:58Z"
        }
      },
      "Id": "i-0cb2b964d3e14fd9f"
    }
  ]
}

```

- API-Einheiten finden Sie [GetInventory](#) in der AWS CLI Befehlsreferenz.

get-maintenance-window-execution-task-invocation

Das folgende Codebeispiel zeigt die Verwendung `get-maintenance-window-execution-task-invocation`.

AWS CLI

Um Informationen über den Aufgabenaufwurf eines Wartungsfensters abzurufen

Das folgende `get-maintenance-window-execution-task-invocation` Beispiel listet Informationen über den angegebenen Aufgabenaufwurf auf, der Teil der Ausführung des angegebenen Wartungsfensters ist.

```
aws ssm get-maintenance-window-execution-task-invocation \
```

```
--window-execution-id "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE" \  
--task-id "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE" \  
--invocation-id "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE"
```

Ausgabe:

```
{  
  "Status": "SUCCESS",  
  "Parameters": {"comment\\":\\"\\", "documentName\\":\\"AWS-RunPowerShellScript\\",  
  "instanceIds\\":["i-1234567890EXAMPLE"], "maxConcurrency\\":\\"1\\", "maxErrors\\":  
  "1\\", "parameters\\":{"executionTimeout\\":["3600"], "workingDirectory\\":["\\""],  
  "commands\\":["echo Hello"]}, "timeoutSeconds\\":600},  
  "ExecutionId": "03b6baa0-5460-4e15-83f2-ea685EXAMPLE",  
  "InvocationId": "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE",  
  "StartTime": 1549998326.421,  
  "TaskType": "RUN_COMMAND",  
  "EndTime": 1550001931.784,  
  "WindowExecutionId": "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE",  
  "StatusDetails": "Failed",  
  "TaskExecutionId": "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Informationen zu Aufgaben und Aufgabenausführungen anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMaintenanceWindowExecutionTaskInvocation](#) in der AWS CLI Befehlsreferenz.

get-maintenance-window-execution-task

Das folgende Codebeispiel zeigt die Verwendung `get-maintenance-window-execution-task`.

AWS CLI

Um Informationen über die Ausführung einer Aufgabe in einem Wartungsfenster abzurufen

Im folgenden `get-maintenance-window-execution-task` Beispiel werden Informationen zu einer Aufgabe aufgeführt, die Teil der Ausführung des angegebenen Wartungsfensters ist.

```
aws ssm get-maintenance-window-execution-task \  
--window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \  
--task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```


Ausgabe:

```
{
  "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
  "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE",
  "TaskArn": "AWS-RunPatchBaseline",
  "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
  "Type": "RUN_COMMAND",
  "TaskParameters": [
    {
      "BaselineOverride": {
        "Values": [
          ""
        ]
      },
      "InstallOverrideList": {
        "Values": [
          ""
        ]
      },
      "Operation": {
        "Values": [
          "Scan"
        ]
      },
      "RebootOption": {
        "Values": [
          "RebootIfNeeded"
        ]
      },
      "SnapshotId": {
        "Values": [
          "{{ aws:ORCHESTRATION_ID }}"
        ]
      },
      "aws:InstanceId": {
        "Values": [
          "i-02573cafcfEXAMPLE",
          "i-0471e04240EXAMPLE",
          "i-07782c72faEXAMPLE"
        ]
      }
    }
  ]
}
```

```
],  
  "Priority": 1,  
  "MaxConcurrency": "1",  
  "MaxErrors": "3",  
  "Status": "SUCCESS",  
  "StartTime": "2021-08-04T11:45:35.088000-07:00",  
  "EndTime": "2021-08-04T11:53:09.079000-07:00"  
}
```

Weitere Informationen finden Sie unter [Informationen zu Aufgaben und Aufgabenausführungen anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetMaintenanceWindowExecutionTask AWS CLI](#) Befehlsreferenz.

get-maintenance-window-execution

Das folgende Codebeispiel zeigt die Verwendung get-maintenance-window-execution.

AWS CLI

Um Informationen über die Ausführung einer Aufgabe in einem Wartungsfenster abzurufen

Das folgende get-maintenance-window-execution Beispiel listet Informationen über eine Aufgabe auf, die im Rahmen der Ausführung des angegebenen Wartungsfensters ausgeführt wurde.

```
aws ssm get-maintenance-window-execution \  
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Ausgabe:

```
{  
  "Status": "SUCCESS",  
  "TaskIds": [  
    "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"  
  ],  
  "StartTime": 1487692834.595,  
  "EndTime": 1487692835.051,  
  "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",  
}
```

Weitere Informationen finden Sie unter [Informationen zu Aufgaben und Aufgabenausführungen anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMaintenanceWindowExecution](#) in der AWS CLI Befehlsreferenz.

get-maintenance-window-task

Das folgende Codebeispiel zeigt die Verwendung `get-maintenance-window-task`.

AWS CLI

Um Informationen über eine Aufgabe im Wartungsfenster abzurufen

Im folgenden `get-maintenance-window-task` Beispiel werden Details zur angegebenen Aufgabe im Wartungsfenster abgerufen.

```
aws ssm get-maintenance-window-task \
  --window-id mw-0c5ed765acEXAMPLE \
  --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

Ausgabe:

```
{
  "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
  ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
  "MaxErrors": "1",
  "TaskArn": "AWS-RunPowerShellScript",
  "MaxConcurrency": "1",
  "WindowTaskId": "0e842a8d-2d44-4886-bb62-af8dcEXAMPLE",
  "TaskParameters": {},
  "Priority": 1,
  "TaskInvocationParameters": {
    "RunCommand": {
      "Comment": "",
      "TimeoutSeconds": 600,
      "Parameters": {
        "commands": [
          "echo Hello"
        ],
        "executionTimeout": [
          "3600"
        ],
        "workingDirectory": [
```

```

        ""
    ]
}
},
"WindowId": "mw-0c5ed765acEXAMPLE",
"TaskType": "RUN_COMMAND",
"Targets": [
    {
        "Values": [
            "84c818da-b619-4d3d-9651-946f3EXAMPLE"
        ],
        "Key": "WindowTargetIds"
    }
],
"Name": "ExampleTask"
}

```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMaintenanceWindowTask](#) in der AWS CLI Befehlsreferenz.

get-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `get-maintenance-window`.

AWS CLI

Um Informationen über ein Wartungsfenster zu erhalten

Im folgenden `get-maintenance-window` Beispiel werden Details zum angegebenen Wartungsfenster abgerufen.

```
aws ssm get-maintenance-window \
  --window-id "mw-03eb9db428EXAMPLE"
```

Ausgabe:

```
{
  "AllowUnassociatedTargets": true,
  "CreateDate": 1515006912.957,
```

```

    "Cutoff": 1,
    "Duration": 6,
    "Enabled": true,
    "ModifiedDate": 2020-01-01T10:04:04.099Z,
    "Name": "My-Maintenance-Window",
    "Schedule": "rate(3 days)",
    "WindowId": "mw-03eb9db428EXAMPLE",
    "NextExecutionTime": "2020-02-25T00:08:15.099Z"
  }

```

Weitere Informationen finden Sie unter [Informationen zu Wartungsfenstern anzeigen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetMaintenanceWindow](#) unter AWS CLI Befehlsreferenz.

get-ops-item

Das folgende Codebeispiel zeigt die Verwendung `get-ops-item`.

AWS CLI

Um Informationen zu einem anzuzeigen OpsItem

Im folgenden `get-ops-item` Beispiel werden Details zu dem angegebenen angezeigt OpsItem.

```

aws ssm get-ops-item \
  --ops-item-id oi-0b725EXAMPLE

```

Ausgabe:

```

{
  "OpsItem": {
    "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
    "CreatedTime": "2019-12-04T15:52:16.793000-08:00",
    "Description": "CloudWatch Event Rule SSMOpsItems-EC2-instance-terminated was triggered. Your EC2 instance has terminated. See below for more details.",
    "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
    "LastModifiedTime": "2019-12-04T15:52:16.793000-08:00",
    "Notifications": [],
    "RelatedOpsItems": [],
  }
}

```

```

    "Status": "Open",
    "OpsItemId": "oi-0b725EXAMPLE",
    "Title": "EC2 instance terminated",
    "Source": "EC2",
    "OperationalData": {
      "/aws/automations": {
        "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
        \"automationId\": \"AWS-CreateManagedWindowsInstance\" }, { \"automationType\":
        \"AWS:SSM:Automation\", \"automationId\": \"AWS-CreateManagedLinuxInstance\" } ]",
        "Type": "SearchableString"
      },
      "/aws/dedup": {
        "Value": "{\"dedupString\": \"SSMOpsItems-EC2-instance-terminated
        \"}",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[{\"arn\": \"arn:aws:ec2:us-east-2:111222333444:instance/
        i-05adec7e97EXAMPLE\"}]",
        "Type": "SearchableString"
      },
      "event-time": {
        "Value": "2019-12-04T23:52:16Z",
        "Type": "String"
      },
      "instance-state": {
        "Value": "terminated",
        "Type": "String"
      }
    },
    "Category": "Availability",
    "Severity": "4"
  }
}

```

Weitere Informationen finden Sie unter [Arbeiten mit OpsItems](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetOpsItem](#) in der AWS CLI Befehlsreferenz.

get-ops-summary

Das folgende Codebeispiel zeigt die Verwendung von `get-ops-summary`.

AWS CLI

Um eine Zusammenfassung aller anzuzeigen OpsItems

Im folgenden `get-ops-summary` Beispiel wird eine Zusammenfassung aller Daten OpsItems in Ihrem AWS Konto angezeigt.

```
aws ssm get-ops-summary
```

Ausgabe:

```
{
  "Entities": [
    {
      "Id": "oi-4309fEXAMPLE",
      "Data": {
        "AWS:OpsItem": {
          "CaptureTime": "2020-02-26T18:58:32.918Z",
          "Content": [
            {
              "AccountId": "111222333444",
              "Category": "Availability",
              "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
              "CreatedTime": "2020-02-26T19:10:44.149Z",
              "Description": "CloudWatch Event Rule SSMOpsItems-EC2-
instance-terminated was triggered. Your EC2 instance has terminated. See below for
more details.",
              "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
              "LastModifiedTime": "2020-02-26T19:10:44.149Z",
              "Notifications": "",
              "OperationalData": "{\"/aws/automations\":
{\"type\": \"SearchableString\", \"value\": \"[ { \\\"automationType\\\": \\
\"AWS:SSM:Automation\\\", \\\"automationId\\\": \\\"AWS-CreateManagedWindowsInstance
\\\" } , { \\\"automationType\\\": \\\"AWS:SSM:Automation\\\", \\\"automationId
\\\": \\\"AWS-CreateManagedLinuxInstance\\\" } ]\"}, \"/aws/resources\":
{\"type\": \"SearchableString\", \"value\": \"[{\\\"arn\\\": \\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0acbd0800fEXAMPLE\\\"}]\"}, \"/aws/dedup\": {\"type\":
\"SearchableString\", \"value\": \"{\\\"dedupString\\\": \\\"SSMOpsItems-EC2-instance-
terminated\\\"}\"}}",
              "OpsItemId": "oi-4309fEXAMPLE",
              "RelatedItems": "",
            }
          ]
        }
      }
    }
  ]
}
```

```

        "Severity": "3",
        "Source": "EC2",
        "Status": "Open",
        "Title": "EC2 instance terminated"
    }
  ]
}
},
{
  "Id": "oi-bb2a0e6a4541",
  "Data": {
    "AWS:OpsItem": {
      "CaptureTime": "2019-11-26T19:20:06.161Z",
      "Content": [
        {
          "AccountId": "111222333444",
          "Category": "Availability",
          "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
          "CreatedTime": "2019-11-26T20:00:07.237Z",
          "Description": "CloudWatch Event Rule SSMOpsItems-SSM-
maintenance-window-execution-failed was triggered. Your SSM Maintenance Window
execution has failed. See below for more details.",
          "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
          "LastModifiedTime": "2019-11-26T20:00:07.237Z",
          "Notifications": "",
          "OperationalData": "{\"/aws/resources\":{\"type
\": \"SearchableString\", \"value\": \"[{\\\"arn\\\": \\\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-0e83ba440dEXAMPLE\\\"]}\"}, \"/aws/dedup\":
{\"type\": \"SearchableString\", \"value\": \"[{\\\"dedupString\\\": \\\"SSMOpsItems-SSM-
maintenance-window-execution-failed\\\"]}\"}",
          "OpsItemId": "oi-bb2a0EXAMPLE",
          "RelatedItems": "",
          "Severity": "3",
          "Source": "SSM",
          "Status": "Open",
          "Title": "SSM Maintenance Window execution failed"
        }
      ]
    }
  }
}
}
}
}

```



```
]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit OpsItems](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetOpsSummary](#) in der AWS CLI Befehlsreferenz.

get-parameter-history

Das folgende Codebeispiel zeigt die Verwendung `get-parameter-history`.

AWS CLI

Um einen Werteverlauf für einen Parameter abzurufen

Im folgenden `get-parameter-history` Beispiel wird der Verlauf der Änderungen für den angegebenen Parameter einschließlich seines Werts aufgeführt.

```
aws ssm get-parameter-history \
  --name "MyStringParameter"
```

Ausgabe:

```
{
  "Parameters": [
    {
      "Name": "MyStringParameter",
      "Type": "String",
      "LastModifiedDate": 1582154711.976,
      "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
      "Description": "This is the first version of my String parameter",
      "Value": "Veni",
      "Version": 1,
      "Labels": [],
      "Tier": "Standard",
      "Policies": []
    },
    {
      "Name": "MyStringParameter",
      "Type": "String",
      "LastModifiedDate": 1582156093.471,
```

```

    "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
    "Description": "This is the second version of my String parameter",
    "Value": "Vidi",
    "Version": 2,
    "Labels": [],
    "Tier": "Standard",
    "Policies": []
  },
  {
    "Name": "MyStringParameter",
    "Type": "String",
    "LastModifiedDate": 1582156117.545,
    "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
    "Description": "This is the third version of my String parameter",
    "Value": "Vici",
    "Version": 3,
    "Labels": [],
    "Tier": "Standard",
    "Policies": []
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Parameterversionen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetParameterHistory](#) in der AWS CLI Befehlsreferenz.

get-parameter

Das folgende Codebeispiel zeigt die Verwendung `get-parameter`.

AWS CLI

Beispiel 1: Um den Wert eines Parameters anzuzeigen

Das folgende `get-parameter` Beispiel listet den Wert für den angegebenen Einzelparameter auf.

```
aws ssm get-parameter \
  --name "MyStringParameter"
```

Ausgabe:

```
{
  "Parameter": {
    "Name": "MyStringParameter",
    "Type": "String",
    "Value": "Veni",
    "Version": 1,
    "LastModifiedDate": 1530018761.888,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
    "DataType": "text"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: Um den Wert eines SecureString Parameters zu entschlüsseln

Im folgenden `get-parameter` Beispiel wird der Wert des angegebenen SecureString Parameters entschlüsselt.

```
aws ssm get-parameter \
  --name "MySecureStringParameter" \
  --with-decryption
```

Ausgabe:

```
{
  "Parameter": {
    "Name": "MySecureStringParameter",
    "Type": "SecureString",
    "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
    "Version": 2,
    "LastModifiedDate": 1582155479.205,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
    "DataType": "text"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So zeigen Sie den Wert eines Parameters mithilfe von Beschriftungen an

Das folgende `get-parameter` Beispiel listet den Wert für den angegebenen Einzelparameter mit einer angegebenen Bezeichnung auf.

```
aws ssm get-parameter \  
  --name "MyParameter:Label"
```

Ausgabe:

```
{  
  "Parameter": {  
    "Name": "MyParameter",  
    "Type": "String",  
    "Value": "parameter version 2",  
    "Version": 2,  
    "Selector": ":label",  
    "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",  
    "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",  
    "DataType": "text"  
  }  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Parameterbeschriftungen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So zeigen Sie den Wert eines Parameters mithilfe von Versionen an

Das folgende `get-parameter` Beispiel listet den Wert für die angegebene Einzelparameter-Version auf.

```
aws ssm get-parameter \  
  --name "MyParameter:2"
```

Ausgabe:

```
{  
  "Parameter": {  
    "Name": "MyParameter",  
    "Type": "String",
```

```
    "Value": "parameter version 2",
    "Version": 2,
    "Selector": ":2",
    "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
    "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
    "DataType": "text"
  }
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Parameterbeschriftungen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetParameter](#) in der AWS CLI Befehlsreferenz.

get-parameters-by-path

Das folgende Codebeispiel zeigt die Verwendung von `get-parameters-by-path`.

AWS CLI

Um Parameter in einem bestimmten Pfad aufzulisten

Das folgende `get-parameters-by-path` Beispiel listet die Parameter innerhalb der angegebenen Hierarchie auf.

```
aws ssm get-parameters-by-path \
  --path "/site/newyork/department/"
```

Ausgabe:

```
{
  "Parameters": [
    {
      "Name": "/site/newyork/department/marketing",
      "Type": "String",
      "Value": "Floor 2",
      "Version": 1,
      "LastModifiedDate": 1530018761.888,
      "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/department/marketing"
    },
  ],
}
```

```

    {
      "Name": "/site/newyork/department/infotech",
      "Type": "String",
      "Value": "Floor 3",
      "Version": 1,
      "LastModifiedDate": 1530018823.429,
      "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/infotech"
    },
    ...
  ]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Parameterhierarchien](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetParametersByPath AWS CLI Befehlsreferenz](#).

get-parameters

Das folgende Codebeispiel zeigt die Verwendung `get-parameters`.

AWS CLI

Beispiel 1: Um die Werte für einen Parameter aufzulisten

Das folgende `get-parameters` Beispiel listet die Werte für die drei angegebenen Parameter auf.

```

aws ssm get-parameters \
  --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"

```

Ausgabe:

```

{
  "Parameters": [
    {
      "Name": "MyStringListParameter",
      "Type": "StringList",
      "Value": "alpha,beta,gamma",
      "Version": 1,

```

```

        "LastModifiedDate": 1582154764.222,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
        "DataType": "text"
    },
    {
        "Name": "MyStringParameter",
        "Type": "String",
        "Value": "Vici",
        "Version": 3,
        "LastModifiedDate": 1582156117.545,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
],
"InvalidParameters": [
    "MyInvalidParameterName"
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit dem Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: Um Namen und Werte mehrerer Parameter mit der Option ``--query`` aufzulisten

Das folgende `get-parameters` Beispiel listet die Namen und Werte für die angegebenen Parameter auf.

```

aws ssm get-parameters \
  --names MyStringParameter MyStringListParameter \
  --query "Parameters[*].{Name:Name, Value:Value}"

```

Ausgabe:

```

[
  {
    "Name": "MyStringListParameter",
    "Value": "alpha,beta,gamma"
  },
  {
    "Name": "MyStringParameter",
    "Value": "Vidi"
  }
]

```

```
}  
]
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So zeigen Sie den Wert eines Parameters mithilfe von Beschriftungen an

Das folgende `get-parameter` Beispiel listet den Wert für den angegebenen Einzelparameter mit einer angegebenen Bezeichnung auf.

```
aws ssm get-parameter \  
  --name "MyParameter:Label"
```

Ausgabe:

```
{  
  "Parameters": [  
    {  
      "Name": "MyLabelParameter",  
      "Type": "String",  
      "Value": "parameter by label",  
      "Version": 1,  
      "Selector": ":label",  
      "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",  
      "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",  
      "DataType": "text"  
    },  
    {  
      "Name": "MyVersionParameter",  
      "Type": "String",  
      "Value": "parameter by version",  
      "Version": 2,  
      "Selector": ":2",  
      "LastModifiedDate": "2021-03-24T16:20:28.236000-07:00",  
      "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/unlabel-param",  
      "DataType": "text"  
    }  
  ],  
  "InvalidParameters": []  
}
```


Weitere Informationen finden Sie unter [Arbeiten mit Parameterbeschriftungen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetParameters](#) in der AWS CLI Befehlsreferenz.

get-patch-baseline-for-patch-group

Das folgende Codebeispiel zeigt die Verwendung `get-patch-baseline-for-patch-group`.

AWS CLI

Um die Patch-Baseline für eine Patch-Gruppe anzuzeigen

Im folgenden `get-patch-baseline-for-patch-group` Beispiel werden Details zur Patch-Baseline für die angegebene Patchgruppe abgerufen.

```
aws ssm get-patch-baseline-for-patch-group \  
  --patch-group "DEV"
```

Ausgabe:

```
{  
  "PatchGroup": "DEV",  
  "BaselineId": "pb-0123456789abcdef0",  
  "OperatingSystem": "WINDOWS"  
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> >__ und [Hinzufügen einer Patchgruppe zu einer Patch-Baseline](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetPatchBaselineForPatchGroup](#) in AWS CLI der Befehlsreferenz.

get-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `get-patch-baseline`.

AWS CLI

Um eine Patch-Baseline anzuzeigen

Im folgenden `get-patch-baseline` Beispiel werden die Details für die angegebene Patch-Baseline abgerufen.

```
aws ssm get-patch-baseline \  
--baseline-id "pb-0123456789abcdef0"
```

Ausgabe:

```
{  
  "BaselineId": "pb-0123456789abcdef0",  
  "Name": "WindowsPatching",  
  "OperatingSystem": "WINDOWS",  
  "GlobalFilters": {  
    "PatchFilters": []  
  },  
  "ApprovalRules": {  
    "PatchRules": [  
      {  
        "PatchFilterGroup": {  
          "PatchFilters": [  
            {  
              "Key": "PRODUCT",  
              "Values": [  
                "WindowsServer2016"  
              ]  
            }  
          ]  
        },  
        "ComplianceLevel": "CRITICAL",  
        "ApproveAfterDays": 0,  
        "EnableNonSecurity": false  
      }  
    ]  
  },  
  "ApprovedPatches": [],  
  "ApprovedPatchesComplianceLevel": "UNSPECIFIED",  
  "ApprovedPatchesEnableNonSecurity": false,  
  "RejectedPatches": [],  
  "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",  
  "PatchGroups": [  
    "QA",  
    "DEV"  
  ],  
}
```

```
"CreateDate": 1550244180.465,  
"ModifiedDate": 1550244180.465,  
"Description": "Patches for Windows Servers",  
"Sources": []  
}
```

Weitere Informationen finden Sie unter [About Patch Baselines](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [GetPatchBaseline AWS CLI Befehlsreferenz](#).

get-service-setting

Das folgende Codebeispiel zeigt die Verwendung `get-service-setting`.

AWS CLI

Um die Diensteinstellung für den Parameter Store-Durchsatz abzurufen

Im folgenden Beispiel `get-service-setting` wird die aktuelle Diensteinstellung für den Parameter Store-Durchsatz in der angegebenen Region abgerufen.

```
aws ssm get-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled
```

Ausgabe:

```
{  
  "ServiceSetting": {  
    "SettingId": "/ssm/parameter-store/high-throughput-enabled",  
    "SettingValue": "false",  
    "LastModifiedDate": 1555532818.578,  
    "LastModifiedUser": "System",  
    "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled",  
    "Status": "Default"  
  }  
}
```

Weitere Informationen finden Sie unter [Erhöhen des Durchsatzes im Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [GetServiceSetting](#) in der AWS CLI Befehlsreferenz.

label-parameter-version

Das folgende Codebeispiel zeigt die Verwendung `label-parameter-version`.

AWS CLI

Beispiel 1: Um der neuesten Version eines Parameters ein Label hinzuzufügen

Im folgenden `label-parameter-version` Beispiel wird der neuesten Version des angegebenen Parameters ein Label hinzugefügt.

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady"
```

Ausgabe:

```
{  
  "InvalidLabels": [],  
  "ParameterVersion": 3  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Parameterbeschriftungen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So fügen Sie einer bestimmten Version eines Parameters eine Bezeichnung hinzu

Im folgenden `label-parameter-version` Beispiel wird der angegebenen Version eines Parameters eine Bezeichnung hinzugefügt.

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady" \  
  --parameter-version "2" --labels "DevelopmentReady"
```

Weitere Informationen finden Sie unter [Arbeiten mit Parameterbeschriftungen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [LabelParameterVersion](#) in der AWS CLI Befehlsreferenz.

list-association-versions

Das folgende Codebeispiel zeigt die Verwendung `list-association-versions`.

AWS CLI

Um alle Versionen einer Assoziation für eine bestimmte Zuordnungs-ID aufzulisten

Das folgende `list-association-versions` Beispiel listet alle Versionen der angegebenen Assoziationen auf.

```
aws ssm list-association-versions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
  "AssociationVersions": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "CreateDate": 1550505536.726,
      "Name": "AWS-UpdateSSMAgent",
      "Parameters": {
        "allowDowngrade": [
          "false"
        ],
        "version": [
          ""
        ]
      },
      "Targets": [
        {
          "Key": "InstanceIds",
          "Values": [
            "i-1234567890abcdef0"
          ]
        }
      ],
      "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
      "AssociationName": "UpdateSSMAgent"
    }
  ]
}
```

```
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Zuordnungen in Systems Manager](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAssociationVersions](#) in der AWS CLI Befehlsreferenz.

list-associations

Das folgende Codebeispiel zeigt die Verwendung `list-associations`.

AWS CLI

Beispiel 1: Um Ihre Assoziationen für eine bestimmte Instanz aufzulisten

Das folgende Beispiel für Listenzuordnungen listet alle Assoziationen mit dem `AssociationName`, `U` auf. `updateSSMAgent`

```
aws ssm list-associations /  
--association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

Ausgabe:

```
{  
  "Associations": [  
    {  
      "Name": "AWS-UpdateSSMAgent",  
      "InstanceId": "i-1234567890abcdef0",  
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",  
      "AssociationVersion": "1",  
      "Targets": [  
        {  
          "Key": "InstanceIds",  
          "Values": [  
            "i-016648b75dd622dab"  
          ]  
        }  
      ],  
      "Overview": {  
        "Status": "Pending",  
        "DetailedStatus": "Associated",  
        "AssociationStatusAggregatedCount": {
```

```

        "Pending": 1
      }
    },
    "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
    "AssociationName": "UpdateSSMAgent"
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Zuordnungen in Systems Manager](#) im Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So listen Sie Ihre Verknüpfungen für ein bestimmtes Dokument auf

Das folgende Beispiel für Listenzuordnungen listet alle Verknüpfungen für das angegebene Dokument auf.

```

aws ssm list-associations /
  --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"

```

Ausgabe:

```

{
  "Associations": [
    {
      "Name": "AWS-UpdateSSMAgent",
      "InstanceId": "i-1234567890abcdef0",
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "Targets": [
        {
          "Key": "InstanceIds",
          "Values": [
            "i-1234567890abcdef0"
          ]
        }
      ],
      "LastExecutionDate": 1550505828.548,
      "Overview": {
        "Status": "Success",
        "DetailedStatus": "Success",
        "AssociationStatusAggregatedCount": {
          "Success": 1
        }
      }
    }
  ]
}

```

```

    }
  },
  "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
  "AssociationName": "UpdateSSMAgent"
},
{
  "Name": "AWS-UpdateSSMAgent",
  "InstanceId": "i-9876543210abcdef0",
  "AssociationId": "fbc07ef7-b985-4684-b82b-0123456789ab",
  "AssociationVersion": "1",
  "Targets": [
    {
      "Key": "InstanceIds",
      "Values": [
        "i-9876543210abcdef0"
      ]
    }
  ],
  "LastExecutionDate": 1550507531.0,
  "Overview": {
    "Status": "Success",
    "AssociationStatusAggregatedCount": {
      "Success": 1
    }
  }
}
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Zuordnungen in Systems Manager](#) im Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListAssociations](#) in der AWS CLI Befehlsreferenz.

list-command-invocations

Das folgende Codebeispiel zeigt die Verwendung `list-command-invocations`.

AWS CLI

Um die Aufrufe eines bestimmten Befehls aufzulisten

Das folgende `list-command-invocations` Beispiel listet alle Aufrufe eines Befehls auf.


```
aws ssm list-command-invocations \
  --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
  --details
```

Ausgabe:

```
{
  "CommandInvocations": [
    {
      "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
      "InstanceId": "i-02573cafcfEXAMPLE",
      "InstanceName": "",
      "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
      "DocumentName": "AWS-UpdateSSMAgent",
      "DocumentVersion": "",
      "RequestedDateTime": 1582136283.089,
      "Status": "Success",
      "StatusDetails": "Success",
      "StandardOutputUrl": "",
      "StandardErrorUrl": "",
      "CommandPlugins": [
        {
          "Name": "aws:updateSsmAgent",
          "Status": "Success",
          "StatusDetails": "Success",
          "ResponseCode": 0,
          "ResponseStartDateTime": 1582136283.419,
          "ResponseFinishDateTime": 1582136283.51,
          "Output": "\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
update skipped\n",
          "StandardOutputUrl": "",
          "StandardErrorUrl": "",
          "OutputS3Region": "us-east-2",
          "OutputS3BucketName": "",
          "OutputS3KeyPrefix": ""
        }
      ],
      "ServiceRole": "",
      "NotificationConfig": {
        "NotificationArn": "",

```

```

        "NotificationEvents": [],
        "NotificationType": ""
    },
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
},
{
    "CommandId": "ef7fd8-9b57-4151-a15c-db9a12345678",
    "InstanceId": "i-0471e04240EXAMPLE",
    "InstanceName": "",
    "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
    "DocumentName": "AWS-UpdateSSMAgent",
    "DocumentVersion": "",
    "RequestedDateTime": 1582136283.02,
    "Status": "Success",
    "StatusDetails": "Success",
    "StandardOutputUrl": "",
    "StandardErrorUrl": "",
    "CommandPlugins": [
        {
            "Name": "aws:updateSsmAgent",
            "Status": "Success",
            "StatusDetails": "Success",
            "ResponseCode": 0,
            "ResponseStartDateTime": 1582136283.812,
            "ResponseFinishDateTime": 1582136295.031,
            "Output": "Updating amazon-ssm-agent from 2.3.672.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/amazon-ssm-agent-updater-
snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/amazon-ssm-agent-snap-amd64.tar.gz
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
amazon-ssm-agent/2.3.842.0/amazon-ssm-agent-snap-amd64.tar.gz\nInitiating amazon-
ssm-agent update to 2.3.842.0\namazon-ssm-agent updated successfully to 2.3.842.0",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "OutputS3Region": "us-east-2",
            "OutputS3BucketName": "",
            "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/
i-0471e04240EXAMPLE/awsupdateSsmAgent"
        }
    ]
}

```

```
    }
  ],
  "ServiceRole": "",
  "NotificationConfig": {
    "NotificationArn": "",
    "NotificationEvents": [],
    "NotificationType": ""
  },
  "CloudWatchOutputConfig": {
    "CloudWatchLogGroupName": "",
    "CloudWatchOutputEnabled": false
  }
}
]
```

Weitere Informationen finden Sie unter [Understanding Command Statuses](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [ListCommandInvocations AWS CLI](#) Befehlsreferenz.

list-commands

Das folgende Codebeispiel zeigt die Verwendung `list-commands`.

AWS CLI

Beispiel 1: Um den Status eines bestimmten Befehls abzurufen

Im folgenden `list-commands` Beispiel wird der Status des angegebenen Befehls abgerufen und angezeigt.

```
aws ssm list-commands \
  --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

Beispiel 2: Um den Status von Befehlen abzurufen, die nach einem bestimmten Datum angefordert wurden

Im folgenden `list-commands` Beispiel werden die Details von Befehlen abgerufen, die nach dem angegebenen Datum angefordert wurden.

```
aws ssm list-commands \
```

```
--filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Beispiel 3: Um alle Befehle aufzulisten, die in einem AWS Konto angefordert wurden

Das folgende `list-commands` Beispiel listet alle Befehle auf, die von Benutzern im aktuellen AWS Konto und in der Region angefordert wurden.

```
aws ssm list-commands
```

Ausgabe:

```
{
  "Commands": [
    {
      "CommandId": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE",
      "DocumentName": "AWS-UpdateSSMAgent",
      "DocumentVersion": "",
      "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
      "ExpiresAfter": "2020-02-19T11:28:02.500000-08:00",
      "Parameters": {},
      "InstanceIds": [
        "i-028ea792daEXAMPLE",
        "i-02feef8c46EXAMPLE",
        "i-038613f3f0EXAMPLE",
        "i-03a530a2d4EXAMPLE",
        "i-083b678d37EXAMPLE",
        "i-0dee81debaEXAMPLE"
      ],
      "Targets": [],
      "RequestedDateTime": "2020-02-19T10:18:02.500000-08:00",
      "Status": "Success",
      "StatusDetails": "Success",
      "OutputS3BucketName": "",
      "OutputS3KeyPrefix": "",
      "MaxConcurrency": "50",
      "MaxErrors": "100%",
      "TargetCount": 6,
      "CompletedCount": 6,
      "ErrorCount": 0,
      "DeliveryTimedOutCount": 0,
      "ServiceRole": "",
      "NotificationConfig": {
```

```
        "NotificationArn": "",
        "NotificationEvents": [],
        "NotificationType": ""
    },
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
}
{
    "CommandId": "e9ade581-c03d-476b-9b07-26667EXAMPLE",
    "DocumentName": "AWS-FindWindowsUpdates",
    "DocumentVersion": "1",
    "Comment": "",
    "ExpiresAfter": "2020-01-24T12:37:31.874000-08:00",
    "Parameters": {
        "KbArticleIds": [
            ""
        ],
        "UpdateLevel": [
            "All"
        ]
    },
    "InstanceIds": [],
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-00ec29b21eEXAMPLE",
                "i-09911ddd90EXAMPLE"
            ]
        }
    ],
    "RequestedDateTime": "2020-01-24T11:27:31.874000-08:00",
    "Status": "Success",
    "StatusDetails": "Success",
    "OutputS3BucketName": "my-us-east-2-bucket",
    "OutputS3KeyPrefix": "my-rc-output",
    "MaxConcurrency": "50",
    "MaxErrors": "0",
    "TargetCount": 2,
    "CompletedCount": 2,
    "ErrorCount": 0,
    "DeliveryTimedOutCount": 0,
```

```
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "NotificationConfig": {
        "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
        "NotificationEvents": [
            "All"
        ],
        "NotificationType": "Invocation"
    },
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
}
{
    "CommandId": "d539b6c3-70e8-4853-80e5-0ce4fEXAMPLE",
    "DocumentName": "AWS-RunPatchBaseline",
    "DocumentVersion": "1",
    "Comment": "",
    "ExpiresAfter": "2020-01-24T12:21:04.350000-08:00",
    "Parameters": {
        "InstallOverrideList": [
            ""
        ],
        "Operation": [
            "Install"
        ],
        "RebootOption": [
            "RebootIfNeeded"
        ],
        "SnapshotId": [
            ""
        ]
    },
    "InstanceIds": [],
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-00ec29b21eEXAMPLE",
                "i-09911ddd90EXAMPLE"
            ]
        }
    ]
}
```

```

    ],
    "RequestedDateTime": "2020-01-24T11:11:04.350000-08:00",
    "Status": "Success",
    "StatusDetails": "Success",
    "OutputS3BucketName": "my-us-east-2-bucket",
    "OutputS3KeyPrefix": "my-rc-output",
    "MaxConcurrency": "50",
    "MaxErrors": "0",
    "TargetCount": 2,
    "CompletedCount": 2,
    "ErrorCount": 0,
    "DeliveryTimedOutCount": 0,
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "NotificationConfig": {
        "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
        "NotificationEvents": [
            "All"
        ],
        "NotificationType": "Invocation"
    },
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
}
]
}

```

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListCommands](#) in der AWS CLI Befehlsreferenz.

list-compliance-items

Das folgende Codebeispiel zeigt die Verwendung `list-compliance-items`.

AWS CLI

Um Compliance-Artikel für eine bestimmte Instanz aufzulisten

In diesem Beispiel werden alle Konformitätselemente für die angegebene Instanz aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-  
types "ManagedInstance"
```

Ausgabe:

```
{  
  "ComplianceItems": [  
    {  
      "ComplianceType": "Association",  
      "ResourceType": "ManagedInstance",  
      "ResourceId": "i-1234567890abcdef0",  
      "Id": "8dfe3659-4309-493a-8755-0123456789ab",  
      "Title": "",  
      "Status": "COMPLIANT",  
      "Severity": "UNSPECIFIED",  
      "ExecutionSummary": {  
        "ExecutionTime": 1550408470.0  
      },  
      "Details": {  
        "DocumentName": "AWS-GatherSoftwareInventory",  
        "DocumentVersion": "1"  
      }  
    },  
    {  
      "ComplianceType": "Association",  
      "ResourceType": "ManagedInstance",  
      "ResourceId": "i-1234567890abcdef0",  
      "Id": "e4c2ed6d-516f-41aa-aa2a-0123456789ab",  
      "Title": "",  
      "Status": "COMPLIANT",  
      "Severity": "UNSPECIFIED",  
      "ExecutionSummary": {  
        "ExecutionTime": 1550508475.0  
      },  
      "Details": {  
        "DocumentName": "AWS-UpdateSSMAgent",  
        "DocumentVersion": "1"  
      }  
    }  
  ],  
}
```



```

    ...
  ],
  "NextToken": "--token string truncated--"
}

```

Um Konformitätselemente für eine bestimmte Instanz und Zuordnungs-ID aufzulisten

In diesem Beispiel werden alle Konformitätselemente für die angegebene Instanz und Zuordnungs-ID aufgeführt.

Befehl:

```

aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ComplianceType,Values=Association,Type=EQUAL" "Key=Id,Values=e4c2ed6d-516f-41aa
aa2a-0123456789ab,Type=EQUAL"

```

Um Compliance-Elemente für eine Instanz nach einem bestimmten Datum und einer bestimmten Uhrzeit aufzulisten

In diesem Beispiel werden alle Compliance-Elemente für eine Instanz nach dem angegebenen Datum und der angegebenen Uhrzeit aufgeführt.

Befehl:

```

aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"

```

- API-Einheiten finden Sie [ListComplianceItems](#) unter AWS CLI Befehlsreferenz.

list-compliance-summaries

Das folgende Codebeispiel zeigt die Verwendung `list-compliance-summaries`.

AWS CLI

Um Konformitätszusammenfassungen für alle Konformitätstypen aufzulisten

In diesem Beispiel werden Konformitätszusammenfassungen für alle Compliance-Typen in Ihrem Konto aufgeführt.

Befehl:

```
aws ssm list-compliance-summaries
```

Ausgabe:

```
{
  "ComplianceSummaryItems": [
    {
      "ComplianceType": "Association",
      "CompliantSummary": {
        "CompliantCount": 2,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 2
        }
      },
      "NonCompliantSummary": {
        "NonCompliantCount": 0,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 0
        }
      }
    },
    {
      "ComplianceType": "Patch",
      "CompliantSummary": {
        "CompliantCount": 1,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 0
        }
      },
      "NonCompliantSummary": {
        "NonCompliantCount": 0,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 0
        }
      }
    }
  ]
}
```

```

        "UnspecifiedCount": 1
      }
    },
    "NonCompliantSummary": {
      "NonCompliantCount": 1,
      "SeveritySummary": {
        "CriticalCount": 1,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
      }
    }
  },
  ...
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

Um Konformitätszusammenfassungen für einen bestimmten Konformitätstyp aufzulisten

In diesem Beispiel wird die Konformitätszusammenfassung für den Kompatibilitätstyp Patch aufgeführt.

Befehl:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- API-Einheiten finden Sie [ListComplianceSummaries](#) in der AWS CLI Befehlsreferenz.

list-document-metadata-history

Das folgende Codebeispiel zeigt die Verwendung `list-document-metadata-history`.

AWS CLI

Beispiel: Um den Genehmigungsverlauf und den Status einer Änderungsvorlage anzuzeigen

Im folgenden `list-document-metadata-history` Beispiel wird der Genehmigungsverlauf für die angegebene Change Manager-Änderungsvorlage zurückgegeben.

```
aws ssm list-document-metadata-history \
  --name MyChangeManageTemplate \
  --metadata DocumentReviews
```

Ausgabe:

```
{
  "Name": "MyChangeManagerTemplate",
  "DocumentVersion": "1",
  "Author": "arn:aws:iam::111222333444;user/JohnDoe",
  "Metadata": {
    "ReviewerResponse": [
      {
        "CreateTime": "2021-07-30T11:58:28.025000-07:00",
        "UpdateTime": "2021-07-30T12:01:19.274000-07:00",
        "ReviewStatus": "APPROVED",
        "Comment": [
          {
            "Type": "COMMENT",
            "Content": "I approve this template version"
          }
        ],
        "Reviewer": "arn:aws:iam::111222333444;user/ShirleyRodriguez"
      },
      {
        "CreateTime": "2021-07-30T11:58:28.025000-07:00",
        "UpdateTime": "2021-07-30T11:58:28.025000-07:00",
        "ReviewStatus": "PENDING"
      }
    ]
  }
}
```

Weitere Informationen finden Sie unter [Überprüfen und Genehmigen oder Ablehnen von Änderungsvorlagen](#) im AWS Systems Manager Benutzerhandbuch.

- API-Einheiten finden Sie [ListDocumentMetadataHistory](#) in der AWS CLI Befehlsreferenz.

list-document-versions

Das folgende Codebeispiel zeigt die Verwendung `list-document-versions`.

AWS CLI

Um Dokumentversionen aufzulisten

Das folgende `list-document-versions` Beispiel listet alle Versionen eines Systems Manager Manager-Dokuments auf.

```
aws ssm list-document-versions \  
  --name "Example"
```

Ausgabe:

```
{  
  "DocumentVersions": [  
    {  
      "Name": "Example",  
      "DocumentVersion": "1",  
      "CreateDate": 1583257938.266,  
      "IsDefaultVersion": true,  
      "DocumentFormat": "YAML",  
      "Status": "Active"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Senden von Befehlen, die den Dokumentversionsparameter verwenden](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListDocumentVersions](#) unter AWS CLI Befehlsreferenz.

list-documents

Das folgende Codebeispiel zeigt die Verwendung `list-documents`.

AWS CLI

Beispiel 1: Um Dokumente aufzulisten

Das folgende `list-documents` Beispiel listet Dokumente auf, die dem anfragenden Konto gehören und mit dem benutzerdefinierten Tag versehen sind.

```
aws ssm list-documents \  
  --tag-key "tag-key" \  
  --tag-value "tag-value"
```

```
--filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Ausgabe:

```
{
  "DocumentIdentifiers": [
    {
      "Name": "Example",
      "Owner": "29884EXAMPLE",
      "PlatformTypes": [
        "Windows",
        "Linux"
      ],
      "DocumentVersion": "1",
      "DocumentType": "Automation",
      "SchemaVersion": "0.3",
      "DocumentFormat": "YAML",
      "Tags": [
        {
          "Key": "DocUse",
          "Value": "Testing"
        }
      ]
    }
  ]
}
```

Weitere Informationen finden Sie unter [AWS Systems Manager Manager-Dokumente](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So listen Sie gemeinsam genutzte Dokumente auf

Das folgende `list-documents` Beispiel listet gemeinsam genutzte Dokumente auf, einschließlich privater geteilter Dokumente, die nicht Eigentum von sind AWS.

```
aws ssm list-documents \  
--filters Key=Name,Values=sharedDocNamePrefix Key=Owner,Values=Private
```

Ausgabe:

```
{
```

```
"DocumentIdentifiers": [
  {
    "Name": "Example",
    "Owner": "12345EXAMPLE",
    "PlatformTypes": [
      "Windows",
      "Linux"
    ],
    "DocumentVersion": "1",
    "DocumentType": "Command",
    "SchemaVersion": "0.3",
    "DocumentFormat": "YAML",
    "Tags": []
  }
]
```

Weitere Informationen finden Sie unter [AWS Systems Manager Manager-Dokumente](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ListDocuments](#) in der AWS CLI Befehlsreferenz.

list-inventory-entries

Das folgende Codebeispiel zeigt die Verwendung `list-inventory-entries`.

AWS CLI

Beispiel 1: Um bestimmte Inventartypen für eine Instanz anzuzeigen

Im folgenden `list-inventory-entries` Beispiel werden die Inventareinträge für den AWS Inventartyp: `Application` für eine bestimmte Instanz aufgeführt.

```
aws ssm list-inventory-entries \
  --instance-id "i-1234567890abcdef0" \
  --type-name "AWS:Application"
```

Ausgabe:

```
{
  "TypeName": "AWS:Application",
  "InstanceId": "i-1234567890abcdef0",
```

```

"SchemaVersion": "1.1",
"CaptureTime": "2019-02-15T12:17:55Z",
"Entries": [
  {
    "Architecture": "i386",
    "Name": "Amazon SSM Agent",
    "PackageId": "{88a60be2-89a1-4df8-812a-80863c2a2b68}",
    "Publisher": "Amazon Web Services",
    "Version": "2.3.274.0"
  },
  {
    "Architecture": "x86_64",
    "InstalledTime": "2018-05-03T13:42:34Z",
    "Name": "AmazonCloudWatchAgent",
    "Publisher": "",
    "Version": "1.200442.0"
  }
]
}

```

Beispiel 2: So zeigen Sie benutzerdefinierte Inventareinträge an, die einer Instanz zugewiesen sind

Das folgende `list-inventory-entries` Beispiel listet einen benutzerdefinierten Inventareintrag auf, der einer Instanz zugewiesen ist.

```

aws ssm list-inventory-entries \
  --instance-id "i-1234567890abcdef0" \
  --type-name "Custom:RackInfo"

```

Ausgabe:

```

{
  "TypeName": "Custom:RackInfo",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.0",
  "CaptureTime": "2021-05-22T10:01:01Z",
  "Entries": [
    {
      "RackLocation": "Bay B/Row C/Rack D/Shelf E"
    }
  ]
}

```



```
}

```

- API-Einheiten finden Sie [ListInventoryEntries](#) unter AWS CLI Befehlsreferenz.

list-ops-item-related-items

Das folgende Codebeispiel zeigt die Verwendung `list-ops-item-related-items`.

AWS CLI

Um die Ressourcen für verwandte Artikel eines aufzulisten OpsItem

Das folgende `list-ops-item-related-items` Beispiel listet die Ressourcen für verwandte Artikel von auf. OpsItem

```
aws ssm list-ops-item-related-items \
  --ops-item-id "oi-f99f2EXAMPLE"
```

Ausgabe:

```
{
  "Summaries": [
    {
      "OpsItemId": "oi-f99f2EXAMPLE",
      "AssociationId": "e2036148-cccb-490e-ac2a-390e5EXAMPLE",
      "ResourceType": "AWS::SSMIncidents::IncidentRecord",
      "AssociationType": "IsParentOf",
      "ResourceUri": "arn:aws:ssm-incidents::111122223333:incident-record/example-response/64bd9b45-1d0e-2622-840d-03a87a1451fa",
      "CreatedBy": {
        "Arn": "arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForIncidentManager/IncidentResponse"
      },
      "CreatedTime": "2021-08-11T18:47:14.994000+00:00",
      "LastModifiedBy": {
        "Arn": "arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForIncidentManager/IncidentResponse"
      },
      "LastModifiedTime": "2021-08-11T18:47:14.994000+00:00"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Incident Manager-Vorfällen OpsCenter im AWS Systems Manager Manager-Benutzerhandbuch](#).

- API-Einheiten finden Sie [ListOpsItemRelatedItems](#) in der AWS CLI Befehlsreferenz.

list-resource-compliance-summaries

Das folgende Codebeispiel zeigt die Verwendung `list-resource-compliance-summaries`.

AWS CLI

Um die Anzahl der Compliance-Richtlinien auf Ressourcenebene in einer Übersicht aufzulisten

In diesem Beispiel wird die Anzahl der Konformitäten auf Ressourcenebene zusammenfassend aufgeführt.

Befehl:

```
aws ssm list-resource-compliance-summaries
```

Ausgabe:

```
{
  "ResourceComplianceSummaryItems": [
    {
      "ComplianceType": "Association",
      "ResourceType": "ManagedInstance",
      "ResourceId": "i-1234567890abcdef0",
      "Status": "COMPLIANT",
      "OverallSeverity": "UNSPECIFIED",
      "ExecutionSummary": {
        "ExecutionTime": 1550509273.0
      },
      "CompliantSummary": {
        "CompliantCount": 2,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 2
        }
      }
    }
  ]
}
```

```
    },
    "NonCompliantSummary": {
      "NonCompliantCount": 0,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
      }
    }
  },
  {
    "ComplianceType": "Patch",
    "ResourceType": "ManagedInstance",
    "ResourceId": "i-9876543210abcdef0",
    "Status": "COMPLIANT",
    "OverallSeverity": "UNSPECIFIED",
    "ExecutionSummary": {
      "ExecutionTime": 1550248550.0,
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
      "ExecutionType": "Command"
    },
    "CompliantSummary": {
      "CompliantCount": 397,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 397
      }
    },
    "NonCompliantSummary": {
      "NonCompliantCount": 0,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
      }
    }
  }
}
```

```
    }
  }
},
"NextToken": "--token string truncated--"
}
```

Um Compliance-Zusammenfassungen auf Ressourcenebene für einen bestimmten Konformitätstyp aufzulisten

In diesem Beispiel werden Konformitätszusammenfassungen auf Ressourcenebene für den Kompatibilitätstyp Patch aufgeführt.

Befehl:

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- API-Einheiten finden Sie [ListResourceComplianceSummaries](#) in AWS CLI der Befehlsreferenz.

list-resource-data-sync

Das folgende Codebeispiel zeigt die Verwendung `list-resource-data-sync`.

AWS CLI

Um Ihre Konfigurationen für die Ressourcendatensynchronisierung aufzulisten

In diesem Beispiel werden Informationen zu Ihren Konfigurationen für die Synchronisierung Ihrer Ressourcendaten abgerufen.

```
aws ssm list-resource-data-sync
```

Ausgabe:

```
{
  "ResourceDataSyncItems": [
    {
      "SyncName": "MyResourceDataSync",
      "S3Destination": {
```

```
        "BucketName": "ssm-resource-data-sync",
        "SyncFormat": "JsonSerDe",
        "Region": "us-east-1"
    },
    "LastSyncTime": 1550261472.003,
    "LastSuccessfulSyncTime": 1550261472.003,
    "LastStatus": "Successful",
    "SyncCreatedTime": 1543235736.72,
    "LastSyncStatusMessage": "The sync was successfully completed"
}
]
```

- API-Einheiten finden Sie [ListResourceDataSync](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags aufzulisten, die auf eine Patch-Baseline angewendet wurden

Im folgenden `list-tags-for-resource` Beispiel werden die Tags für eine Patch-Baseline aufgeführt.

```
aws ssm list-tags-for-resource \
  --resource-type "PatchBaseline" \
  --resource-id "pb-0123456789abcdef0"
```

Ausgabe:

```
{
  "TagList": [
    {
      "Key": "Environment",
      "Value": "Production"
    },
    {
      "Key": "Region",
      "Value": "EMEA"
    }
  ]
}
```

```
]
}
```

Weitere Informationen finden Sie unter [AWS Ressourcen taggen](#) in der AWS allgemeinen Referenz.

- API-Einheiten finden Sie unter [ListTagsForResource AWS CLI Befehlsreferenz](#).

modify-document-permission

Das folgende Codebeispiel zeigt die Verwendung `modify-document-permission`.

AWS CLI

Um Dokumentberechtigungen zu ändern

Im folgenden `modify-document-permission` Beispiel wird ein Systems Manager Manager-Dokument öffentlich freigegeben.

```
aws ssm modify-document-permission \
  --name "Example" \
  --permission-type "Share" \
  --account-ids-to-add "All"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Freigeben eines Systems Manager Manager-Dokuments](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ModifyDocumentPermission](#) in der AWS CLI Befehlsreferenz.

put-compliance-items

Das folgende Codebeispiel zeigt die Verwendung `put-compliance-items`.

AWS CLI

Um einen Konformitätstyp und Konformitätsdetails für eine bestimmte Instanz zu registrieren

In diesem Beispiel wird der Konformitätstyp `Custom:AVCheck` für die angegebene verwaltete Instanz registriert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl:

```
aws ssm put-compliance-items --resource-id "i-1234567890abcdef0" --
resource-type "ManagedInstance" --compliance-type "Custom:AVCheck"
--execution-summary "ExecutionTime=2019-02-18T16:00:00Z" --
items "Id=Version2.0,Title=ScanHost,Severity=CRITICAL,Status=COMPLIANT"
```

- API-Einheiten finden Sie [PutComplianceItems](#) unter AWS CLI Befehlsreferenz.

put-inventory

Das folgende Codebeispiel zeigt die Verwendung `put-inventory`.

AWS CLI

Um einer Instanz Kundenmetadaten zuzuweisen

In diesem Beispiel werden einer Instance Informationen zum Rack-Standort zugewiesen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl (Linux):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items
' [{"TypeName": "Custom:RackInfo", "SchemaVersion": "1.0", "CaptureTime":
"2019-01-22T10:01:01Z", "Content": [{"RackLocation": "Bay B/Row C/Rack D/Shelf
E"}]} ]'
```

Befehl (Windows):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --
items "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Content=[
B/Row C/Rack D/Shelf F']"
```

- API-Einheiten finden Sie [PutInventory](#) in der AWS CLI Befehlsreferenz.

put-parameter

Das folgende Codebeispiel zeigt die Verwendung `put-parameter`.

AWS CLI

Beispiel 1: Um einen Parameterwert zu ändern

Im folgenden `put-parameter` Beispiel wird der Wert des angegebenen Parameters geändert.

```
aws ssm put-parameter \  
  --name "MyStringParameter" \  
  --type "String" \  
  --value "Vici" \  
  --overwrite
```

Ausgabe:

```
{  
  "Version": 2,  
  "Tier": "Standard"  
}
```

Weitere Informationen finden [Sie unter Einen Systems Manager Manager-Parameter erstellen \(AWS CLI\)](#), 'Parameterschichten verwalten < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' und [Arbeiten mit Parameterrichtlinien](#) im Systems Manager Manager-Benutzerhandbuch.AWS

Beispiel 2: So erstellen Sie einen erweiterten Parameter

Im folgenden `put-parameter` Beispiel wird ein erweiterter Parameter erstellt.

```
aws ssm put-parameter \  
  --name "MyAdvancedParameter" \  
  --description "This is an advanced parameter" \  
  --value "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat [truncated]" \  
  --type "String" \  
  --tier Advanced
```

Ausgabe:

```
{  
  "Version": 1,  
  "Tier": "Advanced"  
}
```


Weitere Informationen finden [Sie unter Einen Systems Manager Manager-Parameter erstellen \(AWS CLI\)](#), 'Parameterschichten verwalten < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' und [Arbeiten mit Parameterrichtlinien](#) im Systems Manager Manager-Benutzerhandbuch.AWS

Beispiel 3: So konvertieren Sie einen Standardparameter in einen erweiterten Parameter

Das folgende `put-parameter` Beispiel konvertiert einen vorhandenen Standardparameter in einen erweiterten Parameter.

```
aws ssm put-parameter \
  --name "MyConvertedParameter" \
  --value "abc123" \
  --type "String" \
  --tier Advanced \
  --overwrite
```

Ausgabe:

```
{
  "Version": 2,
  "Tier": "Advanced"
}
```

Weitere Informationen finden [Sie unter Einen Systems Manager Manager-Parameter erstellen \(AWS CLI\)](#), 'Parameterschichten verwalten < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' und [Arbeiten mit Parameterrichtlinien](#) im Systems Manager Manager-Benutzerhandbuch.AWS

Beispiel 4: So erstellen Sie einen Parameter mit angehängter Richtlinie

Im folgenden `put-parameter` Beispiel wird ein erweiterter Parameter mit einer angehängten Parameterrichtlinie erstellt.

```
aws ssm put-parameter \
  --name "/Finance/Payroll/q2accesskey" \
  --value "P@sSw)rd" \
  --type "SecureString" \
  --tier Advanced \
  --policies "[{"Type": "Expiration", "Version": "1.0", "Attributes": {"Timestamp": "2020-06-30T00:00:00.000Z"}}, {"Type": "ExpirationNotification", "Version": "1.0", "Attributes": {"Before": "5", "Unit": "Days"}}, {"Type":
```

```
\\"NoChangeNotification\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":{\\"After\\":\\"60\\",\\"Unit\\":\\"Days\\"}}]"
```

Ausgabe:

```
{
  "Version": 1,
  "Tier": "Advanced"
}
```

Weitere Informationen finden [Sie unter Einen Systems Manager Manager-Parameter erstellen \(AWS CLI\)](#), 'Parameterschichten verwalten < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>`_ und [Arbeiten mit Parameterrichtlinien](#) im Systems Manager Manager-Benutzerhandbuch.AWS

Beispiel 5: So fügen Sie einem vorhandenen Parameter eine Richtlinie hinzu

Im folgenden `put-parameter` Beispiel wird eine Richtlinie an einen vorhandenen erweiterten Parameter angehängt.

```
aws ssm put-parameter \
  --name "/Finance/Payroll/q2accesskey" \
  --value "N3wP@sSwW)rd" \
  --type "SecureString" \
  --tier Advanced \
  --policies "[{\\"Type\\":\\"Expiration\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":{\\"Timestamp\\":\\"2020-06-30T00:00:00.000Z\\"}},{\\"Type\\":\\"ExpirationNotification\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":{\\"Before\\":\\"5\\",\\"Unit\\":\\"Days\\"}},{\\"Type\\":\\"NoChangeNotification\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":{\\"After\\":\\"60\\",\\"Unit\\":\\"Days\\"}}]"
  --overwrite
```

Ausgabe:

```
{
  "Version": 2,
  "Tier": "Advanced"
}
```

Weitere Informationen finden [Sie unter Einen Systems Manager Manager-Parameter erstellen \(AWS CLI\)](#), 'Parameterschichten verwalten < [Systems Manager](https://docs.aws.amazon.com/systems-</p>
</div>
<div data-bbox=)

manager/latest/userguide/parameter-store-advanced-parameters.html>`__ und [Arbeiten mit Parameterrichtlinien](#) im Systems Manager Manager-Benutzerhandbuch.AWS

- API-Einheiten finden Sie in der Befehlsreferenz. [PutParameter](#) AWS CLI

register-default-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `register-default-patch-baseline`.

AWS CLI

Um die Standard-Patch-Baseline festzulegen

Im folgenden `register-default-patch-baseline` Beispiel wird die angegebene benutzerdefinierte Patch-Baseline als Standard-Patch-Baseline für den unterstützten Betriebssystemtyp registriert.

```
aws ssm register-default-patch-baseline \  
  --baseline-id "pb-abc123cf9bEXAMPLE"
```

Ausgabe:

```
{  
  "BaselineId": "pb-abc123cf9bEXAMPLE"  
}
```

Im folgenden `register-default-patch-baseline` Beispiel wird die von AWS für CentOS bereitgestellte Standard-Patch-Baseline als Standard-Patch-Baseline registriert.

```
aws ssm register-default-patch-baseline \  
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/  
pb-0574b43a65ea646ed"
```

Ausgabe:

```
{  
  "BaselineId": "pb-abc123cf9bEXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Über vordefinierte und benutzerdefinierte Patch-Baselines](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [RegisterDefaultPatchBaseline AWS CLI-Befehlsreferenz](#).

register-patch-baseline-for-patch-group

Das folgende Codebeispiel zeigt die Verwendung `register-patch-baseline-for-patch-group`.

AWS CLI

Um eine Patch-Baseline für eine Patchgruppe zu registrieren

Im folgenden `register-patch-baseline-for-patch-group` Beispiel wird eine Patch-Baseline für eine Patchgruppe registriert.

```
aws ssm register-patch-baseline-for-patch-group \  
  --baseline-id "pb-045f10b4f382baeda" \  
  --patch-group "Production"
```

Ausgabe:

```
{  
  "BaselineId": "pb-045f10b4f382baeda",  
  "PatchGroup": "Production"  
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> >__ und [Hinzufügen einer Patchgruppe zu einer Patch-Baseline](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterPatchBaselineForPatchGroup](#) in AWS CLI der Befehlsreferenz.

register-target-with-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `register-target-with-maintenance-window`.

AWS CLI

Beispiel 1: Um ein einzelnes Ziel mit einem Wartungsfenster zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel wird eine Instanz mit einem Wartungsfenster registriert.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \  
  --owner-information "Single instance" \  
  --resource-type "INSTANCE"
```

Ausgabe:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Beispiel 2: Um mehrere Ziele mithilfe einer Instanz für ein Wartungsfenster zu registrieren IDs

Im folgenden `register-target-with-maintenance-window` Beispiel werden zwei Instanzen mit einem Wartungsfenster registriert, indem ihre Instanz angegeben wird IDs.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862,i-0cb2b964d3e14fd9f" \  
  --owner-information "Two instances in a list" \  
  --resource-type "INSTANCE"
```

Ausgabe:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Beispiel 3: Um Ziele mithilfe von Ressourcen-Tags für ein Wartungsfenster zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel werden Instanzen mit einem Wartungsfenster registriert, indem Ressourcen-Tags angegeben werden, die auf die Instanzen angewendet wurden.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \  
  --owner-information "Two instances in a list" \  
  --resource-type "INSTANCE"
```

```
--window-id "mw-06cf17cbefcb4bf4f" \  
--targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \  
--owner-information "Production Web Servers" \  
--resource-type "INSTANCE"
```

Ausgabe:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Beispiel 4: Um Ziele mithilfe einer Gruppe von Tag-Schlüsseln zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel werden Instanzen registriert, denen unabhängig von ihren Schlüsselwerten ein oder mehrere Tag-Schlüssel zugewiesen wurden.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "INSTANCE" \  
  --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Ausgabe:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Beispiel 5: Um Ziele mit einem Ressourcengruppenamen zu registrieren

Im folgenden `register-target-with-maintenance-window` Beispiel wird eine angegebene Ressourcengruppe unabhängig vom darin enthaltenen Ressourcentyp registriert.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "RESOURCE_GROUP" \  
  --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Ausgabe:

```
{
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Weitere Informationen finden Sie unter [Registrieren einer Zielinstanz mit dem Wartungsfenster \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterTargetWithMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

register-task-with-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `register-task-with-maintenance-window`.

AWS CLI

Beispiel 1: Um eine Automatisierungsaufgabe mit einem Wartungsfenster zu registrieren

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Automatisierungsaufgabe mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
  --window-id "mw-082dcd7649EXAMPLE" \
  --targets Key=InstanceIds,Values=i-1234520122EXAMPLE \
  --task-arn AWS-RestartEC2Instance \
  --service-role-arn arn:aws:iam::111222333444:role/SSM --task-type AUTOMATION \
  --task-invocation-parameters "{\"Automation\":{\"DocumentVersion\":\"\\$LATEST\"},\
  \"Parameters\":{\"InstanceId\":[\"{{RESOURCE_ID}}\"]}}\" \
  --priority 0 \
  --max-concurrency 1 \
  --max-errors 1 \
  --name "AutomationExample" \
  --description "Restarting EC2 Instance for maintenance"
```

Ausgabe:

```
{
  "WindowTaskId": "11144444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter [Registrieren einer Aufgabe im Wartungsfenster \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So registrieren Sie eine Lambda-Aufgabe mit einem Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Lambda-Task mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
  --window-id "mw-082dcd7649dee04e4" \
  --targets Key=InstanceIds,Values=i-12344d305eEXAMPLE \
  --task-arn arn:aws:lambda:us-east-1:111222333444:function:SSMTestLAMBDA \
  --service-role-arn arn:aws:iam::111222333444:role/SSM \
  --task-type LAMBDA \
  --task-invocation-parameters '{"Lambda":{"Payload":{"InstanceId\":"
  \{{RESOURCE_ID}}\","targetType\":"\{{TARGET_TYPE}}\"},"Qualifier":"$LATEST"}}' \
  --priority 0 \
  --max-concurrency 10 \
  --max-errors 5 \
  --name "Lambda_Example" \
  --description "My Lambda Example"
```

Ausgabe:

```
{
  "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter [Registrieren einer Aufgabe im Wartungsfenster \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So registrieren Sie eine Run-Command-Task mit einem Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Run Command-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
  --window-id "mw-082dcd7649dee04e4" \
  --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \
  --service-role-arn "arn:aws:iam::111222333444:role/SSM" \
  --task-type "RUN_COMMAND" \
  --name "SSMInstallPowerShellModule" \
```



```

--task-arn "AWS-InstallPowerShellModule" \
--task-invocation-parameters '{"RunCommand":{"Comment":"","\
OutputS3BucketName":{"runcommandlogs"},"Parameters":{"commands":["Get-
Module -ListAvailable"],"executionTimeout":["3600"],"source":["https://
gallery.technet.microsoft.com/EZOut-33ae0fb7/file/110351/1/EZOut.zip"],
workingDirectory":["\\\\\\"]},"TimeoutSeconds":600}}' \
--max-concurrency 1 \
--max-errors 1 \
--priority 10

```

Ausgabe:

```

{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}

```

Weitere Informationen finden Sie unter [Registrieren einer Aufgabe im Wartungsfenster \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So registrieren Sie eine Step Functions Functions-Aufgabe mit einem Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Step Functions Functions-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instanz ausgerichtet ist.

```

aws ssm register-task-with-maintenance-window \
--window-id "mw-1234d787d6EXAMPLE" \
--targets Key=WindowTargetIds,Values=12347414-69c3-49f8-95b8-ed2dcEXAMPLE \
--task-arn arn:aws:states:us-
east-1:111222333444:stateMachine:SSMTestStateMachine \
--service-role-arn arn:aws:iam::111222333444:role/MaintenanceWindows \
--task-type STEP_FUNCTIONS \
--task-invocation-parameters '{"StepFunctions":{"Input":{"InstanceId":
"{{RESOURCE_ID}}"}}}' \
--priority 0 \
--max-concurrency 10 \
--max-errors 5 \
--name "Step_Functions_Example" \
--description "My Step Functions Example"

```

Ausgabe:

```

{

```

```

    "WindowTaskId": "44444444-5555-6666-7777-88888888"
  }

```

Weitere Informationen finden Sie unter [Registrieren einer Aufgabe im Wartungsfenster \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 5: So registrieren Sie eine Aufgabe mithilfe einer Ziel-ID für Wartungsfenster

Im folgenden `register-task-with-maintenance-window` Beispiel wird eine Aufgabe mithilfe einer Ziel-ID für das Wartungsfenster registriert. Die Ziel-ID des Wartungsfensters war in der Ausgabe des `aws ssm register-target-with-maintenance-window` Befehls enthalten. Sie können sie auch aus der Ausgabe des `aws ssm describe-maintenance-window-targets` Befehls abrufen.

```

aws ssm register-task-with-maintenance-window \
  --targets "Key=WindowTargetIds,Values=350d44e6-28cc-44e2-951f-4b2c9EXAMPLE" \
  --task-arn "AWS-RunShellScript" \
  --service-role-arn "arn:aws:iam::111222333444:role/MaintenanceWindowsRole" \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --task-type "RUN_COMMAND" \
  --task-parameters "{\"commands\":{\"Values\":[\"df\"]}}" \
  --max-concurrency 1 \
  --max-errors 1 \
  --priority 10

```

Ausgabe:

```

{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}

```

Weitere Informationen finden Sie unter [Registrieren einer Aufgabe im Wartungsfenster \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterTaskWithMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

remove-tags-from-resource

Das folgende Codebeispiel zeigt die Verwendung `remove-tags-from-resource`.

AWS CLI

Um ein Tag aus einer Patch-Baseline zu entfernen

Im folgenden `remove-tags-from-resource` Beispiel werden Tags aus einer Patch-Baseline entfernt.

```
aws ssm remove-tags-from-resource \  
  --resource-type "PatchBaseline" \  
  --resource-id "pb-0123456789abcdef0" \  
  --tag-keys "Region"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [AWS Ressourcen taggen](#) in der AWS allgemeinen Referenz.

- API-Einheiten finden Sie unter [RemoveTagsFromResource AWS CLI Befehlsreferenz](#).

reset-service-setting

Das folgende Codebeispiel zeigt die Verwendung `reset-service-setting`.

AWS CLI

Um die Serviceeinstellung für den Parameter Store-Durchsatz zurückzusetzen

Im folgenden `reset-service-setting` Beispiel wird die Diensteinstellung für den Parameterspeicher-Durchsatz in der angegebenen Region so zurückgesetzt, dass kein erhöhter Durchsatz mehr verwendet wird.

```
aws ssm reset-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled
```

Ausgabe:

```
{  
  "ServiceSetting": {  
    "SettingId": "/ssm/parameter-store/high-throughput-enabled",  
    "SettingValue": "false",
```

```

    "LastModifiedDate": 1555532818.578,
    "LastModifiedUser": "System",
    "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled",
    "Status": "Default"
  }
}

```

Weitere Informationen finden Sie unter [Erhöhen des Durchsatzes im Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ResetServiceSetting](#) in der AWS CLI Befehlsreferenz.

resume-session

Das folgende Codebeispiel zeigt die Verwendung `resume-session`.

AWS CLI

Um eine Session Manager-Sitzung fortzusetzen

In diesem `resume-session` Beispiel wird eine Session Manager-Sitzung mit einer Instanz fortgesetzt, nachdem die Verbindung getrennt wurde. Beachten Sie, dass für diesen interaktiven Befehl das Session Manager-Plug-In auf dem Client-Computer installiert sein muss, der den Anruf durchführt.

```

aws ssm resume-session \
  --session-id Mary-Major-07a16060613c408b5

```

Ausgabe:

```

{
  "SessionId": "Mary-Major-07a16060613c408b5",
  "TokenValue":
  "AAEAAVbTGsa0nyvcUoNGqifbv5r/8lgxuQljCuY8qVcv0noBAAAAAFxtd3jIXAFUUXGTJ7zF/
AWJpWdvi0lF5p3d1AgrqVIV06IEXhkHLz0/1gXKRKEME71E6TL0p1LDJAMZ
+kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wju1PLex08SH17g5R/
wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11D0IT2doz0qXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd
+ihfGa6MEJJ97Jmat/a2TspEn0jNn9Mvu5iwXIW2yCvWZrGUj+/
QI5Xr7s1XJBEEnSKR54o4fN0GV9RWl0RZsZm1mki0JJtiwwgZ",
  "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-
Major-07a16060613c408b5?role=publish_subscribe"
}

```

```
}
```

Weitere Informationen finden Sie unter [Installieren des Session Manager-Plug-ins für AWS CLI](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [ResumeSession](#) in der AWS CLI Befehlsreferenz.

send-automation-signal

Das folgende Codebeispiel zeigt die Verwendung von `send-automation-signal`.

AWS CLI

Um ein Signal an eine Automatisierungsausführung zu senden

Das folgende `send-automation-signal` Beispiel sendet ein Approve-Signal an eine Automatisierungsausführung.

```
aws ssm send-automation-signal \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \  
  --signal-type "Approve"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Ausführen eines Automatisierungsworkflows mit Genehmigern](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie unter [SendAutomationSignal AWS CLI](#) Befehlsreferenz.

send-command

Das folgende Codebeispiel zeigt die Verwendung von `send-command`.

AWS CLI

Beispiel 1: Um einen Befehl auf einer oder mehreren Remote-Instances auszuführen

Im folgenden `send-command` Beispiel wird ein `echo` Befehl auf einer Zielinstantz ausgeführt.

```
aws ssm send-command \  
  --document-name "AWS-RunShellScript" \  
  --parameters 'commands=["echo HelloWorld"]' \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0" \  
  --
```

```
--comment "echo HelloWorld"
```

Ausgabe:

```
{
  "Command": {
    "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",
    "DocumentName": "AWS-RunShellScript",
    "DocumentVersion": "",
    "Comment": "echo HelloWorld",
    "ExpiresAfter": 1550181014.717,
    "Parameters": {
      "commands": [
        "echo HelloWorld"
      ]
    },
    "InstanceIds": [
      "i-0f00f008a2dcbef2"
    ],
    "Targets": [],
    "RequestedDateTime": 1550173814.717,
    "Status": "Pending",
    "StatusDetails": "Pending",
    "OutputS3BucketName": "",
    "OutputS3KeyPrefix": "",
    "MaxConcurrency": "50",
    "MaxErrors": "0",
    "TargetCount": 1,
    "CompletedCount": 0,
    "ErrorCount": 0,
    "DeliveryTimedOutCount": 0,
    "ServiceRole": "",
    "NotificationConfig": {
      "NotificationArn": "",
      "NotificationEvents": [],
      "NotificationType": ""
    },
    "CloudWatchOutputConfig": {
      "CloudWatchLogGroupName": "",
      "CloudWatchOutputEnabled": false
    }
  }
}
```

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So rufen Sie IP-Informationen über eine Instanz ab

Im folgenden send-command Beispiel werden die IP-Informationen über eine Instanz abgerufen.

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig"
```

In Beispiel 1 finden Sie eine Beispielausgabe.

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So führen Sie einen Befehl für Instanzen mit bestimmten Tags aus

Im folgenden send-command Beispiel wird ein Befehl für Instanzen ausgeführt, die den Tag-Schlüssel "ENV" und den Wert „Dev“ haben.

```
aws ssm send-command \  
  --targets "Key=tag:ENV,Values=Dev" \  
  --document-name "AWS-RunShellScript" \  
  --parameters "commands=ifconfig"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So führen Sie einen Befehl aus, der SNS Benachrichtigungen sendet

Im folgenden send-command Beispiel wird ein Befehl ausgeführt, der SNS Benachrichtigungen für alle Benachrichtigungsereignisse und den Command Benachrichtigungstyp sendet.

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig" \  
  --notification-configuration "sns:all"
```

```
--service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \  
--notification-config "NotificationArn=arn:aws:sns:us-  
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 5: Um einen Befehl auszuführen, der an S3 ausgegeben wird und CloudWatch

Im folgenden send-command Beispiel wird ein Befehl ausgeführt, der Befehlsdetails an einen S3-Bucket und eine CloudWatch Logs-Protokollgruppe ausgibt.

```
aws ssm send-command \  
--instance-ids "i-1234567890abcdef0" \  
--document-name "AWS-RunShellScript" \  
--comment "IP config" \  
--parameters "commands=ifconfig" \  
--output-s3-bucket-name "s3-bucket-name" \  
--output-s3-key-prefix "runcommand" \  
--cloud-watch-output-  
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

In Beispiel 1 finden Sie eine Beispielausgabe.

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 6: So führen Sie Befehle auf mehreren Instanzen mit unterschiedlichen Tags aus

Im folgenden send-command Beispiel wird ein Befehl für Instanzen mit zwei verschiedenen Tag-Schlüsseln und -Werten ausgeführt.

```
aws ssm send-command \  
--document-name "AWS-RunPowerShellScript" \  
--parameters commands=["echo helloWorld"] \  
--targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 7: So zielen Sie auf mehrere Instances mit demselben Tag-Schlüssel ab

Im folgenden send-command Beispiel wird ein Befehl für Instanzen ausgeführt, die denselben Tag-Schlüssel, aber unterschiedliche Werte haben.

```
aws ssm send-command \  
  --document-name "AWS-RunPowerShellScript" \  
  --parameters commands=["echo helloWorld"] \  
  --targets Key=tag:Env,Values=Dev,Test
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Ausführen von Befehlen mit Systems Manager Run Command](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 8: So führen Sie einen Befehl aus, der ein geteiltes Dokument verwendet

Im folgenden send-command Beispiel wird ein gemeinsam verwendetes Dokument auf einer Zielinstanz ausgeführt.

```
aws ssm send-command \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter [Verwenden gemeinsam genutzter SSM Dokumente](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [SendCommand](#) in der AWS CLI Befehlsreferenz.

start-associations-once

Das folgende Codebeispiel zeigt die Verwendung start-associations-once.

AWS CLI

Um eine Assoziation sofort und nur einmal auszuführen

Im folgenden start-associations-once Beispiel wird die angegebene Assoziation sofort und nur einmal ausgeführt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm start-associations-once \  
  --document-name "AWS-RunPowerShellScript" \  
  --parameters commands=["echo helloWorld"] \  
  --targets Key=tag:Env,Values=Dev,Test
```

```
--association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Anzeigen von Zuordnungsverläufen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [StartAssociationsOnce](#) in der AWS CLI Befehlsreferenz.

start-automation-execution

Das folgende Codebeispiel zeigt die Verwendung `start-automation-execution`.

AWS CLI

Beispiel 1: Um ein Automatisierungsdokument auszuführen

Im folgenden `start-automation-execution` Beispiel wird ein Automatisierungsdokument ausgeführt.

```
aws ssm start-automation-execution \  
  --document-name "AWS-UpdateLinuxAmi" \  
  --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/  
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Ausgabe:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

Weitere Informationen finden Sie unter [Manuelles Ausführen eines Automatisierungs-Workflows](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So führen Sie ein gemeinsam genutztes Automatisierungsdokument aus

Im folgenden `start-automation-execution` Beispiel wird ein gemeinsam genutztes Automatisierungsdokument ausgeführt.

```
aws ssm start-automation-execution \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Ausgabe:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Weitere Informationen finden Sie unter [Verwenden gemeinsam genutzter SSM Dokumente](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [StartAutomationExecution](#) in der AWS CLI Befehlsreferenz.

start-change-request-execution

Das folgende Codebeispiel zeigt die Verwendung `start-change-request-execution`.

AWS CLI

Beispiel 1: Um eine Änderungsanforderung zu starten

Im folgenden `start-change-request-execution` Beispiel wird eine Änderungsanforderung mit minimalen angegebenen Optionen gestartet.

```
aws ssm start-change-request-execution \
  --change-request-name MyChangeRequest \
  --document-name AWS-HelloWorldChangeTemplate \
  --runbooks '[{"DocumentName": "AWS-HelloWorld", "Parameters":
  [{"AutomationAssumeRole": ["arn:aws:iam:us-east-2:1112223233444:role/
  MyChangeManagerAssumeRole"]}]}]' \
  --parameters
  Approver="JohnDoe", ApproverType="IamUser", ApproverSnsTopicArn="arn:aws:sns:us-
  east-2:1112223233444:MyNotificationTopic"
```

Ausgabe:

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

Beispiel 2: Um eine Änderungsanforderung mit einer externen JSON Datei zu starten

Im folgenden `start-automation-execution` Beispiel wird eine Änderungsanforderung mit mehreren Optionen gestartet, die in einer JSON Datei angegeben sind.

```
aws ssm start-change-request-execution \
  --cli-input-json file://MyChangeRequest.json
```

Inhalt von MyChangeRequest.json:

```
{
  "ChangeRequestName": "MyChangeRequest",
  "DocumentName": "AWS-HelloWorldChangeTemplate",
  "DocumentVersion": "$DEFAULT",
  "ScheduledTime": "2021-12-30T03:00:00",
  "ScheduledEndTime": "2021-12-30T03:05:00",
  "Tags": [
    {
      "Key": "Purpose",
      "Value": "Testing"
    }
  ],
  "Parameters": {
    "Approver": [
      "JohnDoe"
    ],
    "ApproverType": [
      "IamUser"
    ],
    "ApproverSnsTopicArn": [
      "arn:aws:sns:us-east-2:111222333444:MyNotificationTopic"
    ]
  },
  "Runbooks": [
    {
      "DocumentName": "AWS-HelloWorld",
      "DocumentVersion": "1",
      "MaxConcurrency": "1",
      "MaxErrors": "1",
      "Parameters": {
        "AutomationAssumeRole": [
          "arn:aws:iam::111222333444:role/MyChangeManagerAssumeRole"
        ]
      }
    }
  ],
  "ChangeDetails": "### Document Name: HelloWorldChangeTemplate\n\n## What does this document do?\n\nThis change template demonstrates the feature set available
```

```
for creating change templates for Change Manager. This template starts a Runbook
workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters
\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
  approvers.\n* Approver: (Required) The name of the approver to send this request
  to.\n* ApproverType: (Required) The type of reviewer.\n  * Allowed Values: IamUser,
  IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\nThis document has no
outputs \n"
}
```

Ausgabe:

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

Weitere Informationen finden Sie unter [Änderungsanforderungen erstellen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [StartChangeRequestExecution](#) in der AWS CLI Befehlsreferenz.

start-session

Das folgende Codebeispiel zeigt die Verwendung `start-session`.

AWS CLI

Beispiel 1: Um eine Session Manager-Sitzung zu starten

In diesem `start-session` Beispiel wird eine Verbindung mit einer Instanz für eine Session Manager-Sitzung hergestellt. Beachten Sie, dass für diesen interaktiven Befehl das Session Manager-Plug-In auf dem Client-Computer installiert sein muss, der den Anruf durchführt.

```
aws ssm start-session \
  --target "i-1234567890abcdef0"
```

Ausgabe:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

Beispiel 2: So starten Sie eine Session Manager-Sitzung mit SSH

In diesem `start-session` Beispiel wird eine Verbindung mit einer Instanz für eine Session Manager-Sitzung hergestellt mit SSH. Beachten Sie, dass für diesen interaktiven Befehl das Session Manager-Plug-In auf dem Client-Computer installiert sein muss, der den Aufruf durchführt, und dass der Befehl den Standardbenutzer auf der Instanz verwendet, z. B. `ec2-user` für EC2 Instanzen für Linux.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

Ausgabe:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

Weitere Informationen finden Sie unter [Starten einer Sitzung](#) und [Installieren des Session Manager-Plug-ins für AWS CLI im AWS Systems Manager Manager-Benutzerhandbuch](#).

- API-Einheiten finden Sie [StartSession](#) in der AWS CLI Befehlsreferenz.

stop-automation-execution

Das folgende Codebeispiel zeigt die Verwendung `stop-automation-execution`.

AWS CLI

Um eine Automatisierungsausführung zu beenden

Im folgenden `stop-automation-execution` Beispiel wird ein Automatisierungsdokument gestoppt.

```
aws ssm stop-automation-execution  
  --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Manuelles Ausführen eines Automatisierungs-Workflows](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [StopAutomationExecution](#) unter AWS CLI Befehlsreferenz.

terminate-session

Das folgende Codebeispiel zeigt die Verwendung `terminate-session`.

AWS CLI

Um eine Session Manager-Sitzung zu beenden

In diesem `terminate-session` Beispiel wird eine Sitzung, die vom Benutzer „Shirley-Rodriguez“ erstellt wurde, dauerhaft beendet und die Datenverbindung zwischen dem Session Manager-Client und dem SSM Agenten auf der Instanz geschlossen.

```
aws ssm terminate-session \  
  --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

Ausgabe:

```
{  
  "SessionId": "Shirley-Rodriguez-07a16060613c408b5"  
}
```

Weitere Informationen finden Sie unter [Sitzung beenden](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [TerminateSession](#) in der AWS CLI Befehlsreferenz.

unlabel-parameter-version

Das folgende Codebeispiel zeigt die Verwendung `unlabel-parameter-version`.

AWS CLI

Um Parameterbeschriftungen zu löschen

Im folgenden `unlabel-parameter-version` Beispiel werden die angegebenen Labels aus der angegebenen Parameterversion gelöscht.

```
aws ssm unlabel-parameter-version \  
  --name "parameterName" \  
  --parameter-version "version" \  
  --labels "label_1" "label_2" "label_3"
```

Ausgabe:

```
{  
  "RemovedLabels": [  
    "label_1",  
    "label_2",  
    "label_3"  
  ]  
}
```

```

    "label_1"
    "label_2"
    "label_3"
  ],
  "InvalidLabels": []
}

```

Weitere Informationen finden Sie unter [Löschen von Parameterbezeichnungen \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UnlabelParameterVersion](#) in der AWS CLI Befehlsreferenz.

update-association-status

Das folgende Codebeispiel zeigt die Verwendung `update-association-status`.

AWS CLI

Um den Zuordnungsstatus zu aktualisieren

Im folgenden `update-association-status` Beispiel wird der Zuordnungsstatus der Verknüpfung zwischen einer Instanz und einem Dokument aktualisiert.

```

aws ssm update-association-status \
  --name "AWS-UpdateSSMAgent" \
  --instance-id "i-1234567890abcdef0" \
  --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additional
Config-Needed"

```

Ausgabe:

```

{
  "AssociationDescription": {
    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-1234567890abcdef0",
    "AssociationVersion": "1",
    "Date": 1550507529.604,
    "LastUpdateAssociationDate": 1550507806.974,
    "Status": {
      "Date": 1424421071.0,
      "Name": "Pending",
      "Message": "temp_status_change",

```



```

    "AdditionalInfo": "Additional-Config-Needed"
  },
  "Overview": {
    "Status": "Success",
    "AssociationStatusAggregatedCount": {
      "Success": 1
    }
  },
  "DocumentVersion": "$DEFAULT",
  "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
  "Targets": [
    {
      "Key": "InstanceIds",
      "Values": [
        "i-1234567890abcdef0"
      ]
    }
  ],
  "LastExecutionDate": 1550507808.0,
  "LastSuccessfulExecutionDate": 1550507808.0
}
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Zuordnungen in Systems Manager](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAssociationStatus](#) in der AWS CLI Befehlsreferenz.

update-association

Das folgende Codebeispiel zeigt die Verwendung `update-association`.

AWS CLI

Beispiel 1: Um eine Dokumentzuordnung zu aktualisieren

Im folgenden `update-association` Beispiel wird eine Verknüpfung mit einer neuen Dokumentversion aktualisiert.

```

aws ssm update-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --document-version "$LATEST"

```

Ausgabe:

```
{
  "AssociationDescription": {
    "Name": "AWS-UpdateSSMAgent",
    "AssociationVersion": "2",
    "Date": 1550508093.293,
    "LastUpdateAssociationDate": 1550508106.596,
    "Overview": {
      "Status": "Pending",
      "DetailedStatus": "Creating"
    },
    "DocumentVersion": "$LATEST",
    "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
    "Targets": [
      {
        "Key": "tag:Name",
        "Values": [
          "Linux"
        ]
      }
    ],
    "LastExecutionDate": 1550508094.879,
    "LastSuccessfulExecutionDate": 1550508094.879
  }
}
```

Weitere Informationen finden Sie unter [Bearbeiten und Erstellen einer neuen Version einer Zuordnung](#) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So aktualisieren Sie den Zeitplanausdruck einer Assoziation

Im folgenden `update-association` Beispiel wird der Zeitplanausdruck für die angegebene Zuordnung aktualisiert.

```
aws ssm update-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

Ausgabe:

```
{
  "AssociationDescription": {
```

```

    "Name": "AWS-HelloWorld",
    "AssociationVersion": "2",
    "Date": "2021-02-08T13:54:19.203000-08:00",
    "LastUpdateAssociationDate": "2021-06-29T11:51:07.933000-07:00",
    "Overview": {
      "Status": "Pending",
      "DetailedStatus": "Creating"
    },
    "DocumentVersion": "$DEFAULT",
    "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
    "Targets": [
      {
        "Key": "aws:NoOpAutomationTag",
        "Values": [
          "AWS-NoOpAutomationTarget-Value"
        ]
      }
    ],
    "ScheduleExpression": "cron(0 0 0/4 1/1 * ? *)",
    "LastExecutionDate": "2021-06-26T19:00:48.110000-07:00",
    "ApplyOnlyAtCronInterval": false
  }
}

```

Weitere Informationen finden Sie unter [Bearbeiten und Erstellen einer neuen Version einer Zuordnung](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateAssociation](#) unter AWS CLI Befehlsreferenz.

update-document-default-version

Das folgende Codebeispiel zeigt die Verwendung `update-document-default-version`.

AWS CLI

Um die Standardversion eines Dokuments zu aktualisieren

Im folgenden `update-document-default-version` Beispiel wird die Standardversion eines Systems Manager Manager-Dokuments aktualisiert.

```

aws ssm update-document-default-version \
  --name "Example" \
  --document-version "2"

```

Ausgabe:

```
{
  "Description": {
    "Name": "Example",
    "DefaultVersion": "2"
  }
}
```

Weitere Informationen finden Sie unter [Schreiben von SSM Dokumentinhalten](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDocumentDefaultVersion](#) in der AWS CLI Befehlsreferenz.

update-document-metadata

Das folgende Codebeispiel zeigt die Verwendung `update-document-metadata`.

AWS CLI

Beispiel: Um die neueste Version einer Änderungsvorlage zu genehmigen

Im Folgenden `update-document-metadata` finden Sie eine Genehmigung für die neueste Version einer Änderungsvorlage, die zur Überprüfung eingereicht wurde.

```
aws ssm update-document-metadata \
  --name MyChangeManagerTemplate \
  --document-reviews 'Action=Approve, Comment=[{Type=Comment, Content=Approved!}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Überprüfen und Genehmigen oder Ablehnen von Änderungsvorlagen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateDocumentMetadata](#) in der AWS CLI Befehlsreferenz.

update-document

Das folgende Codebeispiel zeigt die Verwendung `update-document`.

AWS CLI

Um eine neue Version eines Dokuments zu erstellen

Das folgende `update-document` Beispiel erstellt eine neue Version eines Dokuments, wenn es auf einem Windows-Computer ausgeführt wird. Das von angegebene Dokument `--document` muss ein JSON Format haben. Beachten Sie, dass darauf verwiesen werden `file://` muss, gefolgt vom Pfad der Inhaltsdatei. Aufgrund der Tatsache, dass der `--document-version` Parameter `$` am Anfang steht, müssen Sie unter Windows den Wert in doppelte Anführungszeichen setzen. Unter Linux, macOS oder an einer PowerShell Eingabeaufforderung müssen Sie den Wert in einfache Anführungszeichen setzen.

Windows-Version:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version "$LATEST"
```

Linux/Mac-Version:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version '$LATEST'
```

Ausgabe:

```
{  
  "DocumentDescription": {  
    "Status": "Updating",  
    "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",  
    "Name": "RunShellScript",  
    "Parameters": [  
      {  
        "Type": "StringList",  
        "Name": "commands",  
        "Description": "(Required) Specify a shell script or a command to  
run."  
      }  
    ],  
    "DocumentType": "Command",  
    "PlatformTypes": [  
      "Linux"  
    ],  
  },  
}
```

```

    "DocumentVersion": "2",
    "HashType": "Sha256",
    "CreatedDate": 1487899655.152,
    "Owner": "809632081692",
    "SchemaVersion": "2.0",
    "DefaultVersion": "1",
    "LatestVersion": "2",
    "Description": "Run an updated script"
  }
}

```

- API-Einheiten finden Sie [UpdateDocument](#) in AWS CLI der Befehlsreferenz.

update-maintenance-window-target

Das folgende Codebeispiel zeigt die Verwendung `update-maintenance-window-target`.

AWS CLI

Um ein Wartungsfensterziel zu aktualisieren

Im folgenden `update-maintenance-window-target` Beispiel wird nur der Name eines Wartungsfensterziels aktualisiert.

```

aws ssm update-maintenance-window-target \
  --window-id "mw-0c5ed765acEXAMPLE" \
  --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
  --name "NewName" \
  --no-replace

```

Ausgabe:

```

{
  "Description": "",
  "OwnerInformation": "",
  "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
  "WindowId": "mw-0c5ed765acEXAMPLE",
  "Targets": [
    {
      "Values": [
        "i-1234567890EXAMPLE"
      ]
    }
  ],
}

```

```

        "Key": "InstanceIds"
      }
    ],
    "Name": "NewName"
  }

```

Weitere Informationen finden Sie unter [Aktualisieren eines Wartungsfensters \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateMaintenanceWindowTarget](#) in der AWS CLI Befehlsreferenz.

update-maintenance-window-task

Das folgende Codebeispiel zeigt die Verwendung `update-maintenance-window-task`.

AWS CLI

Um eine Wartungsfensteraufgabe zu aktualisieren

Im folgenden `update-maintenance-window-task` Beispiel wird die Servicerolle für eine Aufgabe im Wartungsfenster aktualisiert.

```

aws ssm update-maintenance-window-task \
  --window-id "mw-0c5ed765acEXAMPLE" \
  --window-task-id "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE" \
  --service-role-arn "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM"

```

Ausgabe:

```

{
  "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
  "MaxErrors": "1",
  "TaskArn": "AWS-UpdateEC2Config",
  "MaxConcurrency": "1",
  "WindowTaskId": "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE",
  "TaskParameters": {},
  "Priority": 1,
  "TaskInvocationParameters": {
    "RunCommand": {
      "TimeoutSeconds": 600,
      "Parameters": {

```

```

        "allowDowngrade": [
            "false"
        ]
    }
}
},
"WindowId": "mw-0c5ed765acEXAMPLE",
"Description": "UpdateEC2Config",
"Targets": [
    {
        "Values": [
            "57e8344e-fe64-4023-8191-6bf05EXAMPLE"
        ],
        "Key": "WindowTargetIds"
    }
],
"Name": "UpdateEC2Config"
}

```

Weitere Informationen finden Sie unter [Aktualisieren eines Wartungsfensters \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateMaintenanceWindowTask](#) in der AWS CLI Befehlsreferenz.

update-maintenance-window

Das folgende Codebeispiel zeigt die Verwendung `update-maintenance-window`.

AWS CLI

Beispiel 1: Um ein Wartungsfenster zu aktualisieren

Im folgenden `update-maintenance-window` Beispiel wird der Name eines Wartungsfensters aktualisiert.

```

aws ssm update-maintenance-window \
  --window-id "mw-1a2b3c4d5e6f7g8h9" \
  --name "My-Renamed-MW"

```

Ausgabe:

```

{
  "Cutoff": 1,

```



```
"Name": "My-Renamed-MW",
"Schedule": "cron(0 16 ? * TUE *)",
"Enabled": true,
"AllowUnassociatedTargets": true,
"WindowId": "mw-1a2b3c4d5e6f7g8h9",
"Duration": 4
}
```

Beispiel 2: Um ein Wartungsfenster zu deaktivieren

Das folgende `update-maintenance-window` Beispiel deaktiviert ein Wartungsfenster.

```
aws ssm update-maintenance-window \
  --window-id "mw-1a2b3c4d5e6f7g8h9" \
  --no-enabled
```

Beispiel 3: Um ein Wartungsfenster zu aktivieren

Das folgende `update-maintenance-window` Beispiel aktiviert ein Wartungsfenster.

```
aws ssm update-maintenance-window \
  --window-id "mw-1a2b3c4d5e6f7g8h9" \
  --enabled
```

Weitere Informationen finden Sie unter [Aktualisieren eines Wartungsfensters \(AWS CLI\)](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateMaintenanceWindow](#) in der AWS CLI Befehlsreferenz.

update-managed-instance-role

Das folgende Codebeispiel zeigt die Verwendung `update-managed-instance-role`.

AWS CLI

Um die IAM Rolle einer verwalteten Instanz zu aktualisieren

Im folgenden `update-managed-instance-role` Beispiel wird das IAM Instanzprofil einer verwalteten Instanz aktualisiert.

```
aws ssm update-managed-instance-role \
  --instance-id "mi-08ab247cdfEXAMPLE" \
```

```
--iam-role "ExampleRole"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Schritt 4: Erstellen eines IAM Instanzprofils für Systems Manager](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateManagedInstanceRole](#) in der AWS CLI Befehlsreferenz.

update-ops-item

Das folgende Codebeispiel zeigt die Verwendung `update-ops-item`.

AWS CLI

Um ein zu aktualisieren OpsItem

Im folgenden `update-ops-item` Beispiel werden die Beschreibung, Priorität und Kategorie für ein aktualisiert OpsItem. Darüber hinaus gibt der Befehl ein SNS Thema an, an das die Benachrichtigungen gesendet werden, wenn dieses Thema bearbeitet oder geändert OpsItem wird.

```
aws ssm update-ops-item \  
  --ops-item-id "oi-287b5EXAMPLE" \  
  --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \  
  --priority 2 \  
  --category "Security" \  
  --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

Ausgabe:

```
This command produces no output.
```

Weitere Informationen finden Sie unter [Arbeiten mit OpsItems](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateOpsItem](#) in der AWS CLI Befehlsreferenz.

update-patch-baseline

Das folgende Codebeispiel zeigt die Verwendung `update-patch-baseline`.

AWS CLI

Beispiel 1: Um eine Patch-Baseline zu aktualisieren

Im folgenden `update-patch-baseline` Beispiel werden der angegebenen Patch-Baseline die beiden angegebenen Patches als abgelehnt und ein Patch als genehmigt hinzugefügt.

```
aws ssm update-patch-baseline \  
  --baseline-id "pb-0123456789abcdef0" \  
  --rejected-patches "KB2032276" "MS10-048" \  
  --approved-patches "KB2124261"
```

Ausgabe:

```
{  
  "BaselineId": "pb-0123456789abcdef0",  
  "Name": "WindowsPatching",  
  "OperatingSystem": "WINDOWS",  
  "GlobalFilters": {  
    "PatchFilters": []  
  },  
  "ApprovalRules": {  
    "PatchRules": [  
      {  
        "PatchFilterGroup": {  
          "PatchFilters": [  
            {  
              "Key": "PRODUCT",  
              "Values": [  
                "WindowsServer2016"  
              ]  
            }  
          ]  
        },  
        "ComplianceLevel": "CRITICAL",  
        "ApproveAfterDays": 0,  
        "EnableNonSecurity": false  
      }  
    ]  
  },  
  "ApprovedPatches": [  
    "KB2124261"  
  ],  
}
```

```

    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [
        "KB2032276",
        "MS10-048"
    ],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "CreateDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}

```

Beispiel 2: Um eine Patch-Baseline umzubenennen

Im folgenden `update-patch-baseline` Beispiel wird die angegebene Patch-Baseline umbenannt.

```

aws ssm update-patch-baseline \
  --baseline-id "pb-0713accee01234567" \
  --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"

```

Weitere Informationen finden Sie unter Aktualisieren oder Löschen einer Patch-Baseline < <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html> > im AWS Systems Manager Benutzerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz [UpdatePatchBaseline](#).AWS CLI

update-resource-data-sync

Das folgende Codebeispiel zeigt die Verwendung `update-resource-data-sync`.

AWS CLI

Um eine Ressourcendatensynchronisierung zu aktualisieren

Das folgende `update-resource-data-sync` Beispiel aktualisiert eine `SyncFromSource` Ressourcendatensynchronisierung.

```

aws ssm update-resource-data-sync \
  --sync-name exampleSync \

```

```
--sync-type SyncFromSource \  
--sync-source '{"SourceType":"SingleAccountMultiRegions", "SourceRegions":["us-east-1", "us-west-2"]}'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Einrichten des Systems Manager Explorers für die Anzeige von Daten aus mehreren Konten und Regionen](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateResourceDataSync](#) in der AWS CLI Befehlsreferenz.

update-service-setting

Das folgende Codebeispiel zeigt die Verwendung `update-service-setting`.

AWS CLI

Um die Serviceeinstellung für den Parameter Store-Durchsatz zu aktualisieren

Im folgenden `update-service-setting` Beispiel wird die aktuelle Diensteinstellung für den Parameterspeicher-Durchsatz in der angegebenen Region aktualisiert, sodass ein erhöhter Durchsatz verwendet wird.

```
aws ssm update-service-setting \  
--setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled \  
--setting-value true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erhöhen des Durchsatzes im Parameterspeicher](#) im AWS Systems Manager Manager-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateServiceSetting](#) in der AWS CLI Befehlsreferenz.

Amazon Textract Textract-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Textract Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

analyze-document

Das folgende Codebeispiel zeigt die Verwendung `analyze-document`.

AWS CLI

Um Text in einem Dokument zu analysieren

Das folgende `analyze-document` Beispiel zeigt, wie Text in einem Dokument analysiert wird.

Linux/macOS:

```
aws textract analyze-document \
  --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
  --feature-types ['TABLES','FORMS']
```

Windows:

```
aws textract analyze-document \
  --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}\" \
  --feature-types [\"TABLES\",\"FORMS\"]\" \
  --region region-name
```

Ausgabe:

```
{
  "Blocks": [
    {
      "Geometry": {
        "BoundingBox": {
```

```
        "Width": 1.0,
        "Top": 0.0,
        "Left": 0.0,
        "Height": 1.0
    },
    "Polygon": [
        {
            "Y": 0.0,
            "X": 0.0
        },
        {
            "Y": 0.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 0.0
        }
    ]
},
"Relationships": [
    {
        "Type": "CHILD",
        "Ids": [
            "87586964-d50d-43e2-ace5-8a890657b9a0",
            "a1e72126-21d9-44f4-a8d6-5c385f9002ba",
            "e889d012-8a6b-4d2e-b7cd-7a8b327d876a"
        ]
    }
],
"BlockType": "PAGE",
"Id": "c2227f12-b25d-4e1f-baea-1ee180d926b2"
}
],
"DocumentMetadata": {
    "Pages": 1
}
}
```

Weitere Informationen finden Sie unter Analysieren von Dokumenttext mit Amazon Textract im Amazon Textract Developers Guide

- API-Einheiten finden Sie unter [AnalyzeDocument AWS CLI-Befehlsreferenz](#).

detect-document-text

Das folgende Codebeispiel zeigt die Verwendung `detect-document-text`.

AWS CLI

Um Text in einem Dokument zu erkennen

`detect-document-text` Das folgende Beispiel zeigt, wie Text in einem Dokument erkannt wird.

Linux/macOS:

```
aws textract detect-document-text \  
  --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}'
```

Windows:

```
aws textract detect-document-text \  
  --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}\" \  
  --region region-name
```

Ausgabe:

```
{  
  "Blocks": [  
    {  
      "Geometry": {  
        "BoundingBox": {  
          "Width": 1.0,  
          "Top": 0.0,  
          "Left": 0.0,  
          "Height": 1.0  
        },  
        "Polygon": [  
          {  
            "Y": 0.0,  
            "X": 0.0  
          },  
          {  
            "Y": 0.0,  
            "X": 0.0  
          }  
        ]  
      }  
    }  
  ]  
}
```



```
        {
            "Y": 0.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 0.0
        }
    ]
},
"Relationships": [
    {
        "Type": "CHILD",
        "Ids": [
            "896a9f10-9e70-4412-81ce-49ead73ed881",
            "0da18623-dc4c-463d-a3d1-9ac050e9e720",
            "167338d7-d38c-4760-91f1-79a8ec457bb2"
        ]
    }
],
"BlockType": "PAGE",
"Id": "21f0535e-60d5-4bc7-adf2-c05dd851fa25"
},
{
    "Relationships": [
        {
            "Type": "CHILD",
            "Ids": [
                "62490c26-37ea-49fa-8034-7a9ff9369c9c",
                "1e4f3f21-05bd-4da9-ba10-15d01e66604c"
            ]
        }
    ]
},
"Confidence": 89.11581420898438,
"Geometry": {
    "BoundingBox": {
        "Width": 0.33642634749412537,
        "Top": 0.17169663310050964,
        "Left": 0.13885067403316498,
        "Height": 0.49159330129623413
    }
}
```

```
    },
    "Polygon": [
      {
        "Y": 0.17169663310050964,
        "X": 0.13885067403316498
      },
      {
        "Y": 0.17169663310050964,
        "X": 0.47527703642845154
      },
      {
        "Y": 0.6632899641990662,
        "X": 0.47527703642845154
      },
      {
        "Y": 0.6632899641990662,
        "X": 0.13885067403316498
      }
    ]
  },
  "Text": "Hello,",
  "BlockType": "LINE",
  "Id": "896a9f10-9e70-4412-81ce-49ead73ed881"
},
{
  "Relationships": [
    {
      "Type": "CHILD",
      "Ids": [
        "19b28058-9516-4352-b929-64d7cef29daf"
      ]
    }
  ]
},
"Confidence": 85.5694351196289,
"Geometry": {
  "BoundingBox": {
    "Width": 0.33182239532470703,
    "Top": 0.23131252825260162,
    "Left": 0.5091826915740967,
    "Height": 0.3766750991344452
  }
},
  "Polygon": [
    {
      "Y": 0.23131252825260162,
```

```

        "X": 0.5091826915740967
      },
      {
        "Y": 0.23131252825260162,
        "X": 0.8410050868988037
      },
      {
        "Y": 0.607987642288208,
        "X": 0.8410050868988037
      },
      {
        "Y": 0.607987642288208,
        "X": 0.5091826915740967
      }
    ]
  },
  "Text": "worlc",
  "BlockType": "LINE",
  "Id": "0da18623-dc4c-463d-a3d1-9ac050e9e720"
}
],
"DocumentMetadata": {
  "Pages": 1
}
}

```

Weitere Informationen finden Sie unter Erkennen von Dokumenttext mit Amazon Textract im Amazon Textract Developers Guide

- API-Einheiten finden Sie unter [DetectDocumentText AWS CLI](#) Befehlsreferenz.

get-document-analysis

Das folgende Codebeispiel zeigt die Verwendung `get-document-analysis`.

AWS CLI

Um die Ergebnisse einer asynchronen Textanalyse eines mehrseitigen Dokuments zu erhalten

Das folgende `get-document-analysis` Beispiel zeigt, wie die Ergebnisse einer asynchronen Textanalyse eines mehrseitigen Dokuments abgerufen werden.

```
aws textract get-document-analysis \
```

```
--job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \  
--max-results 1000
```

Ausgabe:

```
{  
  "Blocks": [  
    {  
      "Geometry": {  
        "BoundingBox": {  
          "Width": 1.0,  
          "Top": 0.0,  
          "Left": 0.0,  
          "Height": 1.0  
        },  
        "Polygon": [  
          {  
            "Y": 0.0,  
            "X": 0.0  
          },  
          {  
            "Y": 0.0,  
            "X": 1.0  
          },  
          {  
            "Y": 1.0,  
            "X": 1.0  
          },  
          {  
            "Y": 1.0,  
            "X": 0.0  
          }  
        ]  
      },  
      "Relationships": [  
        {  
          "Type": "CHILD",  
          "Ids": [  
            "75966e64-81c2-4540-9649-d66ec341cd8f",  
            "bb099c24-8282-464c-a179-8a9fa0a057f0",  
            "5ebf522d-f9e4-4dc7-bfae-a288dc094595"  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```

    ],
    "BlockType": "PAGE",
    "Id": "247c28ee-b63d-4aeb-9af0-5f7ea8ba109e",
    "Page": 1
  }
],
"NextToken": "cY1W3eTFvoB0cH7YrKVudI4Gb0H8J0xAYLo8xI/JunCIPWCthaKQ+07n/
ElyutsSy0+1V0ImoTRmP1zw4P0RFtaeV9BzhnFedpx1YqwB4xaGDA==",
"DocumentMetadata": {
  "Pages": 1
},
"JobStatus": "SUCCEEDED"
}

```

Weitere Informationen finden Sie unter Erkennen und Analysieren von Text in mehrseitigen Dokumenten im Amazon Textract Developers Guide

- API-Einheiten finden Sie unter [GetDocumentAnalysis AWS CLI Befehlsreferenz](#).

get-document-text-detection

Das folgende Codebeispiel zeigt die Verwendung `get-document-text-detection`.

AWS CLI

Um die Ergebnisse der asynchronen Texterkennung in einem mehrseitigen Dokument abzurufen

Das folgende `get-document-text-detection` Beispiel zeigt, wie die Ergebnisse der asynchronen Texterkennung in einem mehrseitigen Dokument abgerufen werden.

```

aws textract get-document-text-detection \
  --job-id 57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9 \
  --max-results 1000

```

Output

```

{
  "Blocks": [
    {
      "Geometry": {
        "BoundingBox": {
          "Width": 1.0,
          "Top": 0.0,

```

```
        "Left": 0.0,
        "Height": 1.0
    },
    "Polygon": [
        {
            "Y": 0.0,
            "X": 0.0
        },
        {
            "Y": 0.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 0.0
        }
    ]
},
"Relationships": [
    {
        "Type": "CHILD",
        "Ids": [
            "1b926a34-0357-407b-ac8f-ec473160c6a9",
            "0c35dc17-3605-4c9d-af1a-d9451059df51",
            "dea3db8a-52c2-41c0-b50c-81f66f4aa758"
        ]
    }
],
"BlockType": "PAGE",
"Id": "84671a5e-8c99-43be-a9d1-6838965da33e",
"Page": 1
}
],
"NextToken": "GcqyoAJuZwuj0T35EN4LCI3EUzMtiLq3nKyFFHvU5q1SaIdEBcSty+njNgoWwuMP/
muqc96S4o5NzDqehhXvhkodMyV050JGyms51srCxibWJw==",
"DocumentMetadata": {
    "Pages": 1
},
"JobStatus": "SUCCEEDED"
```

```
}

```

Weitere Informationen finden Sie unter Erkennen und Analysieren von Text in mehrseitigen Dokumenten im Amazon Textract Developers Guide

- API-Einheiten finden Sie unter [GetDocumentTextDetection AWS CLI Befehlsreferenz](#).

start-document-analysis

Das folgende Codebeispiel zeigt die Verwendung `start-document-analysis`.

AWS CLI

Um mit der Analyse von Text in einem mehrseitigen Dokument zu beginnen

Das folgende `start-document-analysis` Beispiel zeigt, wie die asynchrone Analyse von Text in einem mehrseitigen Dokument gestartet wird.

Linux/macOS:

```
aws textract start-document-analysis \
  --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \
  --feature-types ['TABLES','FORMS'] \
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Windows:

```
aws textract start-document-analysis \
  --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\
  \"/>
```

Ausgabe:

```
{
  "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"
}
```

Weitere Informationen finden Sie unter Erkennen und Analysieren von Text in mehrseitigen Dokumenten im Amazon Textract Developers Guide

- API-Einheiten finden Sie unter [StartDocumentAnalysis AWS CLI-Befehlsreferenz](#).

start-document-text-detection

Das folgende Codebeispiel zeigt die Verwendung `start-document-text-detection`.

AWS CLI

Um mit der Erkennung von Text in einem mehrseitigen Dokument zu beginnen

Das folgende `start-document-text-detection` Beispiel zeigt, wie die asynchrone Erkennung von Text in einem mehrseitigen Dokument gestartet wird.

Linux/macOS:

```
aws textract start-document-text-detection \  
  --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \  
  --notification-channel "SNSTopicArn=arn:sns:Topic,RoleArn=roleARN"
```

Windows:

```
aws textract start-document-text-detection \  
  --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"]]\" \  
  --region region-name \  
  --notification-channel "SNSTopicArn=arn:sns:Topic,RoleArn=roleARN"
```

Ausgabe:

```
{  
  "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"  
}
```

Weitere Informationen finden Sie unter Erkennen und Analysieren von Text in mehrseitigen Dokumenten im Amazon Textract Developers Guide

- API-Einheiten finden Sie unter [StartDocumentTextDetection AWS CLI-Befehlsreferenz](#).

Amazon Transcribe Transcribe-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Transcribe Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-language-model

Das folgende Codebeispiel zeigt die Verwendung `create-language-model`.

AWS CLI

Beispiel 1: Um ein benutzerdefiniertes Sprachmodell zu erstellen, das sowohl Trainings- als auch Optimierungsdaten verwendet.

Im folgenden `create-language-model` Beispiel wird ein benutzerdefiniertes Sprachmodell erstellt. Sie können ein benutzerdefiniertes Sprachmodell verwenden, um die Transkriptionsleistung für Bereiche wie Recht, Gastgewerbe, Finanzen und Versicherungen zu verbessern. Geben Sie als Sprachcode einen gültigen Sprachcode ein. Geben Sie für ein Basismodell an `base-model-name`, das für die Samplerate des Audios, das Sie mit Ihrem benutzerdefinierten Sprachmodell transkribieren möchten, am besten geeignet ist. Geben Sie als Modellname den Namen an, den Sie das benutzerdefinierte Sprachmodell nennen möchten.

```
aws transcribe create-language-model \
  --language-code language-code \
  --base-model-name base-model-name \
  --model-name cli-clm-example \
  --input-data-config S3Uri="s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-for-
training-data",TuningDataS3Uri="s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-for-
```

```
tuning-data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-to-create-a-custom-language-model"
```

Ausgabe:

```
{
  "LanguageCode": "language-code",
  "BaseModelName": "base-model-name",
  "ModelName": "cli-clm-example",
  "InputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",
    "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",
    "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-create-a-custom-language-model"
  },
  "ModelStatus": "IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 2: Um ein benutzerdefiniertes Sprachmodell zu erstellen, das nur Trainingsdaten verwendet.

Im folgenden Beispiel für `create-language-model` wird Ihre Audiodatei transkribiert. Sie können ein benutzerdefiniertes Sprachmodell verwenden, um die Transkriptionsleistung für Bereiche wie Recht, Gastgewerbe, Finanzen und Versicherungen zu verbessern. Geben Sie als Sprachcode einen gültigen Sprachcode ein. Geben Sie für ein Basismodell an `base-model-name`, das für die Samplerate des Audios, das Sie mit Ihrem benutzerdefinierten Sprachmodell transkribieren möchten, am besten geeignet ist. Geben Sie als Modellname den Namen an, den Sie das benutzerdefinierte Sprachmodell nennen möchten.

```
aws transcribe create-language-model \
  --language-code en-US \
  --base-model-name base-model-name \
  --model-name cli-clm-example \
  --input-data-config S3Uri="s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-For-
Training-Data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
```

Ausgabe:

```
{
  "LanguageCode": "en-US",
  "BaseModelName": "base-model-name",
  "ModelName": "cli-clm-example",
  "InputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-For-Training-Data/",
    "DataAccessRoleArn": "arn:aws:iam::your-AWS-account-number:role/IAM-role-with-permissions-to-create-a-custom-language-model"
  },
  "ModelStatus": "IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [CreateLanguageModel AWS CLI Befehlsreferenz](#).

create-medical-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `create-medical-vocabulary`.

AWS CLI

Um ein benutzerdefiniertes medizinisches Vokabular zu erstellen

Im folgenden Beispiel für `create-medical-vocabulary` wird ein benutzerdefiniertes Vokabular erstellt. Um ein benutzerdefiniertes Vokabular zu erstellen, müssen Sie eine Textdatei mit allen Begriffen erstellt haben, die Sie genauer transkribieren möchten. Geben Sie für `vocabulary-file-uri` den Amazon Simple Storage Service (Amazon S3) URI dieser Textdatei an. Geben Sie für „`language-code`“ den der Sprache Ihres benutzerdefinierten Vokabulars entsprechenden Sprachcode an. Geben Sie für „`vocabulary-name`“ die gewünschte Bezeichnung für Ihr benutzerdefiniertes Vokabular an.

```
aws transcribe create-medical-vocabulary \
  --vocabulary-name cli-medical-vocab-example \
  --language-code language-code \
  --vocabulary-file-uri https://DOC-EXAMPLE-BUCKET.AWS-Region.amazonaws.com/the-text-file-for-the-medical-custom-vocabulary.txt
```

Ausgabe:

```
{
  "VocabularyName": "cli-medical-vocab-example",
  "LanguageCode": "language-code",
  "VocabularyState": "PENDING"
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateMedicalVocabulary](#) unter AWS CLI Befehlsreferenz.

create-vocabulary-filter

Das folgende Codebeispiel zeigt die Verwendung `create-vocabulary-filter`.

AWS CLI

Um einen Vokabelfilter zu erstellen

Im folgenden `create-vocabulary-filter` Beispiel wird ein Vokabelfilter erstellt, der eine Textdatei verwendet, die eine Liste von Wörtern enthält, die in einer Transkription nicht vorkommen sollen. Geben Sie als Sprachcode den Sprachcode an, der der Sprache Ihres Vokabelfilters entspricht. Geben Sie für `vocabulary-filter-file-uri` den Amazon Simple Storage Service (Amazon S3) URI der Textdatei an. Geben Sie für `vocabulary-filter-name` den Namen Ihres Vokabelfilters an.

```
aws transcribe create-vocabulary-filter \
  --language-code language-code \
  --vocabulary-filter-file-uri s3://DOC-EXAMPLE-BUCKET/vocabulary-filter.txt \
  --vocabulary-filter-name cli-vocabulary-filter-example
```

Ausgabe:

```
{
  "VocabularyFilterName": "cli-vocabulary-filter-example",
  "LanguageCode": "language-code"
}
```

Weitere Informationen finden Sie unter [Filtern unerwünschter Wörter](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [CreateVocabularyFilter](#) in der AWS CLI Befehlsreferenz.

create-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `create-vocabulary`.

AWS CLI

Erstellen eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `create-vocabulary` wird ein benutzerdefiniertes Vokabular erstellt. Um ein benutzerdefiniertes Vokabular zu erstellen, müssen Sie eine Textdatei mit allen Begriffen erstellt haben, die Sie genauer transkribieren möchten. Geben Sie für `vocabulary-file-uri` den Amazon Simple Storage Service (Amazon S3) URI dieser Textdatei an. Geben Sie für „`language-code`“ den der Sprache Ihres benutzerdefinierten Vokabulars entsprechenden Sprachcode an. Geben Sie für „`vocabulary-name`“ die gewünschte Bezeichnung für Ihr benutzerdefiniertes Vokabular an.

```
aws transcribe create-vocabulary \  
  --language-code language-code \  
  --vocabulary-name cli-vocab-example \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/the-text-file-  
for-the-custom-vocabulary.txt
```

Ausgabe:

```
{  
  "VocabularyName": "cli-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [CreateVocabulary](#) unter AWS CLI Befehlsreferenz.

delete-language-model

Das folgende Codebeispiel zeigt die Verwendung `delete-language-model`.

AWS CLI

Um ein benutzerdefiniertes Sprachmodell zu löschen

Im folgenden `delete-language-model` Beispiel wird ein benutzerdefiniertes Sprachmodell gelöscht.

```
aws transcribe delete-language-model \
  --model-name model-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DeleteLanguageModel AWS CLI Befehlsreferenz](#).

delete-medical-transcription-job

Das folgende Codebeispiel zeigt die Verwendung `delete-medical-transcription-job`.

AWS CLI

Löschen eines medizinischen Transkriptionsauftrags

Im folgenden Beispiel für `delete-medical-transcription-job` wird ein medizinischer Transkriptionsauftrag gelöscht.

```
aws transcribe delete-medical-transcription-job \
  --medical-transcription-job-name medical-transcription-job-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteMedicalTranscriptionJob](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [DeleteMedicalTranscriptionJob](#) in der AWS CLI Befehlsreferenz.

delete-medical-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `delete-medical-vocabulary`.

AWS CLI

Um ein benutzerdefiniertes medizinisches Vokabular zu löschen

Im folgenden `delete-medical-vocabulary` Beispiel wird ein benutzerdefiniertes medizinisches Vokabular gelöscht. Geben Sie für `vocabulary-name` den Namen des benutzerdefinierten medizinischen Vokabulars an.

```
aws transcribe delete-vocabulary \  
  --vocabulary-name medical-custom-vocabulary-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [DeleteMedicalVocabulary](#) Befehlsreferenz. AWS CLI

delete-transcription-job

Das folgende Codebeispiel zeigt die Verwendung `delete-transcription-job`.

AWS CLI

Löschen eines Ihrer Transkriptionsaufträge

Im folgenden Beispiel für `delete-transcription-job` wird einer Ihrer Transkriptionsaufträge gelöscht.

```
aws transcribe delete-transcription-job \  
  --transcription-job-name your-transcription-job
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteTranscriptionJob](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [DeleteTranscriptionJob](#) in der AWS CLI Befehlsreferenz.

delete-vocabulary-filter

Das folgende Codebeispiel zeigt die Verwendung `delete-vocabulary-filter`.

AWS CLI

Um einen Vokabelfilter zu löschen

Im folgenden `delete-vocabulary-filter` Beispiel wird ein Vokabelfilter gelöscht.

```
aws transcribe delete-vocabulary-filter \  
  --vocabulary-filter-name vocabulary-filter-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Filtern unerwünschter Wörter](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [DeleteVocabularyFilter](#) in der AWS CLI Befehlsreferenz.

delete-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `delete-vocabulary`.

AWS CLI

Löschen eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `delete-vocabulary` wird ein benutzerdefiniertes Vokabular gelöscht.

```
aws transcribe delete-vocabulary \  
  --vocabulary-name vocabulary-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [DeleteVocabulary](#) in der AWS CLI Befehlsreferenz.

describe-language-model

Das folgende Codebeispiel zeigt die Verwendung des `describe-language-model`.

AWS CLI

Um Informationen über ein bestimmtes benutzerdefiniertes Sprachmodell zu erhalten

Im folgenden `describe-language-model` Beispiel werden Informationen zu einem bestimmten benutzerdefinierten Sprachmodell abgerufen. Im Folgenden können `BaseModelName` Sie beispielsweise sehen, ob Ihr Modell mit einem `NarrowBand` `WideBand` OR-Modell trainiert wurde. Benutzerdefinierte Sprachmodelle mit einem `NarrowBand` Basismodell können Audiodaten mit einer Samplerate von weniger als 16 transkribieren. kHz Sprachmodelle, die ein `WideBand` Basismodell verwenden, können Audio mit einer Samplerate von mehr als 16 transkribieren. kHz Der Parameter `S3Uri` gibt das Amazon S3 S3-Präfix an, mit dem Sie auf die Trainingsdaten zugegriffen haben, um das benutzerdefinierte Sprachmodell zu erstellen.

```
aws transcribe describe-language-model \  
  --model-name cli-clm-example
```

Ausgabe:

```
{  
  "LanguageModel": {  
    "ModelName": "cli-clm-example",  
    "CreateTime": "2020-09-25T17:57:38.504000+00:00",  
    "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",  
    "LanguageCode": "language-code",  
    "BaseModelName": "base-model-name",  
    "ModelStatus": "IN_PROGRESS",  
    "UpgradeAvailability": false,  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",  
      "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",  
      "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-with-permissions-to-create-a-custom-language-model"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [DescribeLanguageModel](#) in der AWS CLI Befehlsreferenz.

get-medical-transcription-job

Das folgende Codebeispiel zeigt die Verwendung `get-medical-transcription-job`.

AWS CLI

Um Informationen zu einem bestimmten medizinischen Transkriptionsjob zu erhalten

Im folgenden `get-medical-transcription-job` Beispiel werden Informationen zu einem bestimmten medizinischen Transkriptionsauftrag abgerufen. Verwenden Sie den Parameter, um auf die Transkriptionsergebnisse zuzugreifen. `TranscriptFileUri` Wenn Sie zusätzliche Funktionen für den Transkriptionsjob aktiviert haben, können Sie diese im Objekt Einstellungen sehen. Der Parameter `Specialty` zeigt das medizinische Fachgebiet des Anbieters an. Der Parameter `Type` gibt an, ob es sich bei der Sprache im Transkriptionsjob um ein medizinisches Gespräch oder um ein medizinisches Diktat handelt.

```
aws transcribe get-medical-transcription-job \  
  --medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

Ausgabe:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-  
job",  
    "TranscriptionJobStatus": "COMPLETED",  
    "LanguageCode": "en-US",  
    "MediaSampleRateHertz": 48000,  
    "MediaFormat": "mp4",  
    "Media": {  
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-audio-file.file-extension"  
    },  
    "Transcript": {  
      "TranscriptFileUri": "https://s3.Region.amazonaws.com/Amazon-S3-Prefix/  
vocabulary-dictation-medical-transcription-job.json"  
    }  
  }  
}
```

```
    },
    "StartTime": "2020-09-21T21:17:27.045000+00:00",
    "CreationTime": "2020-09-21T21:17:27.016000+00:00",
    "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
    "Settings": {
      "ChannelIdentification": false,
      "ShowAlternatives": false,
      "VocabularyName": "cli-medical-vocab-example"
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

Weitere Informationen finden Sie unter [Batch-Transkription](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [GetMedicalTranscriptionJob](#) in AWS CLI der Befehlsreferenz.

get-medical-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `get-medical-vocabulary`.

AWS CLI

Um Informationen über ein benutzerdefiniertes medizinisches Vokabular zu erhalten

Im folgenden `get-medical-vocabulary` Beispiel werden Informationen zu einem benutzerdefinierten medizinischen Vokabular abgerufen. Sie können den `VocabularyState` Parameter verwenden, um den Verarbeitungsstatus des Vokabulars zu sehen. Wenn `jaREADY`, können Sie ihn in der `StartMedicalTranscriptionJob` Operation verwenden. :

```
aws transcribe get-medical-vocabulary \
  --vocabulary-name medical-vocab-example
```

Ausgabe:

```
{
  "VocabularyName": "medical-vocab-example",
  "LanguageCode": "en-US",
  "VocabularyState": "READY",
  "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
```

```
"DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-
medical-custom-vocabulary"
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetMedicalVocabulary](#) in der AWS CLI Befehlsreferenz.

get-transcription-job

Das folgende Codebeispiel zeigt die Verwendung `get-transcription-job`.

AWS CLI

Abrufen von Informationen zu einem bestimmten Transkriptionsauftrag

Im folgenden Beispiel für `get-transcription-job` werden Informationen zu einem bestimmten Transkriptionsauftrag abgerufen. Verwenden Sie den `TranscriptFileUri` Parameter, um auf die Transkriptionsergebnisse zuzugreifen. Verwenden Sie den `MediaFileUri` Parameter, um zu sehen, welche Audiodatei Sie mit diesem Job transkribiert haben. Sie können das Objekt „Settings“ verwenden, um die optionalen Features zu sehen, die Sie im Transkriptionsauftrag aktiviert haben.

```
aws transcribe get-transcription-job \
  --transcription-job-name your-transcription-job
```

Ausgabe:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "your-transcription-job",
    "TranscriptionJobStatus": "COMPLETED",
    "LanguageCode": "language-code",
    "MediaSampleRateHertz": 48000,
    "MediaFormat": "mp4",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"
    },
    "Transcript": {
      "TranscriptFileUri": "https://Amazon-S3-file-location-of-transcription-
output"
```

```
    },
    "StartTime": "2020-09-18T22:27:23.970000+00:00",
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",
    "CompletionTime": "2020-09-18T22:28:21.197000+00:00",
    "Settings": {
      "ChannelIdentification": false,
      "ShowAlternatives": false
    },
    "IdentifyLanguage": true,
    "IdentifiedLanguageScore": 0.8672199249267578
  }
}
```

Weitere Informationen finden Sie unter [Erste Schritte \(AWS Befehlszeilenschnittstelle\)](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie unter [GetTranscriptionJob AWS CLI Befehlsreferenz](#).

get-vocabulary-filter

Das folgende Codebeispiel zeigt die Verwendung `get-vocabulary-filter`.

AWS CLI

Um Informationen über einen Vokabelfilter zu erhalten

Im folgenden `get-vocabulary-filter` Beispiel werden Informationen über einen Vokabelfilter abgerufen. Sie können den `DownloadUri` Parameter verwenden, um die Liste der Wörter abzurufen, mit denen Sie den Vokabelfilter erstellt haben.

```
aws transcribe get-vocabulary-filter \
  --vocabulary-filter-name testFilter
```

Ausgabe:

```
{
  "VocabularyFilterName": "testFilter",
  "LanguageCode": "language-code",
  "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",
  "DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"
}
```

Weitere Informationen finden Sie unter [Unerwünschte Wörter filtern](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [GetVocabularyFilter](#) in der AWS CLI Befehlsreferenz.

get-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `get-vocabulary`.

AWS CLI

Abrufen von Informationen zu einem benutzerdefinierten Vokabular

Im folgenden Beispiel für `get-vocabulary` werden Informationen zu einem zuvor erstellten benutzerdefinierten Vokabular abgerufen.

```
aws transcribe get-vocabulary \  
  --vocabulary-name cli-vocab-1
```

Ausgabe:

```
{  
  "VocabularyName": "cli-vocab-1",  
  "LanguageCode": "language-code",  
  "VocabularyState": "READY",  
  "LastModifiedTime": "2020-09-19T23:22:32.836000+00:00",  
  "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-  
  custom-vocabulary"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetVocabulary](#) in der AWS CLI Befehlsreferenz.

list-language-models

Das folgende Codebeispiel zeigt die Verwendung `list-language-models`.

AWS CLI

Um Ihre benutzerdefinierten Sprachmodelle aufzulisten

Im folgenden `list-language-models` Beispiel werden die benutzerdefinierten Sprachmodelle aufgeführt, die Ihrem AWS Konto und Ihrer Region zugeordnet sind. Sie können die `TuningDataS3Uri` Parameter `S3Uri` und verwenden, um die Amazon S3 `S3`-Präfixe zu finden, die Sie als Ihre Trainingsdaten oder Ihre Tuning-Daten verwendet haben. Das `BaseModelName` sagt Ihnen, ob Sie ein oder `WideBand` -Modell verwendet haben `NarrowBand`, um ein benutzerdefiniertes Sprachmodell zu erstellen. Sie können Audio mit einer Samplerate von weniger als 16 kHz mit einem benutzerdefinierten Sprachmodell transkribieren, das ein `NarrowBand` Basismodell verwendet. Sie können Audio 16 kHz oder höher mit einem benutzerdefinierten Sprachmodell transkribieren, das ein Basismodell verwendet. `WideBand` Der `ModelStatus` Parameter zeigt an, ob Sie das benutzerdefinierte Sprachmodell in einem Transkriptionsauftrag verwenden können. Wenn der Wert lautet `COMPLETED`, können Sie ihn in einem Transkriptionsauftrag verwenden.

```
aws transcribe list-language-models
```

Ausgabe:

```
{
  "Models": [
    {
      "ModelName": "cli-clm-2",
      "CreateTime": "2020-09-25T17:57:38.504000+00:00",
      "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",
      "LanguageCode": "language-code",
      "BaseModelName": "WideBand",
      "ModelStatus": "IN_PROGRESS",
      "UpgradeAvailability": false,
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-training-data/",
        "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-tuning-data/",
        "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-used-to-create-the-custom-language-model"
      }
    },
    {
      "ModelName": "cli-clm-1",
      "CreateTime": "2020-09-25T17:16:01.835000+00:00",
      "LastModifiedTime": "2020-09-25T17:16:15.555000+00:00",
      "LanguageCode": "language-code",
      "BaseModelName": "WideBand",
      "ModelStatus": "IN_PROGRESS",
```

```

    "UpgradeAvailability": false,
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-training-data/",
      "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-used-to-create-the-custom-language-model"
    }
  },
  {
    "ModelName": "clm-console-1",
    "CreateTime": "2020-09-24T19:26:28.076000+00:00",
    "LastModifiedTime": "2020-09-25T04:25:22.271000+00:00",
    "LanguageCode": "language-code",
    "BaseModelName": "NarrowBand",
    "ModelStatus": "COMPLETED",
    "UpgradeAvailability": false,
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-training-data/",
      "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-used-to-create-the-custom-language-model"
    }
  }
]
}

```

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListLanguageModels](#) in der AWS CLI Befehlsreferenz.

list-medical-transcription-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-medical-transcription-jobs`.

AWS CLI

Auflisten von medizinischen Transkriptionsaufträgen

Im folgenden `list-medical-transcription-jobs` Beispiel werden die medizinischen Transkriptionsaufträge aufgeführt, die Ihrem AWS Konto und Ihrer Region zugeordnet sind. Um weitere Informationen zu einem bestimmten Transkriptionsauftrag zu erhalten, kopieren Sie den Wert eines `MedicalTranscriptionJobName` Parameters in die Transkriptionsausgabe und

geben Sie diesen Wert für die `MedicalTranscriptionJobName` Option des Befehls an. `get-medical-transcription-job` Um mehr Ihrer Transkriptionsaufträge zu sehen, kopieren Sie den Wert des `NextToken` Parameters, führen Sie den `list-medical-transcription-jobs` Befehl erneut aus und geben Sie diesen Wert in der Option an. `--next-token`

```
aws transcribe list-medical-transcription-jobs
```

Ausgabe:

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb0l
+LaDtZPC4u6ttoHLRL1EfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPBQ0HPjtwmFI+IWX15nSUjWuN3TUy1HgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSenzq49Q9xHgHdT2yBADFmvFK7myZBsJ75+2vQZ0SVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfyq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPW1D7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
Wff934aKqfD6UTggm/zV3GA0WiBpfvAZRvEb924i6yGHyMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJl1th6aM5vUPXg4BpyUhtbRhwd/KxCvf9K0tLJGyL1A==",
  "MedicalTranscriptionJobSummaries": [
    {
      "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:17:27.016000+00:00",
      "StartTime": "2020-09-21T21:17:27.045000+00:00",
      "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
      "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
```

```
    "CreationTime": "2020-09-21T21:01:14.569000+00:00",
    "StartTime": "2020-09-21T21:01:14.592000+00:00",
    "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  },
  {
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",
    "StartTime": "2020-09-21T19:09:18.199000+00:00",
    "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  },
  {
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  },
  {
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
```

```
    }  
  ]  
}
```

Weitere Informationen finden Sie unter <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html> im Amazon Transcribe Developer Guide.

- API-Einheiten [ListMedicalTranscriptionJobs AWS CLI](#) finden Sie in der Befehlsreferenz.

list-medical-vocabularies

Das folgende Codebeispiel zeigt die Verwendung `list-medical-vocabularies`.

AWS CLI

Um Ihre benutzerdefinierten medizinischen Vokabulare aufzulisten

Im folgenden `list-medical-vocabularies` Beispiel werden die benutzerdefinierten medizinischen Vokabeln aufgeführt, die mit Ihrem AWS Konto und Ihrer Region verknüpft sind. Um weitere Informationen zu einem bestimmten Transkriptionsauftrag zu erhalten, kopieren Sie den Wert eines `MedicalTranscriptionJobName` Parameters in die Transkriptionsausgabe und geben Sie diesen Wert für die `MedicalTranscriptionJobName` Option des Befehls an. `get-medical-transcription-job` Um mehr Ihrer Transkriptionsaufträge zu sehen, kopieren Sie den Wert des `NextToken` Parameters, führen Sie den `list-medical-transcription-jobs` Befehl erneut aus und geben Sie diesen Wert in der Option an. `--next-token`

```
aws transcribe list-medical-vocabularies
```

Ausgabe:

```
{  
  "Vocabularies": [  
    {  
      "VocabularyName": "cli-medical-vocab-2",  
      "LanguageCode": "en-US",  
      "LastModifiedTime": "2020-09-21T21:44:59.521000+00:00",  
      "VocabularyState": "READY"  
    },  
    {  
      "VocabularyName": "cli-medical-vocab-1",  
      "LanguageCode": "en-US",  
      "LastModifiedTime": "2020-09-21T21:44:59.521000+00:00",  
      "VocabularyState": "READY"  
    }  
  ]  
}
```

```
        "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
        "VocabularyState": "READY"
    }
]
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListMedicalVocabularies](#) in der AWS CLI Befehlsreferenz.

list-transcription-jobs

Das folgende Codebeispiel zeigt die Verwendung `list-transcription-jobs`.

AWS CLI

Auflisten Ihrer Transkriptionsaufträge

Das folgende `list-transcription-jobs` Beispiel listet die Transkriptionsaufträge auf, die Ihrem AWS Konto und Ihrer Region zugeordnet sind.

```
aws transcribe list-transcription-jobs
```

Ausgabe:

```
{
  "NextToken": "NextToken",
  "TranscriptionJobSummaries": [
    {
      "TranscriptionJobName": "speak-id-job-1",
      "CreationTime": "2020-08-17T21:06:15.391000+00:00",
      "StartTime": "2020-08-17T21:06:15.416000+00:00",
      "CompletionTime": "2020-08-17T21:07:05.098000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "job-1",
      "CreationTime": "2020-08-17T20:50:24.207000+00:00",
      "StartTime": "2020-08-17T20:50:24.230000+00:00",

```

```

    "CompletionTime": "2020-08-17T20:52:18.737000+00:00",
    "LanguageCode": "language-code",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "SERVICE_BUCKET"
  },
  {
    "TranscriptionJobName": "sdk-test-job-4",
    "CreationTime": "2020-08-17T20:32:27.917000+00:00",
    "StartTime": "2020-08-17T20:32:27.956000+00:00",
    "CompletionTime": "2020-08-17T20:33:15.126000+00:00",
    "LanguageCode": "language-code",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "SERVICE_BUCKET"
  },
  {
    "TranscriptionJobName": "Diarization-speak-id",
    "CreationTime": "2020-08-10T22:10:09.066000+00:00",
    "StartTime": "2020-08-10T22:10:09.116000+00:00",
    "CompletionTime": "2020-08-10T22:26:48.172000+00:00",
    "LanguageCode": "language-code",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "SERVICE_BUCKET"
  },
  {
    "TranscriptionJobName": "your-transcription-job-name",
    "CreationTime": "2020-07-29T17:45:09.791000+00:00",
    "StartTime": "2020-07-29T17:45:09.826000+00:00",
    "CompletionTime": "2020-07-29T17:46:20.831000+00:00",
    "LanguageCode": "language-code",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "SERVICE_BUCKET"
  }
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte \(AWS Befehlszeilenschnittstelle\)](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie unter [ListTranscriptionJobs AWS CLI Befehlsreferenz](#).

list-vocabularies

Das folgende Codebeispiel zeigt die Verwendung `list-vocabularies`.

AWS CLI

Auflisten Ihrer benutzerdefinierten Vokabulare

Das folgende `list-vocabularies` Beispiel listet die benutzerdefinierten Vokabulare auf, die mit Ihrem AWS Konto und Ihrer Region verknüpft sind.

```
aws transcribe list-vocabularies
```

Ausgabe:

```
{
  "NextToken": "NextToken",
  "Vocabularies": [
    {
      "VocabularyName": "ards-test-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-27T22:00:27.330000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "sample-test",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T23:04:11.044000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-test-3-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T22:12:22.277000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-test-2",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T21:53:50.455000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-1-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T21:39:33.356000+00:00",
      "VocabularyState": "READY"
    }
  ]
}
```

```
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [ListVocabularies](#) in der AWS CLI Befehlsreferenz.

list-vocabulary-filters

Das folgende Codebeispiel zeigt die Verwendung `list-vocabulary-filters`.

AWS CLI

Um Ihre Vokabelfilter aufzulisten

Das folgende `list-vocabulary-filters` Beispiel listet die Vokabelfilter auf, die mit Ihrem AWS Konto und Ihrer Region verknüpft sind.

```
aws transcribe list-vocabulary-filters
```

Ausgabe:

```
{  
  "NextToken": "NextToken": [  
    {  
      "VocabularyFilterName": "testFilter",  
      "LanguageCode": "language-code",  
      "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00"  
    },  
    {  
      "VocabularyFilterName": "testFilter2",  
      "LanguageCode": "language-code",  
      "LastModifiedTime": "2020-05-21T23:29:35.174000+00:00"  
    },  
    {  
      "VocabularyFilterName": "filter2",  
      "LanguageCode": "language-code",  
      "LastModifiedTime": "2020-05-08T20:18:26.426000+00:00"  
    },  
    {
```

```

    "VocabularyFilterName": "filter-review",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-06-03T18:52:30.448000+00:00"
  },
  {
    "VocabularyFilterName": "crlf-filt",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-05-22T19:42:42.737000+00:00"
  }
]
}

```

Weitere Informationen finden Sie unter [Filtern unerwünschter Wörter](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [ListVocabularyFilters](#) in der AWS CLI Befehlsreferenz.

start-medical-transcription-job

Das folgende Codebeispiel zeigt die Verwendung `start-medical-transcription-job`.

AWS CLI

Beispiel 1: Transkribieren eines als Audiodatei gespeicherten medizinischen Diktats

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfile.json

```

Inhalt von `myfile.json`:

```

{
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}

```



```
}
}
```

Ausgabe:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T00:35:22.256000+00:00",
    "CreationTime": "2020-09-20T00:35:22.218000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

Weitere Informationen finden Sie unter [Übersicht über die Batch-Transkription](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 2: Transkribieren eines als Audiodatei gespeicherten Dialogs zwischen Arzt und Patient

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei mit einem Dialog zwischen Arzt und Patient transkribiert. Sie geben den Speicherort der Transkriptionsausgabe im `OutputBucketName` Parameter an.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysecondfile.json
```

Inhalt von `mysecondfile.json`:

```
{
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
```

```

    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}

```

Ausgabe:

```

{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-conversation-medical-transcription-
job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:19:49.965000+00:00",
    "CreationTime": "2020-09-20T23:19:49.941000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}

```

Weitere Informationen finden Sie unter [Übersicht über die Batch-Transkription](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 3: Transkribieren einer Mehrkanal-Audiodatei eines Dialogs zwischen Arzt und Patient

Im folgenden Beispiel für `start-medical-transcription-job` werden die Audiodaten aus jedem Kanal in der Audiodatei transkribiert und die einzelnen Transkriptionen von jedem Kanal zu einer einzigen Transkriptionsausgabe zusammengeführt. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://mythirdfile.json

```

Inhalt von `mythirdfile.json`:

```

{
  "MedicalTranscriptionJobName": "multichannel-conversation-medical-transcription-
job",
  "LanguageCode": "language-code",

```

```
"Specialty": "PRIMARYCARE",
>Type": "CONVERSATION",
>OutputBucketName": "DOC-EXAMPLE-BUCKET",
>  Media": {
>    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
>  },
>  Settings": {
>    "ChannelIdentification": true
>  }
}
```

Ausgabe:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "Settings": {
      "ChannelIdentification": true
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

Weitere Informationen finden Sie unter [Kanalidentifizierung](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 4: Transkribieren einer Audiodatei eines Dialogs zwischen Arzt und Patient und Identifizieren der Sprecher in der Transkriptionsausgabe

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert und die Sprache der einzelnen Sprecher wird in der Transkriptionsausgabe gekennzeichnet. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-medical-transcription-job \  
--cli-input-json file://myfourthfile.json
```

Inhalt von `myfourthfile.json`:

```
{  
  "MedicalTranscriptionJobName": "speaker-id-conversation-medical-transcription-  
job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  },  
  "Settings": {  
    "ShowSpeakerLabels": true,  
    "MaxSpeakerLabels": 2  
  }  
}
```

Ausgabe:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-21T18:43:37.265000+00:00",  
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",  
    "Settings": {  
      "ShowSpeakerLabels": true,  
      "MaxSpeakerLabels": 2  
    },  
    "Specialty": "PRIMARYCARE",  
    "Type": "CONVERSATION"  
  }  
}
```

Weitere Informationen finden Sie unter [Identifizieren von Sprechern](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 5: Transkribieren eines als Audiodatei gespeicherten medizinischen Gesprächs mit bis zu zwei Transkriptionsalternativen

Im folgenden Beispiel für `start-medical-transcription-job` werden bis zu zwei alternative Transkriptionen aus einer einzigen Audiodatei erstellt. Jeder Transkription ist ein gewisses Konfidenzniveau zugeordnet. Standardmäßig gibt Amazon Transcribe die Transkription mit dem höchsten Konfidenzniveau zurück. Sie können angeben, dass Amazon Transcribe zusätzliche Transkriptionen mit niedrigerem Konfidenzniveau zurückgeben soll. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://myfifthfile.json
```

Inhalt von `myfifthfile.json`:

```
{  
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-  
job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  },  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 2  
  }  
}
```

Ausgabe:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",
```

```

    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T19:09:18.199000+00:00",
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}

```

Weitere Informationen finden Sie unter [Alternative Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 6: Transkribieren einer Audiodatei eines medizinischen Diktats mit bis zu zwei alternativen Transkriptionen

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein Vokabularfilter verwendet. Sie geben den Ort der Transkriptionsausgabe im Parameter an `OutputBucketName`.

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysixthfile.json

```

Inhalt von `mysixthfile.json`:

```

{
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "ShowAlternatives": true,
    "MaxAlternatives": 2
  }
}

```

```
}
}
```

Ausgabe:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T21:01:14.592000+00:00",
    "CreationTime": "2020-09-21T21:01:14.569000+00:00",
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

Weitere Informationen finden Sie unter [Alternative Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 7: Transkribieren einer Audiodatei eines medizinischen Diktats mit höherer Genauigkeit durch Verwendung eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert und zur Verbesserung der Transkriptionsgenauigkeit wird ein zuvor von Ihnen erstelltes benutzerdefiniertes medizinisches Vokabular verwendet. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json
```

Inhalt von `mysixthfile.json`:

```
{
```

```

"MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-job",
"LanguageCode": "language-code",
"Specialty": "PRIMARYCARE",
"Type": "DICTATION",
"OutputBucketName": "DOC-EXAMPLE-BUCKET",
"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
},
"Settings": {
  "VocabularyName": "cli-medical-vocab-1"
}
}

```

Ausgabe:

```

{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T21:17:27.045000+00:00",
    "CreationTime": "2020-09-21T21:17:27.016000+00:00",
    "Settings": {
      "VocabularyName": "cli-medical-vocab-1"
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}

```

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie unter [StartMedicalTranscriptionJob AWS CLI Befehlsreferenz](#).

start-transcription-job

Das folgende Codebeispiel zeigt die Verwendung `start-transcription-job`.

AWS CLI

Beispiel 1: Transkribieren einer Audiodatei

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei transkribiert.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myfile.json
```

Inhalt von `myfile.json`:

```
{  
  "TranscriptionJobName": "cli-simple-transcription-job",  
  "LanguageCode": "the-language-of-your-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-  
name.file-extension"  
  }  
}
```

Weitere Informationen finden Sie unter [Erste Schritte \(AWS Befehlszeilenschnittstelle\)](#) im Amazon Transcribe Developer Guide.

Beispiel 2: Transkribieren einer Mehrkanal-Audiodatei

Im folgenden Beispiel für `start-transcription-job` wird Ihre Mehrkanal-Audiodatei transkribiert.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysecondfile.json
```

Inhalt von `mysecondfile.json`:

```
{  
  "TranscriptionJobName": "cli-channelid-job",  
  "LanguageCode": "the-language-of-your-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-  
name.file-extension"  
  },  
  "Settings": {  
    "ChannelIdentification": true  
  }  
}
```

```
}
}
```

Ausgabe:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-channelid-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-17T16:07:56.817000+00:00",
    "CreationTime": "2020-09-17T16:07:56.784000+00:00",
    "Settings": {
      "ChannelIdentification": true
    }
  }
}
```

Weitere Informationen finden Sie unter [Transkribieren von Mehrkanal-Audio](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 3: Transkribieren einer Audiodatei und Identifizieren der verschiedenen Sprecher

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei transkribiert und die Sprecher werden in der Transkriptionsausgabe identifiziert.

```
aws transcribe start-transcription-job \
  --cli-input-json file://mythirdfile.json
```

Inhalt von `mythirdfile.json`:

```
{
  "TranscriptionJobName": "cli-speakerid-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
  },
}
```

```
"Settings":{
  "ShowSpeakerLabels": true,
  "MaxSpeakerLabels": 2
}
```

Ausgabe:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-speakerid-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-17T16:22:59.696000+00:00",
    "CreationTime": "2020-09-17T16:22:59.676000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    }
  }
}
```

Weitere Informationen finden Sie unter [Identifizieren von Sprechern](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 4: Transkribieren einer Audiodatei und Maskieren aller unerwünschten Wörter in der Transkriptionsausgabe

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfourthfile.json
```

Inhalt von `myfourthfile.json`:

```
{
```

```

    "TranscriptionJobName": "cli-filter-mask-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "mask"
    }
}

```

Ausgabe:

```

{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-filter-mask-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "mask"
    }
  }
}

```

Weitere Informationen finden Sie unter [Filtern von Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 5: Transkribieren einer Audiodatei und Entfernen aller unerwünschten Wörter aus der Transkriptionsausgabe

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```
aws transcribe start-transcription-job \
```

```
--cli-input-json file://myfifthfile.json
```

Inhalt von myfifthfile.json:

```
{
  "TranscriptionJobName": "cli-filter-remove-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
  },
  "Settings":{
    "VocabularyFilterName": "your-vocabulary-filter",
    "VocabularyFilterMethod": "remove"
  }
}
```

Ausgabe:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-filter-remove-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "remove"
    }
  }
}
```

Weitere Informationen finden Sie unter [Filtern von Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 6: Transkribieren einer Audiodatei mit höherer Genauigkeit durch Verwendung eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysixthfile.json
```

Inhalt von `mysixthfile.json`:

```
{  
  "TranscriptionJobName": "cli-vocab-job",  
  "LanguageCode": "the-language-of-your-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"  
  },  
  "Settings": {  
    "VocabularyName": "your-vocabulary"  
  }  
}
```

Ausgabe:

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-vocab-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "the-language-of-your-transcription-job",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"  
    },  
    "StartTime": "2020-09-18T16:36:18.568000+00:00",  
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",  
    "Settings": {  
      "VocabularyName": "your-vocabulary"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Filtern von Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 7: Identifizieren der Sprache einer Audiodatei und Transkribieren der Datei

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myseventhfile.json
```

Inhalt von `myseventhfile.json`:

```
{  
  "TranscriptionJobName": "cli-identify-language-transcription-job",  
  "IdentifyLanguage": true,  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-  
name.file-extension"  
  }  
}
```

Ausgabe:

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-identify-language-transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
    },  
    "StartTime": "2020-09-18T22:27:23.970000+00:00",  
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",  
    "IdentifyLanguage": true  
  }  
}
```

Weitere Informationen finden Sie unter [Identifizieren der Sprache](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 8: Transkribieren einer Audiodatei mit unkenntlich gemachten persönlich identifizierbaren Informationen

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei transkribiert und die persönlich identifizierbaren Informationen werden in der Transkriptionsausgabe unkenntlich gemacht.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myeighthfile.json
```

Inhalt von `myeighthfile.json`:

```
{  
  "TranscriptionJobName": "cli-redaction-job",  
  "LanguageCode": "language-code",  
  "Media": {  
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"  
  },  
  "ContentRedaction": {  
    "RedactionOutput": "redacted",  
    "RedactionType": "PII"  
  }  
}
```

Ausgabe:

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-redaction-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"  
    },  
    "StartTime": "2020-09-25T23:49:13.195000+00:00",  
    "CreationTime": "2020-09-25T23:49:13.176000+00:00",  
    "ContentRedaction": {  
      "RedactionType": "PII",  
      "RedactionOutput": "redacted"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Automatische Inhaltsschwärzung](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 9: Um ein Transkript mit geschwärzten personenbezogenen Daten (PII) und einem unredigierten Protokoll zu erstellen

Im folgenden Beispiel für `start-transcription-job` werden zwei Transkriptionen Ihrer Audiodatei generiert, eine mit unkenntlich gemachten persönlich identifizierbaren Informationen und die andere ohne Schwärzungen.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myninthfile.json
```

Inhalt von `myninthfile.json`:

```
{  
  "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",  
  "LanguageCode": "language-code",  
  "Media": {  
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"  
  },  
  "ContentRedaction": {  
    "RedactionOutput": "redacted_and_unredacted",  
    "RedactionType": "PII"  
  }  
}
```

Ausgabe:

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"  
    },  
    "StartTime": "2020-09-25T23:59:47.677000+00:00",  
    "CreationTime": "2020-09-25T23:59:47.653000+00:00",  
    "ContentRedaction": {  
      "RedactionType": "PII",  
      "RedactionOutput": "redacted_and_unredacted"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Automatische Inhaltsschwärzung](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 10: Verwenden eines benutzerdefinierten Sprachmodells, das Sie zuvor erstellt haben, um eine Audiodatei zu transkribieren

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei mit einem benutzerdefinierten Sprachmodell transkribiert, das Sie zuvor erstellt haben.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mytenthfile.json
```

Inhalt von `mytenthfile.json`:

```
{  
  "TranscriptionJobName": "cli-clm-2-job-1",  
  "LanguageCode": "language-code",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"  
  },  
  "ModelSettings": {  
    "LanguageModelName": "cli-clm-2"  
  }  
}
```

Ausgabe:

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-clm-2-job-1",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"  
    },  
    "StartTime": "2020-09-28T17:56:01.835000+00:00",  
    "CreationTime": "2020-09-28T17:56:01.801000+00:00",  
    "ModelSettings": {  
      "LanguageModelName": "cli-clm-2"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie in der Befehlsreferenz. [StartTranscriptionJob](#) AWS CLI

update-medical-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `update-medical-vocabulary`.

AWS CLI

Um ein benutzerdefiniertes medizinisches Vokabular mit neuen Begriffen zu aktualisieren.

Das folgende `update-medical-vocabulary` Beispiel ersetzt die Begriffe, die in einem benutzerdefinierten medizinischen Vokabular verwendet werden, durch die neuen Begriffe. Voraussetzung: Um die Begriffe in einem medizinischen Standardvokabular zu ersetzen, benötigen Sie eine Datei mit neuen Begriffen.

```
aws transcribe update-medical-vocabulary \
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/medical-custom-vocabulary.txt \
  --vocabulary-name medical-custom-vocabulary \
  --language-code language
```

Ausgabe:

```
{
  "VocabularyName": "medical-custom-vocabulary",
  "LanguageCode": "en-US",
  "VocabularyState": "PENDING"
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateMedicalVocabulary](#) in der AWS CLI Befehlsreferenz.

update-vocabulary-filter

Das folgende Codebeispiel zeigt die Verwendung `update-vocabulary-filter`.

AWS CLI

Um die Wörter in einem Vokabelfilter zu ersetzen

Im folgenden `update-vocabulary-filter` Beispiel werden die Wörter in einem Vokabelfilter durch neue Wörter ersetzt. Voraussetzung: Um einen Vokabelfilter mit den neuen Wörtern zu aktualisieren, müssen Sie diese Wörter als Textdatei gespeichert haben.

```
aws transcribe update-vocabulary-filter \  
  --vocabulary-filter-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/your-text-  
file-to-update-your-vocabulary-filter.txt \  
  --vocabulary-filter-name vocabulary-filter-name
```

Ausgabe:

```
{  
  "VocabularyFilterName": "vocabulary-filter-name",  
  "LanguageCode": "language-code",  
  "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"  
}
```

Weitere Informationen finden Sie unter [Filtern unerwünschter Wörter](#) im Amazon Transcribe Developer Guide.

- API-Einheiten finden Sie [UpdateVocabularyFilter](#) in der AWS CLI Befehlsreferenz.

update-vocabulary

Das folgende Codebeispiel zeigt die Verwendung `update-vocabulary`.

AWS CLI

Aktualisieren eines benutzerdefinierten Vokabular mit neuen Begriffen

Im folgenden Beispiel für `update-vocabulary` werden die Begriffe, die zur Erstellung eines benutzerdefinierten Vokabulars verwendet wurden, mit den von Ihnen angegebenen neuen Begriffen überschrieben. Voraussetzung: Um die Begriffe in einem benutzerdefinierten Wortschatz zu ersetzen, benötigen Sie eine Datei mit neuen Begriffen.

```
aws transcribe update-vocabulary \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/custom-  
vocabulary.txt \  
  --vocabulary-name vocabulary-name
```

```
--vocabulary-name custom-vocabulary \  
--language-code language-code
```

Ausgabe:

```
{  
  "VocabularyName": "custom-vocabulary",  
  "LanguageCode": "language",  
  "VocabularyState": "PENDING"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateVocabulary](#) in der AWS CLI Befehlsreferenz.

Amazon Translate Translate-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon Translate Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

import-terminology

Das folgende Codebeispiel zeigt die Verwendung `import-terminology`.

AWS CLI

Um eine benutzerdefinierte Terminologie aus einer Datei zu importieren

Im folgenden `import-terminology` Beispiel wird eine Terminologie erstellt, die `MyTestTerminology` aus der `test-terminology.csv` Datei aufgerufen wird:

```
aws translate import-terminology \  
  --name MyTestTerminology \  
  --description "Creating a test terminology in AWS Translate" \  
  --merge-strategy OVERWRITE \  
  --data-file fileb://test-terminology.csv \  
  --terminology-data Format=CSV
```

Inhalt von `test-terminology.csv`:

en, fr, es, zh Hallo Welt! , Bonjour tout le monde! , Hola Mundo! ,???? Amazon, Amazon, Amazon

Ausgabe:

```
{  
  "TerminologyProperties": {  
    "SourceLanguageCode": "en",  
    "Name": "MyTestTerminology",  
    "TargetLanguageCodes": [  
      "fr",  
      "es",  
      "zh"  
    ],  
    "SizeBytes": 97,  
    "LastUpdatedAt": 1571089500.851,  
    "CreatedAt": 1571089500.851,  
    "TermCount": 6,  
    "Arn": "arn:aws:translate:us-west-2:123456789012:terminology/  
MyTestTerminology/LATEST",  
    "Description": "Creating a test terminology in AWS Translate"  
  }  
}
```

- API-Einheiten finden Sie [ImportTerminology](#) in der AWS CLI Befehlsreferenz.

Trusted Advisor Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren Trusted Advisor.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-organization-recommendation

Das folgende Codebeispiel zeigt die Verwendung `get-organization-recommendation`.

AWS CLI

Um eine Organisationsempfehlung zu erhalten

Im folgenden `get-organization-recommendation` Beispiel wird eine Organisationsempfehlung anhand ihrer ID abgerufen.

```
aws trustedadvisor get-organization-recommendation \  
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-  
  recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

Ausgabe:

```
{  
  "organizationRecommendation": {  
    "arn": "arn:aws:trustedadvisor::organization-recommendation/9534ec9b-  
bf3a-44e8-8213-2ed68b39d9d5",  
    "name": "Lambda Runtime Deprecation Warning",  
    "description": "One or more lambdas are using a deprecated runtime",  
    "awsServices": [  
      "lambda"  
    ],  
    "checkArn": "arn:aws:trustedadvisor::check/L4dfs2Q4C5",  
    "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",  
    "lifecycleStage": "resolved",
```

```

    "pillars": [
      "security"
    ],
    "resourcesAggregates": {
      "errorCount": 0,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "warning",
    "type": "priority"
  }
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [GetOrganizationRecommendation](#) unter AWS CLI Befehlsreferenz.

get-recommendation

Das folgende Codebeispiel zeigt die Verwendung `get-recommendation`.

AWS CLI

Um eine Empfehlung zu erhalten

Im folgenden `get-recommendation` Beispiel wird eine Empfehlung anhand ihrer ID abgerufen.

```

aws trustedadvisor get-recommendation \
  --recommendation-
  identifizier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
  bbb7-491a-833b-5773e9589578

```

Ausgabe:

```

{
  "recommendation": {
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
    bbb7-491a-833b-5773e9589578",
    "name": "MFA Recommendation",
    "description": "Enable multi-factor authentication",

```



```
    "awsServices": [
      "iam"
    ],
    "checkArn": "arn:aws:trustedadvisor:::check/7DAFEemoDos",
    "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
    "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
    "pillarSpecificAggregates": {
      "costOptimizing": {
        "estimatedMonthlySavings": 0.0,
        "estimatedPercentMonthlySavings": 0.0
      }
    },
    "pillars": [
      "security"
    ],
    "resourcesAggregates": {
      "errorCount": 1,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "error",
    "type": "standard"
  }
}
```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [GetRecommendation](#) unter AWS CLI Befehlsreferenz.

list-checks

Das folgende Codebeispiel zeigt die Verwendung `list-checks`.

AWS CLI

Um Trusted Advisor Advisor-Checks aufzulisten

Das folgende `list-checks` Beispiel listet alle Trusted Advisor Advisor-Prüfungen auf.

```
aws trustedadvisor list-checks
```

Ausgabe:

```
{
  "checkSummaries": [
    {
      "arn": "arn:aws:trustedadvisor:::check/1iG5NDGVre",
      "awsServices": [
        "EC2"
      ],
      "description": "Checks security groups for rules that allow unrestricted
access to a resource. Unrestricted access increases opportunities for malicious
activity (hacking, denial-of-service attacks, loss of data)",
      "id": "1iG5NDGVre",
      "metadata": {
        "0": "Region",
        "1": "Security Group Name",
        "2": "Security Group ID",
        "3": "Protocol",
        "4": "Port",
        "5": "Status",
        "6": "IP Range"
      },
      "name": "Security Groups - Unrestricted Access",
      "pillars": [
        "security"
      ],
      "source": "ta_check"
    },
    {
      "arn": "arn:aws:trustedadvisor:::check/1qazXsw23e",
      "awsServices": [
        "RDS"
      ],
      "description": "Checks your usage of RDS and provides recommendations
on purchase of Reserved Instances to help reduce costs incurred from using RDS
On-Demand. AWS generates these recommendations by analyzing your On-Demand usage
for the past 30 days. We then simulate every combination of reservations in the
generated category of usage in order to identify the best number of each type
of Reserved Instance to purchase to maximize your savings. This check covers
recommendations based on partial upfront payment option with 1-year or 3-year
commitment. This check is not available to accounts linked in Consolidated Billing.
Recommendations are only available for the Paying Account.",
      "id": "1qazXsw23e",
      "metadata": {
```

```

        "0": "Region",
        "1": "Family",
        "2": "Instance Type",
        "3": "License Model",
        "4": "Database Edition",
        "5": "Database Engine",
        "6": "Deployment Option",
        "7": "Recommended number of Reserved Instances to purchase",
        "8": "Expected Average Reserved Instance Utilization",
        "9": "Estimated Savings with Recommendation (monthly)"
        "10": "Upfront Cost of Reserved Instances",
        "11": "Estimated cost of Reserved Instances (monthly)",
        "12": "Estimated On-Demand Cost Post Recommended Reserved Instance
Purchase (monthly)",
        "13": "Estimated Break Even (months)",
        "14": "Lookback Period (days)",
        "15": "Term (years)"
    },
    "name": "Amazon Relational Database Service (RDS) Reserved Instance
Optimization",
    "pillars": [
        "cost_optimizing"
    ],
    "source": "ta_check"
},
{
    "arn": "arn:aws:trustedadvisor:::check/1qw23er45t",
    "awsServices": [
        "Redshift"
    ],
    "description": "Checks your usage of Redshift and provides
recommendations on purchase of Reserved Nodes to help reduce costs incurred from
using Redshift On-Demand. AWS generates these recommendations by analyzing your
On-Demand usage for the past 30 days. We then simulate every combination of
reservations in the generated category of usage in order to identify the best
number of each type of Reserved Nodes to purchase to maximize your savings. This
check covers recommendations based on partial upfront payment option with 1-year or
3-year commitment. This check is not available to accounts linked in Consolidated
Billing. Recommendations are only available for the Paying Account.",
    "id": "1qw23er45t",
    "metadata": {
        "0": "Region",
        "1": "Family",
        "2": "Node Type",

```

```

        "3": "Recommended number of Reserved Nodes to purchase",
        "4": "Expected Average Reserved Node Utilization",
        "5": "Estimated Savings with Recommendation (monthly)",
        "6": "Upfront Cost of Reserved Nodes",
        "7": "Estimated cost of Reserved Nodes (monthly)",
        "8": "Estimated On-Demand Cost Post Recommended Reserved Nodes
Purchase (monthly)",
        "9": "Estimated Break Even (months)",
        "10": "Lookback Period (days)",
        "11": "Term (years)",
    },
    "name": "Amazon Redshift Reserved Node Optimization",
    "pillars": [
        "cost_optimizing"
    ],
    "source": "ta_check"
},
],
"nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [ListChecks](#) unter AWS CLI Befehlsreferenz.

list-organization-recommendation-accounts

Das folgende Codebeispiel zeigt die Verwendung `list-organization-recommendation-accounts`.

AWS CLI

Um Empfehlungskonten von Organisationen aufzulisten

Im folgenden `list-organization-recommendation-accounts` Beispiel werden alle Zusammenfassungen von Kontoempfehlungen für eine Organisationsempfehlung nach ihrer ID aufgelistet.

```

aws trustedadvisor list-organization-recommendation-accounts \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-
recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5

```

Ausgabe:

```
{
  "accountRecommendationLifecycleSummaries": [{
    "accountId": "000000000000",
    "accountRecommendationArn":
"arn:aws:trustedadvisor::000000000000:recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
    "lifecycleStage": "resolved",
    "updateReason": "Resolved issue",
    "updateReasonCode": "valid_business_case",
    "lastUpdatedAt": "2023-01-17T18:25:44.552Z"
  }],
  "nextToken": "REDACTED"
}
```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [ListOrganizationRecommendationAccounts](#) unter AWS CLI Befehlsreferenz.

list-organization-recommendation-resources

Das folgende Codebeispiel zeigt die Verwendung `list-organization-recommendation-resources`.

AWS CLI

Um Ressourcen für Organisationsempfehlungen aufzulisten

Im folgenden `list-organization-recommendation-resources` Beispiel werden alle Ressourcen für eine Organisationsempfehlung anhand ihrer ID aufgelistet.

```
aws trustedadvisor list-organization-recommendation-resources \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-
  recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

Ausgabe:

```
{
  "organizationRecommendationResourceSummaries": [
```

```

    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
bb38affc0ce0681d9a6cd13f30238ba03a8f63dfe7a379dc403c619119d86af",
      "awsResourceId": "database-1-instance-1",
      "id":
"bb38affc0ce0681d9a6cd13f302383ba03a8f63dfe7a379dc403c619119d86af",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {
        "0": "14",
        "1": "208.79999999999998",
        "2": "database-1-instance-1",
        "3": "db.r5.large",
        "4": "false",
        "5": "us-west-2",
        "6": "arn:aws:rds:us-west-2:000000000000:db:database-1-instance-1",
        "7": "1"
      },
      "recommendationArn": "arn:aws:trustedadvisor::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
      "regionCode": "us-west-2",
      "status": "warning"
    },
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-
ae72-730598fa89d0/51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
      "awsResourceId": "database-1",
      "id":
"51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {
        "0": "14",
        "1": "31.679999999999996",
        "2": "database-1",
        "3": "db.t3.small",
        "4": "false",
        "5": "us-west-2",
        "6": "arn:aws:rds:us-west-2:000000000000:db:database-1",
        "7": "20"
      },
      "recommendationArn": "arn:aws:trustedadvisor::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
      "regionCode": "us-west-2",

```

```

        "status": "warning"
    },
    {
        "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
        "awsResourceId": "database-2-instance-1-us-west-2a",
        "id":
"f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
        "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
        "metadata": {
            "0": "14",
            "1": "187.200000000000002",
            "2": "database-2-instance-1-us-west-2a",
            "3": "db.r6g.large",
            "4": "true",
            "5": "us-west-2",
            "6": "arn:aws:rds:us-west-2:000000000000:db:database-2-instance-1-
us-west-2a",
            "7": "1"
        },
        "recommendationArn": "arn:aws:trustedadvisor::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
        "regionCode": "us-west-2",
        "status": "warning"
    },
],
"nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [ListOrganizationRecommendationResources](#) unter AWS CLI Befehlsreferenz.

list-organization-recommendations

Das folgende Codebeispiel zeigt die Verwendung `list-organization-recommendations`.

AWS CLI

Beispiel 1: Um Organisationsempfehlungen aufzulisten

Das folgende `list-organization-recommendations` Beispiel listet alle Organisationsempfehlungen auf und enthält keinen Filter.

```
aws trustedadvisor list-organization-recommendations
```

Ausgabe:

```
{
  "organizationRecommendationSummaries": [
    {
      "arn": "arn:aws:trustedadvisor::organization-recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
      "name": "Lambda Runtime Deprecation Warning",
      "awsServices": [
        "lambda"
      ],
      "checkArn": "arn:aws:trustedadvisor::check/L4dfs2Q4C5",
      "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
      "lifecycleStage": "resolved",
      "pillars": [
        "security"
      ],
      "resourcesAggregates": {
        "errorCount": 0,
        "okCount": 0,
        "warningCount": 0
      },
      "source": "ta_check",
      "status": "warning",
      "type": "priority"
    },
    {
      "arn": "arn:aws:trustedadvisor::organization-
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
      "name": "Lambda Runtime Deprecation Warning",
      "awsServices": [
        "lambda"
      ],
      "checkArn": "arn:aws:trustedadvisor::check/L4dfs2Q4C5",
      "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
      "lifecycleStage": "resolved",
      "pillars": [
        "security"
      ]
    }
  ]
}
```



```

    ],
    "resourcesAggregates": {
      "errorCount": 0,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "warning",
    "type": "priority"
  },
],
"nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

Beispiel 2: Um Organisationsempfehlungen mit einem Filter aufzulisten

Das folgende `list-organization-recommendations` Beispiel filtert und gibt maximal eine Organisationsempfehlung zurück, die Teil der Säule „Sicherheit“ ist.

```

aws trustedadvisor list-organization-recommendations \
  --pillar security \
  --max-items 100

```

Ausgabe:

```

{
  "organizationRecommendationSummaries": [{
    "arn": "arn:aws:trustedadvisor::organization-recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
    "name": "Lambda Runtime Deprecation Warning",
    "awsServices": [
      "lambda"
    ],
    "checkArn": "arn:aws:trustedadvisor::check/L4dfs2Q4C5",
    "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
    "lifecycleStage": "resolved",
    "pillars": [
      "security"
    ],
    "resourcesAggregates": {

```

```

        "errorCount": 0,
        "okCount": 0,
        "warningCount": 0
    },
    "source": "ta_check",
    "status": "warning",
    "type": "priority"
}],
"nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

Beispiel 3: Um Organisationsempfehlungen mit einem Paginierungstoken aufzulisten

Im folgenden `list-organization-recommendations` Beispiel wird das von einer vorherigen Anfrage zurückgegebene `nextToken` "" verwendet, um die nächste Seite mit Organisationsempfehlungen abzurufen.

```

aws trustedadvisor list-organization-recommendations \
  --pillar security \
  --max-items 100 \
  --starting-token <next-token>

```

Ausgabe:

```

{
  "organizationRecommendationSummaries": [{
    "arn": "arn:aws:trustedadvisor:::organization-recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
    "name": "Lambda Runtime Deprecation Warning",
    "awsServices": [
      "lambda"
    ],
    "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
    "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
    "lifecycleStage": "resolved",
    "pillars": [
      "security"
    ],
    "resourcesAggregates": {
      "errorCount": 0,

```

```

        "okCount": 0,
        "warningCount": 0
    },
    "source": "ta_check",
    "status": "warning",
    "type": "priority"
  ]
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [ListOrganizationRecommendations](#) unter AWS CLI Befehlsreferenz.

list-recommendation-resources

Das folgende Codebeispiel zeigt die Verwendung `list-recommendation-resources`.

AWS CLI

Um Ressourcen für Empfehlungen aufzulisten

Im folgenden `list-recommendation-resources` Beispiel werden alle Ressourcen für eine Empfehlung anhand ihrer ID aufgelistet.

```

aws trustedadvisor list-recommendation-resources \
  --recommendation-
  identifizier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
  bbb7-491a-833b-5773e9589578

```

Ausgabe:

```

{
  "recommendationResourceSummaries": [
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
      resource/55fa4d2e-
      bbb7-491a-833b-5773e9589578/18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
      "id":
      "18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
      "awsResourceId": "webcms-dev-01",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {

```

```

        "0": "14",
        "1": "123.12000000000002",
        "2": "webcms-dev-01",
        "3": "db.m6i.large",
        "4": "false",
        "5": "us-east-1",
        "6": "arn:aws:rds:us-east-1:000000000000:db:webcms-dev-01",
        "7": "20"
    },
    "recommendationArn":
    "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
    "regionCode": "us-east-1",
    "status": "warning"
},
{
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-bbb7-491a-833b-5773e9589578/
e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcfce4b9e4fefcec9eb63e",
    "id":
    "e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcfce4b9e4fefcec9eb63e",
    "awsResourceId": "aws-dev-db-stack-instance-1",
    "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
    "metadata": {
        "0": "14",
        "1": "29.52",
        "2": "aws-dev-db-stack-instance-1",
        "3": "db.t2.small",
        "4": "false",
        "5": "us-east-1",
        "6": "arn:aws:rds:us-east-1:000000000000:db:aws-dev-db-stack-
instance-1",
        "7": "1"
    },
    "recommendationArn":
    "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
    "regionCode": "us-east-1",
    "status": "warning"
},
{
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459

```

```

        "id":
        "31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459",
        "awsResourceId": "aws-awesome-apps-stack-db",
        "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
        "metadata": {
            "0": "14",
            "1": "114.48000000000002",
            "2": "aws-awesome-apps-stack-db",
            "3": "db.m6g.large",
            "4": "false",
            "5": "us-east-1",
            "6": "arn:aws:rds:us-east-1:000000000000:db:aws-awesome-apps-stack-
db",
            "7": "100"
        },
        "recommendationArn":
        "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
        "regionCode": "us-east-1",
        "status": "warning"
    }
],
    "nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRecommendationResources](#) unter AWS CLI Befehlsreferenz.

list-recommendations

Das folgende Codebeispiel zeigt die Verwendung `list-recommendations`.

AWS CLI

Beispiel 1: Um Empfehlungen aufzulisten

Das folgende `list-recommendations` Beispiel listet alle Empfehlungen auf und enthält keinen Filter.

```
aws trustedadvisor list-recommendations
```

Ausgabe:

```
{
  "recommendationSummaries": [
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
      "name": "MFA Recommendation",
      "awsServices": [
        "iam"
      ],
      "checkArn": "arn:aws:trustedadvisor:::check/7DAFEemoDos",
      "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
      "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
      "pillarSpecificAggregates": {
        "costOptimizing": {
          "estimatedMonthlySavings": 0.0,
          "estimatedPercentMonthlySavings": 0.0
        }
      },
      "pillars": [
        "security"
      ],
      "resourcesAggregates": {
        "errorCount": 1,
        "okCount": 0,
        "warningCount": 0
      },
      "source": "ta_check",
      "status": "error",
      "type": "standard"
    },
    {
      "arn":
"arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
      "name": "RDS clusters quota warning",
      "awsServices": [
        "rds"
      ],
      "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
      "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
      "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
      "pillarSpecificAggregates": {
```

```

        "costOptimizing": {
            "estimatedMonthlySavings": 0.0,
            "estimatedPercentMonthlySavings": 0.0
        }
    },
    "pillars": [
        "service_limits"
    ],
    "resourcesAggregates": {
        "errorCount": 0,
        "okCount": 3,
        "warningCount": 6
    },
    "source": "ta_check",
    "status": "warning",
    "type": "standard"
}
],
"nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

Beispiel 2: Um Empfehlungen mit einem Filter aufzulisten

Das folgende `list-recommendations` Beispiel listet Empfehlungen auf und enthält einen Filter.

```

aws trustedadvisor list-recommendations \
  --aws-service iam \
  --max-items 100

```

Ausgabe:

```

{
  "recommendationSummaries": [{
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
    "name": "MFA Recommendation",
    "awsServices": [
      "iam"
    ],
  ],
}

```

```

    "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
    "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
    "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
    "pillarSpecificAggregates": {
      "costOptimizing": {
        "estimatedMonthlySavings": 0.0,
        "estimatedPercentMonthlySavings": 0.0
      }
    },
    "pillars": [
      "security"
    ],
    "resourcesAggregates": {
      "errorCount": 1,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "error",
    "type": "standard"
  }],
  "nextToken": "REDACTED"
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

Beispiel 3: Um Empfehlungen mit einem Paginierungstoken aufzulisten

Im folgenden `list-recommendations` Beispiel wird das von einer vorherigen Anfrage zurückgegebene `nextToken` "" verwendet, um die nächste Seite mit gefilterten Empfehlungen abzurufen.

```

aws trustedadvisor list-recommendations \
  --aws-service rds \
  --max-items 100 \
  --starting-token <next-token>

```

Ausgabe:

```

{
  "recommendationSummaries": [{

```



```

    "arn":
      "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
      "name": "RDS clusters quota warning",
      "awsServices": [
        "rds"
      ],
      "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
      "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
      "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
      "pillarSpecificAggregates": {
        "costOptimizing": {
          "estimatedMonthlySavings": 0.0,
          "estimatedPercentMonthlySavings": 0.0
        }
      },
      "pillars": [
        "service_limits"
      ],
      "resourcesAggregates": {
        "errorCount": 0,
        "okCount": 3,
        "warningCount": 6
      },
      "source": "ta_check",
      "status": "warning",
      "type": "standard"
    }
  ]
}

```

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [ListRecommendations](#) unter AWS CLI Befehlsreferenz.

update-organization-recommendation-lifecycle

Das folgende Codebeispiel zeigt die Verwendung `update-organization-recommendation-lifecycle`.

AWS CLI

Um den Empfehlungslebenszyklus einer Organisation zu aktualisieren

Im folgenden `update-organization-recommendation-lifecycle` Beispiel wird der Lebenszyklus einer Organisationsempfehlung anhand ihrer ID aktualisiert.

```
aws trustedadvisor update-organization-recommendation-lifecycle \  
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-  
recommendation/96b5e5ca-7930-444c-90c6-06d386128100 \  
  --lifecycle-stage dismissed \  
  --update-reason-code not_applicable
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateOrganizationRecommendationLifecycle](#) unter AWS CLI Befehlsreferenz.

update-recommendation-lifecycle

Das folgende Codebeispiel zeigt die Verwendung `update-recommendation-lifecycle`.

AWS CLI

Um einen Empfehlungslebenszyklus zu aktualisieren

Im folgenden `update-recommendation-lifecycle` Beispiel wird der Lebenszyklus einer Empfehlung anhand ihrer Kennung aktualisiert.

```
aws trustedadvisor update-recommendation-lifecycle \  
  --recommendation-  
identifier arn:aws:trustedadvisor::000000000000:recommendation/861c9c6e-  
f169-405a-8b59-537a8cacc7a \  
  --lifecycle-stage resolved \  
  --update-reason-code valid_business_case
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Erste Schritte mit dem Trusted Advisor API](#) im AWS Trusted Advisor Advisor-Benutzerhandbuch.

- API-Einheiten finden Sie [UpdateRecommendationLifecycle](#) unter AWS CLI Befehlsreferenz.

Beispiele für verifizierte Berechtigungen mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Verified Permissions Aktionen ausführen und gängige Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-identity-source

Das folgende Codebeispiel zeigt die Verwendung `create-identity-source`.

AWS CLI

Um eine Identitätsquelle zu erstellen

Im folgenden `create-identity-source` Beispiel wird eine Identitätsquelle erstellt, mit der Sie auf Identitäten verweisen können, die im angegebenen Amazon Cognito Cognito-Benutzerpool gespeichert sind. Diese Identitäten sind in Verified Permissions als Entitäten des Typs verfügbar.
User

```
aws verifiedpermissions create-identity-source \  
  --configuration file://config.txt \  
  --principal-entity-type "User" \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Inhalt von `config.txt`:

```
{  
  "cognitoUserPoolConfiguration": {
```

```
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
    "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
  }
}
```

Ausgabe:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefgh111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Weitere Informationen zu Identitätsquellen finden Sie unter [Verwenden von Amazon Verified Permissions mit Identitätsanbietern](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [CreateIdentitySource](#) in der AWS CLI Befehlsreferenz.

create-policy-store

Das folgende Codebeispiel zeigt die Verwendung `create-policy-store`.

AWS CLI

Um einen Richtlinienpeicher zu erstellen

Im folgenden `create-policy-store` Beispiel wird ein Richtlinienpeicher in der aktuellen AWS Region erstellt.

```
aws verifiedpermissions create-policy-store \
  --validation-settings "mode=STRICT"
```

Ausgabe:

```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefgh111111",
  "createdDate": "2023-05-16T17:41:29.103459+00:00",
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

```
}
```

Weitere Informationen zu Policy Stores finden Sie unter [Amazon Verified Permissions Policy Stores](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [CreatePolicyStore](#) in der AWS CLI Befehlsreferenz.

create-policy-template

Das folgende Codebeispiel zeigt die Verwendung `create-policy-template`.

AWS CLI

Beispiel 1: Um eine Richtlinienvorlage zu erstellen

Im folgenden `create-policy-template` Beispiel wird eine Richtlinienvorlage mit einer Anweisung erstellt, die einen Platzhalter für den Prinzipal enthält.

```
aws verifiedpermissions create-policy-template \  
  --definition file://template1.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Inhalt der Datei `template1.txt`:

```
permit(  
  principal in ?principal,  
  action == Action::"view",  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Ausgabe:

```
{  
  "createdDate": "2023-06-12T20:47:42.804511+00:00",  
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyTemplateId": "PTEXAMPLEabcdefg111111"  
}
```

Weitere Informationen zu Richtlinienvorlagen finden Sie unter [Richtlinienvorlagen von Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [CreatePolicyTemplate](#) in der AWS CLI Befehlsreferenz.

create-policy

Das folgende Codebeispiel zeigt die Verwendung `create-policy`.

AWS CLI

Beispiel 1: Um eine statische Richtlinie zu erstellen

Im folgenden `create-policy` Beispiel wird eine statische Richtlinie mit einem Richtlinienbereich erstellt, der sowohl einen Prinzipal als auch eine Ressource angibt.

```
aws verifiedpermissions create-policy \  
  --definition file://definition1.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Inhalt der Datei `definition1.txt`:

```
{  
  "static": {  
    "description": "Grant everyone of janeFriends UserGroup access to the  
vacationFolder Album",  
    "statement": "permit(principal in UserGroup::\\"janeFriends\\", action,  
resource in Album::\\"vacationFolder\\" );"  
  }  
}
```

Ausgabe:

```
{  
  "createdDate": "2023-06-12T20:33:37.382907+00:00",  
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",  
  "policyId": "SPEXAMPLEabcdefg111111",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyType": "STATIC",  
  "principal": {  
    "entityId": "janeFriends",  
    "entityType": "UserGroup"  
  },  
  "resource": {
```

```

    "entityId": "vacationFolder",
    "entityType": "Album"
  }
}

```

Beispiel 2: Um eine statische Richtlinie zu erstellen, die allen Benutzern Zugriff auf eine Ressource gewährt

Im folgenden `create-policy` Beispiel wird eine statische Richtlinie mit einem Richtlinienbereich erstellt, der nur eine Ressource angibt.

```

aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111

```

Inhalt der Dateidefinition2.txt:

```

{
  "static": {
    "description": "Grant everyone access to the publicFolder Album",
    "statement": "permit(principal, action, resource in Album:\""publicFolder
\");"
  }
}

```

Ausgabe:

```

{
  "createdDate": "2023-06-12T20:39:44.975897+00:00",
  "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
  "policyId": "PbfR73F8oh5MMfr9uRtFDB",
  "policyStoreId": "PSEXAMPLEEabcdefg222222",
  "policyType": "STATIC",
  "resource": {
    "entityId": "publicFolder",
    "entityType": "Album"
  }
}

```

Beispiel 3: Um eine mit einer Vorlage verknüpfte Richtlinie zu erstellen, die der angegebenen Vorlage zugeordnet ist

Das folgende `create-policy` Beispiel erstellt eine mit einer Vorlage verknüpfte Richtlinie unter Verwendung der angegebenen Richtlinienvorlage und ordnet den angegebenen Prinzipal, der verwendet werden soll, der neuen, mit der Vorlage verknüpften Richtlinie zu.

```
aws verifiedpermissions create-policy \  
  --definition file://definition.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Inhalt von `definition.txt`:

```
{  
  "templateLinked": {  
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",  
    "principal": {  
      "entityType": "User",  
      "entityId": "alice"  
    }  
  }  
}
```

Ausgabe:

```
{  
  "createdDate": "2023-06-12T20:49:51.490211+00:00",  
  "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",  
  "policyId": "TPEXAMPLEEabcdefg111111",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "policyType": "TEMPLATE_LINKED",  
  "principal": {  
    "entityId": "alice",  
    "entityType": "User"  
  },  
  "resource": {  
    "entityId": "VacationPhoto94.jpg",  
    "entityType": "Photo"  
  }  
}
```

Weitere Informationen zu Richtlinien finden Sie unter Richtlinien von [Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [CreatePolicy](#) unter AWS CLI Befehlsreferenz.

delete-identity-source

Das folgende Codebeispiel zeigt die Verwendung `delete-identity-source`.

AWS CLI

Um eine Identitätsquelle zu löschen

Im folgenden `delete-identity-source` Beispiel wird die Identitätsquelle mit der angegebenen ID gelöscht.

```
aws verifiedpermissions delete-identity-source \  
  --identity-source-id IEXAMPLEabcdefgh111111 \  
  --policy-store-id PEXAMPLEabcdefgh111111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Identitätsquellen finden Sie unter [Verwenden von Amazon Verified Permissions mit Identitätsanbietern](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [DeleteIdentitySource](#) in der AWS CLI Befehlsreferenz.

delete-policy-store

Das folgende Codebeispiel zeigt die Verwendung `delete-policy-store`.

AWS CLI

Um einen Richtlinienpeicher zu löschen

Im folgenden `delete-policy-store` Beispiel wird der Richtlinienpeicher mit der angegebenen ID gelöscht.

```
aws verifiedpermissions delete-policy-store \  
  --policy-store-id PEXAMPLEabcdefgh111111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Policy Stores finden Sie unter [Amazon Verified Permissions Policy Stores](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [DeletePolicyStore](#) in der AWS CLI Befehlsreferenz.

delete-policy-template

Das folgende Codebeispiel zeigt die Verwendung `delete-policy-template`.

AWS CLI

Um eine Richtlinienvorlage zu löschen

Im folgenden `delete-policy-template` Beispiel wird die Richtlinienvorlage mit der angegebenen ID gelöscht.

```
aws verifiedpermissions delete-policy \  
  --policy-template-id PTEXAMPLEabcdefgh111111 \  
  --policy-store-id PSEXAMPLEabcdefgh111111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Richtlinienvorlagen finden Sie unter [Richtlinienvorlagen von Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [DeletePolicyTemplate](#) in der AWS CLI Befehlsreferenz.

delete-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-policy`.

AWS CLI

Um eine statische oder mit einer Vorlage verknüpfte Richtlinie zu löschen

Im folgenden `delete-policy` Beispiel wird die Richtlinie mit der angegebenen ID gelöscht.

```
aws verifiedpermissions delete-policy \  
  --policy-id SPEXAMPLEabcdefgh111111 \  
  --policy-store-id PSEXAMPLEabcdefgh111111
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen zu Richtlinien finden Sie unter Richtlinien von [Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [DeletePolicy](#) in der AWS CLI Befehlsreferenz.

get-identity-source

Das folgende Codebeispiel zeigt die Verwendung `get-identity-source`.

AWS CLI

Um Details zu einer Identitätsquelle abzurufen

Im folgenden `get-identity-source` Beispiel werden die Details für die Identitätsquelle mit der angegebenen ID angezeigt.

```
aws verifiedpermissions get-identity-source \  
  --identity-source ISEXAMPLEabcdefg111111 \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Ausgabe:

```
{  
  "createdDate": "2023-06-12T22:27:49.150035+00:00",  
  "details": {  
    "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],  
    "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-  
west-2_1a2b3c4d5",  
    "openIdIssuer": "COGNITO",  
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-  
west-2_1a2b3c4d5"  
  },  
  "identitySourceId": "ISEXAMPLEabcdefg111111",  
  "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "principalEntityType": "User"  
}
```

Weitere Informationen zu Identitätsquellen finden Sie unter [Verwenden von Amazon Verified Permissions mit Identitätsanbietern](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [GetIdentitySource](#) in der AWS CLI Befehlsreferenz.

get-policy-store

Das folgende Codebeispiel zeigt die Verwendung `get-policy-store`.

AWS CLI

Um Details zu einem Richtlinienpeicher abzurufen

Im folgenden `get-policy-store` Beispiel werden die Details für den Richtlinienpeicher mit der angegebenen ID angezeigt.

```
aws verifiedpermissions get-policy-store \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Ausgabe:

```
{  
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PSEXAMPLEEabcdefg111111",  
  "createdDate": "2023-06-05T20:16:46.225598+00:00",  
  "lastUpdatedDate": "2023-06-08T20:40:23.173691+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "validationSettings": { "mode": "OFF" }  
}
```

Weitere Informationen zu Policy Stores finden Sie unter [Amazon Verified Permissions Policy Stores](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [GetPolicyStore](#) in der AWS CLI Befehlsreferenz.

get-policy-template

Das folgende Codebeispiel zeigt die Verwendung `get-policy-template`.

AWS CLI

Um Details zu einer Richtlinienvorlage abzurufen

Im folgenden `get-policy-template` Beispiel werden die Details für die Richtlinienvorlage mit der angegebenen ID angezeigt.

```
aws verifiedpermissions get-policy-template \  
  --policy-template-id PTEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Ausgabe:

```
{
  "createdDate": "2023-06-12T20:47:42.804511+00:00",
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111",
  "statement": "permit(\n  principal in ?principal,\n  action == Action::\n  \"view\", \n  resource == Photo::\"VacationPhoto94.jpg\"\n);"
}
```

Weitere Informationen zu Richtlinienvorlagen finden Sie unter [Richtlinienvorlagen von Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [GetPolicyTemplate](#) in der AWS CLI Befehlsreferenz.

get-policy

Das folgende Codebeispiel zeigt die Verwendung `get-policy`.

AWS CLI

Um Details zu einer Richtlinie abzurufen

Im folgenden `get-policy` Beispiel werden die Details für die Richtlinie mit der angegebenen ID angezeigt.

```
aws verifiedpermissions get-policy \
  --policy-id PSEXAMPLEabcdefg111111 \
  --policy-store-id PSEXAMPLEabcdefg111111
```

Ausgabe:

```
{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "definition": {
    "static": {
      "description": "Grant everyone of janeFriends UserGroup access to the vacationFolder Album",
      "statement": "permit(principal in UserGroup::\"janeFriends\", action, resource in Album::\"vacationFolder\" );"
    }
  },
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
}
```

```

    "policyId": "SPEXAMPLEabcdefg111111",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "STATIC",
    "principal": {
      "entityId": "janeFriends",
      "entityType": "UserGroup"
    },
    "resource": {
      "entityId": "vacationFolder",
      "entityType": "Album"
    }
  }
}

```

Weitere Informationen zu Richtlinien finden Sie unter Richtlinien von [Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [GetPolicy](#) in der AWS CLI Befehlsreferenz.

get-schema

Das folgende Codebeispiel zeigt die Verwendung `get-schema`.

AWS CLI

Um das Schema in einem Richtlinienpeicher abzurufen

Im folgenden `get-schema` Beispiel werden die Details des Schemas im angegebenen Richtlinienpeicher angezeigt.

```

aws verifiedpermissions get-schema \
  --policy-store-id PSEXAMPLEabcdefg111111

```

Ausgabe:

```

{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "schema": "{\n  \"MySampleNamespace\":{\n    \"entityTypes\":{\n      \"Employee\":{\n        \"shape\n        \":{\n          \"attributes\":{\n            \"jobLevel\":{\n              \"type\":\n              \"Long\n            },\n            \"name\":{\n              \"type\":\n              \"String\n            }\n          },\n          \"type\":\n          \"Record\n        }\n      },\n      \"actions\":{\n        \"remoteAccess\":{\n          \"appliesTo\":\n          {\n            \"principalTypes\":[\n              \"Employee\n            ]\n          }\n        }\n      }\n    }\n  }",
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}

```

```
}
```

Weitere Informationen zum Schema finden Sie unter [Policy Store Schema](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [GetSchema](#) unter AWS CLI Befehlsreferenz.

is-authorized-with-token

Das folgende Codebeispiel zeigt die Verwendung `is-authorized-with-token`.

AWS CLI

Beispiel 1: Um eine Autorisierungsentscheidung für eine Benutzeranfrage anzufordern (zulassen)

Im folgenden `is-authorized-with-token` Beispiel wird eine Autorisierungsentscheidung für einen Benutzer angefordert, der von Amazon Cognito authentifiziert wurde. Die Anfrage verwendet das von Cognito bereitgestellte Identitätstoken und nicht das Zugriffstoken. In diesem Beispiel ist der angegebene Informationsspeicher so konfiguriert, dass er Prinzipale als Entitäten des Typs zurückgibt. `CognitoUser`

```
aws verifiedpermissions is-authorized-with-token \  
  --action actionId="View",actionType="Action" \  
  --resource entityId="vacationPhoto94.jpg",entityType="Photo" \  
  --policy-store-id PSEXAMPLEabcdefg111111 \  
  --identity-token "AbCdE12345...long.string...54321EdCbA"
```

Der Richtlinienpeicher enthält eine Richtlinie mit der folgenden Anweisung, die Identitäten aus dem angegebenen Cognito-Benutzerpool und der angegebenen Anwendungs-ID akzeptiert.

```
permit(  
  principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0ka1bmc",  
  action,  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Ausgabe:

```
{  
  "decision":"Allow",
```

```

    "determiningPolicies": [
      {
        "determiningPolicyId": "SPEXAMPLEabcdefg111111"
      }
    ],
    "errors": []
  }

```

Weitere Informationen zur Verwendung von Identitäten aus einem Cognito-Benutzerpool finden Sie unter [Verwenden von Amazon Verified Permissions with Identity Providers](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [IsAuthorizedWithToken](#) in der AWS CLI Befehlsreferenz.

is-authorized

Das folgende Codebeispiel zeigt die Verwendung `is-authorized`.

AWS CLI

Beispiel 1: Um eine Autorisierungsentscheidung für eine Benutzeranfrage anzufordern (zulassen)

Im folgenden `is-authorized` Beispiel wird eine Autorisierungsentscheidung für einen Principal des Typs `User` „Named“ `Alice`, der den `updatePhoto` Vorgang ausführen möchte, für eine Ressource des Typs `Photo` „Named“ angefordert `VactionPhoto94.jpg`.

Die Antwort zeigt, dass die Anforderung durch eine Richtlinie zugelassen ist.

```

aws verifiedpermissions is-authorized \
  --principal entityType=User,entityId=alice \
  --action actionType=Action,actionId=view \
  --resource entityType=Photo,entityId=VactionPhoto94.jpg \
  --policy-store-id PSEXAMPLEabcdefg111111

```

Ausgabe:

```

{
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEabcdefg111111"
    }
  ]
}

```



```
  ],  
  "errors": []  
}
```

Beispiel 2: Um eine Autorisierungsentscheidung für eine Benutzeranfrage anzufordern (ablehnen)

Das folgende Beispiel entspricht dem vorherigen Beispiel, mit dem Unterschied, dass es der Prinzipal ist `User: : "Bob"`. Der Richtlinienpeicher enthält keine Richtlinie, die diesem Benutzer Zugriff auf `Album: : "alice_folder"` gewährt.

Die Ausgabe gibt an, dass das implizit Deny war, weil die Liste von leer `DeterminingPolicies` ist.

```
aws verifiedpermissions create-policy \  
  --definition file://definition2.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Ausgabe:

```
{  
  "decision": "DENY",  
  "determiningPolicies": [],  
  "errors": []  
}
```

Weitere Informationen finden Sie im [Amazon Verified Permissions User Guide](#).

- API-Einheiten finden Sie [IsAuthorized](#) in der AWS CLI Befehlsreferenz.

list-identity-sources

Das folgende Codebeispiel zeigt die Verwendung `list-identity-sources`.

AWS CLI

Um die verfügbaren Identitätsquellen aufzulisten

Im folgenden `list-identity-sources` Beispiel werden alle Identitätsquellen im angegebenen Richtlinienpeicher aufgeführt.

```
aws verifiedpermissions list-identity-sources \  

```

```
--policy-store-id PSEXAMPLEEabcdefg111111
```

Ausgabe:

```
{
  "identitySources": [
    {
      "createdDate": "2023-06-12T22:27:49.150035+00:00",
      "details": {
        "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
        "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_1a2b3c4d5",
        "openIdIssuer": "COGNITO",
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5"
      },
      "identitySourceId": "ISEXAMPLEEabcdefg111111",
      "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg111111",
      "principalEntityType": "User"
    }
  ]
}
```

Weitere Informationen zu Identitätsquellen finden Sie unter [Verwenden von Amazon Verified Permissions mit Identitätsanbietern](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [ListIdentitySources](#) in der AWS CLI Befehlsreferenz.

list-policies

Das folgende Codebeispiel zeigt die Verwendung `list-policies`.

AWS CLI

Um die verfügbaren Richtlinien aufzulisten

Das folgende `list-policies` Beispiel listet alle Richtlinien im angegebenen Richtlinienpeicher auf.

```
aws verifiedpermissions list-policies \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Ausgabe:

```
{
  "policies": [
    {
      "createdDate": "2023-06-12T20:33:37.382907+00:00",
      "definition": {
        "static": {
          "description": "Grant everyone of janeFriends UserGroup access
to the vacationFolder Album"
        }
      },
      "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
      "policyId": "SPEXAMPLEabcdefghijklmnop111111",
      "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111",
      "policyType": "STATIC",
      "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
      },
      "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
      }
    },
    {
      "createdDate": "2023-06-12T20:39:44.975897+00:00",
      "definition": {
        "static": {
          "description": "Grant everyone access to the publicFolder Album"
        }
      },
      "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
      "policyId": "SPEXAMPLEabcdefghijklmnop222222",
      "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111",
      "policyType": "STATIC",
      "resource": {
        "entityId": "publicFolder",
        "entityType": "Album"
      }
    },
    {
      "createdDate": "2023-06-12T20:49:51.490211+00:00",
      "definition": {
```

```
        "templateLinked": {
            "policyTemplateId": "PTEXAMPLEabcdefg111111"
        }
    },
    "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
    "policyId": "SPEXAMPLEabcdefg333333",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyType": "TEMPLATE_LINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
}
]
```

Weitere Informationen zu Richtlinien finden Sie unter Richtlinien von [Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [ListPolicies](#) in der AWS CLI Befehlsreferenz.

list-policy-stores

Das folgende Codebeispiel zeigt die Verwendung `list-policy-stores`.

AWS CLI

Um die verfügbaren Policy Stores aufzulisten

Im folgenden `list-policy-stores` Beispiel werden alle Richtlinienpeicher in der AWS Region aufgeführt. Alle Befehle für Verifizierte Berechtigungen, `list-policy-stores` außer `create-policy-store` dass Sie die ID des RichtlinienSpeichers angeben, mit dem Sie arbeiten möchten.

```
aws verifiedpermissions list-policy-stores
```

Ausgabe:

```
{
  "policyStores": [
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg111111",
      "createdDate": "2023-06-05T20:16:46.225598+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg111111"
    },
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg222222",
      "createdDate": "2023-06-08T18:09:37.364356+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg222222"
    },
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg333333",
      "createdDate": "2023-06-08T18:09:46.920600+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg333333"
    }
  ]
}
```

Weitere Informationen zu Policy Stores finden Sie unter [Amazon Verified Permissions Policy Stores](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [ListPolicyStores](#) in der AWS CLI Befehlsreferenz.

list-policy-templates

Das folgende Codebeispiel zeigt die Verwendung `list-policy-templates`.

AWS CLI

Um die verfügbaren Richtlinienvorlagen aufzulisten

Im folgenden `list-policy-templates` Beispiel werden alle Richtlinienvorlagen im angegebenen Richtlinienspeicher aufgeführt.

```
aws verifiedpermissions list-policy-templates \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Ausgabe:

```
{
  "policyTemplates": [
    {
      "createdDate": "2023-06-12T20:47:42.804511+00:00",
      "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
      "policyStoreId": "PSEXAMPLEabcdefg111111",
      "policyTemplateId": "PTEXAMPLEabcdefg111111"
    }
  ]
}
```

Weitere Informationen zu Richtlinienvorlagen finden Sie unter [Richtlinienvorlagen von Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [ListPolicyTemplates](#) in der AWS CLI Befehlsreferenz.

put - schema

Das folgende Codebeispiel zeigt die Verwendung `put - schema`.

AWS CLI

Um ein Schema in einem Richtlinienspeicher zu speichern

Im folgenden `put - schema` Beispiel wird das Schema im angegebenen Richtlinienspeicher erstellt oder ersetzt.

Der `cedarJson` Parameter in der Eingabedatei verwendet eine Zeichenfolgendarstellung eines JSON Objekts. Er enthält eingebettete Anführungszeichen (,) innerhalb des äußersten Anführungszeichenpaars. Dazu müssen Sie das in eine Zeichenfolge konvertieren JSON, indem Sie allen eingebetteten Anführungszeichen einen umgekehrten Schrägstrich (\) voranstellen und alle Zeilen zu einer einzigen Textzeile ohne Zeilenumbrüche zusammenfassen.

Beispielzeichenfolgen können hier aus Gründen der besseren Lesbarkeit über mehrere Zeilen verteilt angezeigt werden, aber für den Vorgang müssen die Parameter als einzeilige Zeichenfolgen übermittelt werden.

```
aws verifiedpermissions put-schema --Definitionsdatei: //schema.txt -- policy-store-id
PSEXAMPLEabcdefg111111
```

Inhalt von `schema.txt`:

```
{
  "cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
    \"appliesTo\": {\"principalTypes\": [\"Employee\"]}},\"entityTypes\": {
    \"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\":
    \"Long\"}},\"name\": {\"type\": \"String\"}},\"type\": \"Record\"}}}}}"
}
```

Ausgabe:

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "namespaces": [
    "MySampleNamespace"
  ],
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

Weitere Informationen zum Schema finden Sie unter [Policy Store Schema](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [PutSchema](#) unter AWS CLI Befehlsreferenz.

update-identity-source

Das folgende Codebeispiel zeigt die Verwendung `update-identity-source`.

AWS CLI

Um eine Identitätsquelle zu aktualisieren

Im folgenden `update-identity-source` Beispiel wird die angegebene Identitätsquelle geändert, indem eine neue Cognito-Benutzerpoolkonfiguration bereitgestellt und der von der Identitätsquelle zurückgegebene Entitätstyp geändert wird.

```
aws verifiedpermissions update-identity-source
--identity-source-id ISEXAMPLEabcdefg111111 \
--update-configuration file://config.txt \
--principal-entity-type "Employee" \
```

```
--policy-store-id PSEXAMPLEabcdefg111111
```

Inhalt von config.txt:

```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5",
    "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"]
  }
}
```

Ausgabe:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Weitere Informationen zu Identitätsquellen finden Sie unter [Verwenden von Amazon Verified Permissions mit Identitätsanbietern](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [UpdateIdentitySource](#) in der AWS CLI Befehlsreferenz.

update-policy-store

Das folgende Codebeispiel zeigt die Verwendung `update-policy-store`.

AWS CLI

Um einen Richtlinienpeicher zu aktualisieren

Im folgenden `update-policy-store` Beispiel wird ein Richtlinienpeicher geändert, indem seine Validierungseinstellung geändert wird.

```
aws verifiedpermissions update-policy-store \
  --validation-settings "mode=STRICT" \
  --policy-store-id PSEXAMPLEabcdefg111111
```


Ausgabe:

```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
  "createdDate": "2023-05-16T17:41:29.103459+00:00",
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

Weitere Informationen zu Policy Stores finden Sie unter [Amazon Verified Permissions Policy Stores](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [UpdatePolicyStore](#) in der AWS CLI Befehlsreferenz.

update-policy-template

Das folgende Codebeispiel zeigt die Verwendung `update-policy-template`.

AWS CLI

Beispiel 1: Um eine Richtlinienvorlage zu aktualisieren

Im folgenden `update-policy-template` Beispiel wird die angegebene, mit der Vorlage verknüpfte Richtlinie so geändert, dass sie ihre Richtlinienaussage ersetzt.

```
aws verifiedpermissions update-policy-template \
  --policy-template-id PTEXAMPLEabcdefg111111 \
  --statement file://template1.txt \
  --policy-store-id PSEXAMPLEabcdefg111111
```

Inhalt der Datei: `template1.txt`

```
permit(
  principal in ?principal,
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

Ausgabe:

```
{
```

```
"createdDate": "2023-06-12T20:47:42.804511+00:00",  
"lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
"policyStoreId": "PSEXAMPLEEabcdefg111111",  
"policyTemplateId": "PTEXAMPLEEabcdefg111111"  
}
```

Weitere Informationen zu Richtlinienvorlagen finden Sie unter [Richtlinienvorlagen von Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [UpdatePolicyTemplate](#) in der AWS CLI Befehlsreferenz.

update-policy

Das folgende Codebeispiel zeigt die Verwendung `update-policy`.

AWS CLI

Beispiel 1: Um eine statische Richtlinie zu erstellen

Im folgenden `create-policy` Beispiel wird eine statische Richtlinie mit einem Richtlinienbereich erstellt, der sowohl einen Prinzipal als auch eine Ressource angibt.

```
aws verifiedpermissions create-policy \  
  --definition file://definition.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Der `statement` Parameter verwendet eine Zeichenfolgendarstellung eines JSON Objekts. Er enthält eingebettete Anführungszeichen („) innerhalb des äußersten Anführungszeichenpaars. Dazu müssen Sie das in eine Zeichenfolge konvertieren JSON, indem Sie allen eingebetteten Anführungszeichen einen umgekehrten Schrägstrich (") voranstellen und alle Zeilen zu einer einzigen Textzeile ohne Zeilenumbrüche zusammenfassen.

Beispielzeichenfolgen können hier aus Gründen der besseren Lesbarkeit über mehrere Zeilen verteilt angezeigt werden, aber für den Vorgang müssen die Parameter als einzeilige Zeichenfolgen übermittelt werden.

Inhalt der `definition.txt`:

```
{  
  "static": {
```

```

    "description": "Grant everyone of janeFriends UserGroup access to the
vacationFolder Album",
    "statement": "permit(principal in UserGroup::\\"janeFriends\\", action,
resource in Album::\\"vacationFolder\\" );"
  }
}

```

Ausgabe:

```

{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
  "policyId": "SPEXAMPLEEabcdefg111111",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyType": "STATIC",
  "principal": {
    "entityId": "janeFriends",
    "entityType": "UserGroup"
  },
  "resource": {
    "entityId": "vacationFolder",
    "entityType": "Album"
  }
}

```

Beispiel 2: Um eine statische Richtlinie zu erstellen, die allen Benutzern Zugriff auf eine Ressource gewährt

Im folgenden `create-policy` Beispiel wird eine statische Richtlinie mit einem Richtlinienbereich erstellt, der nur eine Ressource angibt.

```

aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111

```

Inhalt der `dateidefinition2.txt`:

```

{
  "static": {
    "description": "Grant everyone access to the publicFolder Album",
    "statement": "permit(principal, action, resource in Album::\\"publicFolder
\\");"
  }
}

```

```
}
}
```

Ausgabe:

```
{
  "createdDate": "2023-06-12T20:39:44.975897+00:00",
  "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
  "policyId": "PbfR73F8oh5MMfr9uRtFDB",
  "policyStoreId": "PSEXAMPLEEabcdefg222222",
  "policyType": "STATIC",
  "resource": {
    "entityId": "publicFolder",
    "entityType": "Album"
  }
}
```

Beispiel 3: Um eine mit einer Vorlage verknüpfte Richtlinie zu erstellen, die der angegebenen Vorlage zugeordnet ist

Das folgende `create-policy` Beispiel erstellt eine mit einer Vorlage verknüpfte Richtlinie unter Verwendung der angegebenen Richtlinienvorlage und ordnet den angegebenen Prinzipal, der verwendet werden soll, der neuen, mit der Vorlage verknüpften Richtlinie zu.

```
aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Inhalt von `definition3.txt`:

```
{
  "templateLinked": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    }
  }
}
```

Ausgabe:

```
{
  "createdDate": "2023-06-12T20:49:51.490211+00:00",
  "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
  "policyId": "TPEXAMPLEabcdefgh111111",
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "policyType": "TEMPLATE_LINKED",
  "principal": {
    "entityId": "alice",
    "entityType": "User"
  },
  "resource": {
    "entityId": "VacationPhoto94.jpg",
    "entityType": "Photo"
  }
}
```

Weitere Informationen zu Richtlinien finden Sie unter Richtlinien von [Amazon Verified Permissions](#) im Amazon Verified Permissions User Guide.

- API-Einheiten finden Sie [UpdatePolicy](#) in der AWS CLI Befehlsreferenz.

VPC-Beispiele für Gitter unter Verwendung AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit VPC Lattice Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-listener

Das folgende Codebeispiel zeigt die Verwendung `create-listener`.

AWS CLI

Um einen Listener zu erstellen

Im folgenden `create-listener` Beispiel wird ein HTTPS Listener mit einer Standardregel erstellt, die den Datenverkehr an die angegebene VPC Lattice-Zielgruppe weiterleitet.

```
aws vpc-lattice create-listener \  
  --name my-service-listener \  
  --protocol HTTPS \  
  --port 443 \  
  --service-identifier svc-0285b53b2eEXAMPLE \  
  --default-action file://listener-config.json
```

Inhalt von `listener-config.json`:

```
{  
  "forward": {  
    "targetGroups": [  
      {  
        "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"  
      }  
    ]  
  }  
}
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE/listener/listener-07cc7fb0abEXAMPLE",  
  "defaultAction": {  
    "forward": {  
      "targetGroups": [  
        {  
          "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE",  
          "weight": 100  
        }  
      ]  
    }  
  },  
  "id": "listener-07cc7fb0abEXAMPLE",
```

```

    "name": "my-service-listener",
    "port": 443,
    "protocol": "HTTPS",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE"
}

```

Weitere Informationen finden Sie unter [Listeners](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateListener](#) in der AWS CLI Befehlsreferenz.

create-service-network-service-association

Das folgende Codebeispiel zeigt die Verwendung `create-service-network-service-association`.

AWS CLI

Um eine Dienstverknüpfung zu erstellen

Im folgenden `create-service-network-service-association` Beispiel wird der angegebene Dienst dem angegebenen Dienstnetzwerk zugeordnet.

```

aws vpc-lattice create-service-network-service-association \
  --service-identifier svc-0285b53b2eEXAMPLE \
  --service-network-identifier sn-080ec7dc93EXAMPLE

```

Ausgabe:

```

{
  "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-0e16955a8cEXAMPLE",
  "createdBy": "123456789012",
  "dnsEntry": {
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
    "hostedZoneId": "Z09127221KTH2CEXAMPLE"
  },
  "id": "snsa-0e16955a8cEXAMPLE",
  "status": "CREATE_IN_PROGRESS"
}

```

Weitere Informationen finden Sie unter [Servicezuordnungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateServiceNetworkServiceAssociation](#) in der AWS CLI Befehlsreferenz.

create-service-network-vpc-association

Das folgende Codebeispiel zeigt die Verwendung `create-service-network-vpc-association`.

AWS CLI

Um eine VPC-Assoziation zu erstellen

Im folgenden `create-service-network-vpc-association`-Beispiel wird die angegebene VPC dem angegebenen Dienstnetzwerk zugeordnet. Die angegebene Sicherheitsgruppe steuert, welche Ressourcen in der auf das Dienstnetzwerk und dessen Dienste zugreifen VPC können.

```
aws vpc-lattice create-service-network-vpc-association \
  --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \
  --service-network-identifier sn-080ec7dc93EXAMPLE \
  --security-group-ids sg-0aee16bc6cEXAMPLE
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/snva-0821fc8631EXAMPLE",
  "createdBy": "123456789012",
  "id": "snva-0821fc8631EXAMPLE",
  "securityGroupIds": [
    "sg-0aee16bc6cEXAMPLE"
  ],
  "status": "CREATE_IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [VPC-Verknüpfungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateServiceNetworkVpcAssociation](#) in der AWS CLI Befehlsreferenz.

create-service-network

Das folgende Codebeispiel zeigt die Verwendung `create-service-network`.

AWS CLI

Um ein Servicenetzwerk zu erstellen

Im folgenden `create-service-network` Beispiel wird ein Dienstnetzwerk mit dem angegebenen Namen erstellt.

```
aws vpc-lattice create-service-network \  
  --name my-service-network
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/  
sn-080ec7dc93EXAMPLE",  
  "authType": "NONE",  
  "id": "sn-080ec7dc93EXAMPLE",  
  "name": "my-service-network"  
}
```

Weitere Informationen finden Sie unter [Servicenetzwerke](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateServiceNetwork](#) in der AWS CLI Befehlsreferenz.

create-service

Das folgende Codebeispiel zeigt die Verwendung `create-service`.

AWS CLI

Um einen Dienst zu erstellen

Im folgenden `create-service` Beispiel wird ein Dienst mit dem angegebenen Namen erstellt.

```
aws vpc-lattice create-service \  
  --name my-service
```

```
--name my-lattice-service
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
  "authType": "NONE",
  "dnsEntry": {
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.1a2b3c4.vpc-lattice-
svcs.us-east-2.on.aws",
    "hostedZoneId": "Z09127221KTH2CEXAMPLE"
  },
  "id": "svc-0285b53b2eEXAMPLE",
  "name": "my-lattice-service",
  "status": "CREATE_IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [Services in VPC Lattice](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie unter [CreateService AWS CLI](#) Befehlsreferenz.

create-target-group

Das folgende Codebeispiel zeigt die Verwendung `create-target-group`.

AWS CLI

Beispiel 1: Um eine Zielgruppe vom Typ `INSTANCE` zu erstellen

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe mit dem angegebenen Namen, Typ und Konfiguration erstellt.

```
aws vpc-lattice create-target-group \
  --name my-lattice-target-group-instance \
  --type INSTANCE \
  --config file://tg-config.json
```

Inhalt von `tg-config.json`:

```
{
```

```
"port": 443,  
"protocol": "HTTPS",  
"protocolVersion": "HTTP1",  
"vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "config": {  
    "healthCheck": {  
      "enabled": true,  
      "healthCheckIntervalSeconds": 30,  
      "healthCheckTimeoutSeconds": 5,  
      "healthyThresholdCount": 5,  
      "matcher": {  
        "httpCode": "200"  
      },  
      "path": "/",  
      "protocol": "HTTPS",  
      "protocolVersion": "HTTP1",  
      "unhealthyThresholdCount": 2  
    },  
    "port": 443,  
    "protocol": "HTTPS",  
    "protocolVersion": "HTTP1",  
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
  },  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "name": "my-lattice-target-group-instance",  
  "status": "CREATE_IN_PROGRESS",  
  "type": "INSTANCE"  
}
```

Beispiel 2: Um eine Zielgruppe vom Typ IP zu erstellen

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe mit dem angegebenen Namen, Typ und Konfiguration erstellt.

```
aws vpc-lattice create-target-group \
```

```
--name my-lattice-target-group-ip \  
--type IP \  
--config file://tg-config.json
```

Inhalt von `tg-config.json`:

```
{  
  "ipAddressType": "IPV4",  
  "port": 443,  
  "protocol": "HTTPS",  
  "protocolVersion": "HTTP1",  
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "config": {  
    "healthCheck": {  
      "enabled": true,  
      "healthCheckIntervalSeconds": 30,  
      "healthCheckTimeoutSeconds": 5,  
      "healthyThresholdCount": 5,  
      "matcher": {  
        "httpCode": "200"  
      },  
      "path": "/",  
      "protocol": "HTTPS",  
      "protocolVersion": "HTTP1",  
      "unhealthyThresholdCount": 2  
    },  
    "ipAddressType": "IPV4",  
    "port": 443,  
    "protocol": "HTTPS",  
    "protocolVersion": "HTTP1",  
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
  },  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "name": "my-lattice-target-group-ip",  
  "status": "CREATE_IN_PROGRESS",  
  "type": "IP"  
}
```

```
}
```

Beispiel 3: Um eine Zielgruppe vom Typ zu erstellen LAMBDA

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe mit dem angegebenen Namen, Typ und Konfiguration erstellt.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-lambda \  
  --type LAMBDA
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "name": "my-lattice-target-group-lambda",  
  "status": "CREATE_IN_PROGRESS",  
  "type": "LAMBDA"  
}
```

Beispiel 4: Um eine Zielgruppe vom Typ zu erstellen ALB

Im folgenden `create-target-group` Beispiel wird eine Zielgruppe mit dem angegebenen Namen, Typ und Konfiguration erstellt.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-alb \  
  --type ALB \  
  --config file://tg-config.json
```

Inhalt von `tg-config.json`:

```
{  
  "port": 443,  
  "protocol": "HTTPS",  
  "protocolVersion": "HTTP1",  
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "config": {
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
  },
  "id": "tg-0eaa4b9ab4EXAMPLE",
  "name": "my-lattice-target-group-alb",
  "status": "CREATE_IN_PROGRESS",
  "type": "ALB"
}
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [CreateTargetGroup](#) in der AWS CLI Befehlsreferenz.

delete-auth-policy

Das folgende Codebeispiel zeigt die Verwendung `delete-auth-policy`.

AWS CLI

Um eine Authentifizierungsrichtlinie zu löschen

Im folgenden `delete-auth-policy` Beispiel wird die Authentifizierungsrichtlinie für den angegebenen Dienst gelöscht.

```
aws vpc-lattice delete-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Authentifizierungsrichtlinien](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteAuthPolicy](#) in der AWS CLI Befehlsreferenz.

delete-listener

Das folgende Codebeispiel zeigt die Verwendung `delete-listener`.

AWS CLI

Um einen Listener zu löschen

Im folgenden `delete-listener` Beispiel wird der angegebene Listener gelöscht.

```
aws vpc-lattice delete-listener \  
  --listener-identifizier listener-07cc7fb0abEXAMPLE \  
  --service-identifizier svc-0285b53b2eEXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Listeners](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteListener](#) in der AWS CLI Befehlsreferenz.

delete-service-network-service-association

Das folgende Codebeispiel zeigt die Verwendung `delete-service-network-service-association`.

AWS CLI

Um eine Dienstverknüpfung zu löschen

Im folgenden `delete-service-network-service-association` Beispiel wird die Zuordnung der angegebenen Dienstzuordnung aufgehoben.

```
aws vpc-lattice delete-service-network-service-association \  
  --service-network-service-association-identifizier snsa-031fabb4d8EXAMPLE
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-  
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",  
  "id": "snsa-031fabb4d8EXAMPLE",
```

```
"status": "DELETE_IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [Servicezuordnungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteServiceNetworkServiceAssociation](#) in der AWS CLI Befehlsreferenz.

delete-service-network-vpc-association

Das folgende Codebeispiel zeigt die Verwendung `delete-service-network-vpc-association`.

AWS CLI

Um eine VPC Assoziation zu löschen

Im folgenden `delete-service-network-vpc-association` Beispiel wird die Zuordnung der angegebenen VPC Zuordnung aufgehoben.

```
aws vpc-lattice delete-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
  "id": "snva-0821fc8631EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

Weitere Informationen finden Sie unter [VPC-Verknüpfungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteServiceNetworkVpcAssociation](#) in der AWS CLI Befehlsreferenz.

delete-service-network

Das folgende Codebeispiel zeigt die Verwendung `delete-service-network`.

AWS CLI

Um ein Servicenetzwerk zu löschen

Im folgenden `delete-service-network` Beispiel wird das angegebene Dienstnetzwerk gelöscht.

```
aws vpc-lattice delete-service-network \  
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Servicenetzwerke](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteServiceNetwork](#) in der AWS CLI Befehlsreferenz.

`delete-service`

Das folgende Codebeispiel zeigt die Verwendung `delete-service`.

AWS CLI

Um einen Dienst zu löschen

Im folgenden `delete-service` Beispiel wird der angegebene Dienst gelöscht.

```
aws vpc-lattice delete-service \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/  
svc-0285b53b2eEXAMPLE",  
  "id": "svc-0285b53b2eEXAMPLE",  
  "name": "my-lattice-service",  
  "status": "DELETE_IN_PROGRESS"  
}
```

Weitere Informationen finden Sie unter [Services in VPC Lattice](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie unter [DeleteService AWS CLI](#) Befehlsreferenz.

delete-target-group

Das folgende Codebeispiel zeigt die Verwendung `delete-target-group`.

AWS CLI

Um eine Zielgruppe zu löschen

Im folgenden `delete-target-group` Beispiel wird die angegebene Zielgruppe gelöscht.

```
aws vpc-lattice delete-target-group \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "status": "DELETE_IN_PROGRESS"  
}
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeleteTargetGroup](#) in der AWS CLI Befehlsreferenz.

deregister-targets

Das folgende Codebeispiel zeigt die Verwendung `deregister-targets`.

AWS CLI

Um ein Ziel zu deregistrieren

Im folgenden `deregister-targets` Beispiel wird die Registrierung des angegebenen Ziels von der angegebenen Zielgruppe aufgehoben.

```
aws vpc-lattice deregister-targets \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

```
--targets i-07dd579bc5EXAMPLE \  
--target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Ausgabe:

```
{  
  "successful": [  
    {  
      "id": "i-07dd579bc5EXAMPLE",  
      "port": 443  
    }  
  ],  
  "unsuccessful": []  
}
```

Weitere Informationen finden Sie unter [Ziele registrieren](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [DeregisterTargets](#) in der AWS CLI Befehlsreferenz.

get-auth-policy

Das folgende Codebeispiel zeigt die Verwendung `get-auth-policy`.

AWS CLI

Um Informationen über eine Authentifizierungsrichtlinie zu erhalten

Im folgenden `get-auth-policy` Beispiel werden Informationen zur Authentifizierungsrichtlinie für den angegebenen Dienst abgerufen.

```
aws vpc-lattice get-auth-policy \  
--resource-identifier svc-0285b53b2eEXAMPLE
```

Ausgabe:

```
{  
  "createdAt": "2023-06-07T03:51:20.266Z",  
  "lastUpdatedAt": "2023-06-07T04:39:27.082Z",  
  "policy": "{\n    \"Version\": \"2012-10-17\",  
    \"Statement\": [\n      {\n        \"Effect\": \"Allow\",  
        \"Principal\": {\n          \"AWS\": \"arn:aws:iam::123456789012:role/my-clients\"  
        }\n      }\n    ]  
  }"
```

```

{"Action": "\vpc-lattice-svcs:Invoke", "Resource": "\arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE"}],
  "state": "Active"
}

```

Weitere Informationen finden Sie unter [Authentifizierungsrichtlinien](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetAuthPolicy](#) in der AWS CLI Befehlsreferenz.

get-listener

Das folgende Codebeispiel zeigt die Verwendung `get-listener`.

AWS CLI

Um Informationen über einen Service-Listener abzurufen

Im folgenden `get-listener` Beispiel werden Informationen über den angegebenen Listener für den angegebenen Dienst abgerufen.

```

aws vpc-lattice get-listener \
  --listener-identifizier listener-0ccf55918cEXAMPLE \
  --service-identifizier svc-0285b53b2eEXAMPLE

```

Ausgabe:

```

{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
  "createdAt": "2023-05-07T05:08:45.192Z",
  "defaultAction": {
    "forward": {
      "targetGroups": [
        {
          "targetGroupIdentifizier": "tg-0ff213abb6EXAMPLE",
          "weight": 1
        }
      ]
    }
  },
  "id": "listener-0ccf55918cEXAMPLE",
}

```

```
"lastUpdatedAt": "2023-05-07T05:08:45.192Z",
"name": "http-80",
"port": 80,
"protocol": "HTTP",
"serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
"serviceId": "svc-0285b53b2eEXAMPLE"
}
```

Weitere Informationen finden Sie unter [Routing definieren](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetListener](#) in der AWS CLI Befehlsreferenz.

get-service-network-service-association

Das folgende Codebeispiel zeigt die Verwendung `get-service-network-service-association`.

AWS CLI

Um Informationen über einen Serviceverband zu erhalten

Im folgenden `get-service-network-service-association` Beispiel werden Informationen über die angegebene Dienstverknüpfung abgerufen.

```
aws vpc-lattice get-service-network-service-association \
--service-network-service-association-identifizier snsa-031fabb4d8EXAMPLE
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
  "createdAt": "2023-05-05T21:48:16.076Z",
  "createdBy": "123456789012",
  "dnsEntry": {
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
    "hostedZoneId": "Z09127221KTH2CEXAMPLE"
  },
}
```

```
"id": "snsa-031fabb4d8EXAMPLE",
"serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
"serviceId": "svc-0285b53b2eEXAMPLE",
"serviceName": "my-lattice-service",
"serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
"serviceNetworkId": "sn-080ec7dc93EXAMPLE",
"serviceNetworkName": "my-service-network",
"status": "ACTIVE"
}
```

Weitere Informationen finden Sie unter [Servicezuordnungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetServiceNetworkServiceAssociation](#) in der AWS CLI Befehlsreferenz.

get-service-network-vpc-association

Das folgende Codebeispiel zeigt die Verwendung `get-service-network-vpc-association`.

AWS CLI

Um Informationen über eine VPC Assoziation zu erhalten

Im folgenden `get-service-network-vpc-association` Beispiel werden Informationen über die angegebene VPC Assoziation abgerufen.

```
aws vpc-lattice get-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
  "createdAt": "2023-06-06T23:41:08.421Z",
  "createdBy": "123456789012",
  "id": "snva-0c5dcb60d6EXAMPLE",
  "lastUpdatedAt": "2023-06-06T23:41:08.421Z",
```

```
"securityGroupIds": [
  "sg-0aee16bc6cEXAMPLE"
],
"serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
"serviceNetworkId": "sn-080ec7dc93EXAMPLE",
"serviceNetworkName": "my-service-network",
"status": "ACTIVE",
"vpcId": "vpc-0a1b2c3d4eEXAMPLE"
}
```

Weitere Informationen finden Sie unter [VPCVerknüpfungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetServiceNetworkVpcAssociation](#) in der AWS CLI Befehlsreferenz.

get-service-network

Das folgende Codebeispiel zeigt die Verwendung `get-service-network`.

AWS CLI

Um Informationen über ein Servicenetzwerk zu erhalten

Im folgenden `get-service-network` Beispiel werden Informationen über das angegebene Dienstnetzwerk abgerufen.

```
aws vpc-lattice get-service-network \
  --service-network-identifizier sn-080ec7dc93EXAMPLE
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
  "authType": "AWS_IAM",
  "createdAt": "2023-05-05T15:26:08.417Z",
  "id": "sn-080ec7dc93EXAMPLE",
  "lastUpdatedAt": "2023-05-05T15:26:08.417Z",
  "name": "my-service-network",
  "numberOfAssociatedServices": 2,
  "numberOfAssociatedVPCs": 3
}
```

```
}
```

Weitere Informationen finden Sie unter [Servicenetze](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetServiceNetwork](#) in der AWS CLI Befehlsreferenz.

get-service

Das folgende Codebeispiel zeigt die Verwendung `get-service`.

AWS CLI

Um Informationen über einen Dienst zu erhalten

Im folgenden `get-service` Beispiel werden Informationen über den angegebenen Dienst abgerufen.

```
aws vpc-lattice get-service \  
--service-identifizier svc-0285b53b2eEXAMPLE
```

Ausgabe:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE",  
  "authType": "AWS_IAM",  
  "createdAt": "2023-05-05T21:35:29.339Z",  
  "dnsEntry": {  
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-  
svcs.us-east-2.on.aws",  
    "hostedZoneId": "Z09127221KTH2CFU0HIZH"  
  },  
  "id": "svc-0285b53b2eEXAMPLE",  
  "lastUpdatedAt": "2023-05-05T21:35:29.339Z",  
  "name": "my-lattice-service",  
  "status": "ACTIVE"  
}
```

Weitere Informationen finden Sie unter [Services](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetService](#) in der AWS CLI Befehlsreferenz.

get-target-group

Das folgende Codebeispiel zeigt die Verwendung `get-target-group`.

AWS CLI

Um Informationen über eine Zielgruppe zu erhalten

Im folgenden `get-target-group` Beispiel werden Informationen über die angegebene Zielgruppe abgerufen, die den Zieltyp `INSTANCE` hat.

```
aws vpc-lattice get-target-group \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Ausgabe:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "config": {
    "healthCheck": {
      "enabled": true,
      "healthCheckIntervalSeconds": 30,
      "healthCheckTimeoutSeconds": 5,
      "healthyThresholdCount": 5,
      "matcher": {
        "httpCode": "200"
      },
      "path": "/",
      "protocol": "HTTPS",
      "protocolVersion": "HTTP1",
      "unhealthyThresholdCount": 2
    },
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
  },
  "createdAt": "2023-05-06T04:41:04.122Z",
  "id": "tg-0eaa4b9ab4EXAMPLE",
  "lastUpdatedAt": "2023-05-06T04:41:04.122Z",
  "name": "my-target-group",
  "serviceArns": [
```

```
    "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE"
  ],
  "status": "ACTIVE",
  "type": "INSTANCE"
}
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [GetTargetGroup](#) in der AWS CLI Befehlsreferenz.

list-listeners

Das folgende Codebeispiel zeigt die Verwendung `list-listeners`.

AWS CLI

Um Service-Listener aufzulisten

Das folgende `list-listeners` Beispiel listet die Listener für den angegebenen Dienst auf.

```
aws vpc-lattice list-listeners \
  --service-identifier svc-0285b53b2eEXAMPLE
```

Ausgabe:

```
{
  "items": [
    {
      "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
      "createdAt": "2023-05-07T05:08:45.192Z",
      "id": "listener-0ccf55918cEXAMPLE",
      "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
      "name": "http-80",
      "port": 80,
      "protocol": "HTTP"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Routing definieren](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListListeners](#) in der AWS CLI Befehlsreferenz.

list-service-network-service-associations

Das folgende Codebeispiel zeigt die Verwendung `list-service-network-service-associations`.

AWS CLI

Um Dienstzuordnungen aufzulisten

Im folgenden `list-service-network-service-associations` Beispiel werden die Dienstzuordnungen für das angegebene Dienstnetzwerk aufgeführt. Die `--query` Option beschränkt die Ausgabe auf die IDs der Dienstzuordnungen.

```
aws vpc-lattice list-service-network-service-associations \
  --service-network-identifier sn-080ec7dc93EXAMPLE \
  --query items[*].id
```

Ausgabe:

```
[
  "snsa-031fabb4d8EXAMPLE",
  "snsa-0e16955a8cEXAMPLE"
]
```

Weitere Informationen finden Sie unter [Servicezuordnungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListServiceNetworkServiceAssociations](#) in der AWS CLI Befehlsreferenz.

list-service-network-vpc-associations

Das folgende Codebeispiel zeigt die Verwendung `list-service-network-vpc-associations`.

AWS CLI

Um VPC Assoziationen aufzulisten

Im folgenden `list-service-network-vpc-associations` Beispiel werden die VPC Zuordnungen für das angegebene Dienstnetzwerk aufgeführt. Die `--query` Option beschränkt die Ausgabe auf die IDs der VPC Zuordnungen.

```
aws vpc-lattice list-service-network-vpc-associations \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --query items[*].id
```

Ausgabe:

```
[  
  "snva-0821fc8631EXAMPLE",  
  "snva-0c5dcb60d6EXAMPLE"  
]
```

Weitere Informationen finden Sie unter [VPCVerknüpfungen verwalten](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListServiceNetworkVpcAssociations](#) in der AWS CLI Befehlsreferenz.

list-service-networks

Das folgende Codebeispiel zeigt die Verwendung `list-service-networks`.

AWS CLI

Um Ihre Servicenetzwerke aufzulisten

Im folgenden `list-service-networks` Beispiel werden die Dienstnetzwerke aufgeführt, die dem anrufenden Konto gehören oder von diesem gemeinsam genutzt werden. Die `--query` Option beschränkt die Ergebnisse auf die Amazon-Ressourcennamen (ARN) der Servicenetzwerke.

```
aws vpc-lattice list-service-networks \  
  --query items[*].arn
```

Ausgabe:

```
[  
  "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/  
  sn-080ec7dc93EXAMPLE",  
]
```

```
"arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"  
]
```

Weitere Informationen finden Sie unter [Servicenetze](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListServiceNetworks](#) in der AWS CLI Befehlsreferenz.

list-services

Das folgende Codebeispiel zeigt die Verwendung `list-services`.

AWS CLI

Um Ihre Dienste aufzulisten

Im folgenden `list-services` Beispiel werden die Dienste aufgeführt, die dem aufrufenden Konto gehören oder mit diesem geteilt werden. Die `--query` Option beschränkt die Ergebnisse auf die Amazon-Ressourcennamen (ARN) der Services.

```
aws vpc-lattice list-services \  
  --query items[*].arn
```

Ausgabe:

```
[  
  "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",  
  "arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"  
]
```

Weitere Informationen finden Sie unter [Services](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListServices](#) in der AWS CLI Befehlsreferenz.

list-target-groups

Das folgende Codebeispiel zeigt die Verwendung `list-target-groups`.

AWS CLI

Um Ihre Zielgruppen aufzulisten

Das folgende `list-target-groups` Beispiel listet die Zielgruppen mit dem Zieltyp auf LAMBDA.

```
aws vpc-lattice list-target-groups \  
  --target-group-type LAMBDA
```

Ausgabe:

```
{  
  "items": [  
    {  
      "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-045c1b7d9dEXAMPLE",  
      "createdAt": "2023-05-06T05:22:16.637Z",  
      "id": "tg-045c1b7d9dEXAMPLE",  
      "lastUpdatedAt": "2023-05-06T05:22:16.637Z",  
      "name": "my-target-group-lam",  
      "serviceArns": [  
        "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE"  
      ],  
      "status": "ACTIVE",  
      "type": "LAMBDA"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTargetGroups](#) in der AWS CLI Befehlsreferenz.

list-targets

Das folgende Codebeispiel zeigt die Verwendung `list-targets`.

AWS CLI

Um die Ziele für eine Zielgruppe aufzulisten

Das folgende `list-targets` Beispiel listet die Ziele für die angegebene Zielgruppe auf.

```
aws vpc-lattice list-targets \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Ausgabe:

```
{
  "items": [
    {
      "id": "i-07dd579bc5EXAMPLE",
      "port": 443,
      "status": "HEALTHY"
    },
    {
      "id": "i-047b3c9078EXAMPLE",
      "port": 443,
      "reasonCode": "HealthCheckFailed",
      "status": "UNHEALTHY"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Zielgruppen](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [ListTargets](#) in der AWS CLI Befehlsreferenz.

put-auth-policy

Das folgende Codebeispiel zeigt die Verwendung `put-auth-policy`.

AWS CLI

Um eine Authentifizierungsrichtlinie für einen Dienst zu erstellen

Das folgende `put-auth-policy` Beispiel gewährt Zugriff auf Anfragen von jedem authentifizierten Prinzipal, der die angegebene IAM Rolle verwendet. Die Ressource ist die ARN des Dienstes, an den die Richtlinie angehängt ist.

```
aws vpc-lattice put-auth-policy \
  --resource-identifizier svc-0285b53b2eEXAMPLE \
  --policy file://auth-policy.json
```

Inhalt von `auth-policy.json`:

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:role/my-clients"
    },
    "Action": "vpc-lattice-svcs:Invoke",
    "Resource": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE"
  }
]
}

```

Ausgabe:

```

{
  "policy": "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\":\"Allow\",
\", \"Principal\": {\"AWS\": \"arn:aws:iam::123456789012:role/my-clients\"},
\", \"Action\": \"vpc-lattice-svcs:Invoke\", \"Resource\": \"arn:aws:vpc-lattice:us-
east-2:123456789012:service/svc-0285b53b2eEXAMPLE\"}]}",
  "state": "Active"
}

```

Weitere Informationen finden Sie unter [Authentifizierungsrichtlinien](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [PutAuthPolicy](#) in der AWS CLI Befehlsreferenz.

register-targets

Das folgende Codebeispiel zeigt die Verwendung `register-targets`.

AWS CLI

Um ein Ziel zu registrieren

Im folgenden `register-targets` Beispiel werden die angegebenen Ziele bei der angegebenen Zielgruppe registriert.

```

aws vpc-lattice register-targets \
  --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \

```



```
--target-group-identifizier tg-0eaa4b9ab4EXAMPLE
```

Ausgabe:

```
{
  "successful": [
    {
      "id": "i-07dd579bc5EXAMPLE",
      "port": 443
    }
  ],
  "unsuccessful": [
    {
      "failureCode": "UnsupportedTarget",
      "failureMessage": "Instance targets must be in the same VPC as their
target group",
      "id": "i-047b3c9078EXAMPLE",
      "port": 443
    }
  ]
}
```

Weitere Informationen finden Sie unter [Ziele registrieren](#) im Amazon VPC Lattice-Benutzerhandbuch.

- API-Einheiten finden Sie [RegisterTargets](#) in der AWS CLI Befehlsreferenz.

AWS WAF Classic Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS WAF Classic.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

put-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-logging-configuration`.

AWS CLI

Um eine Logging-Konfiguration für das Web ACL ARN mit dem angegebenen Kinesis Firehose-Stream zu erstellen ARN

Das folgende `put-logging-configuration` Beispiel zeigt die Logging-Konfiguration für WAF with. CloudFront

```
aws waf put-logging-configuration \  
  --logging-configuration ResourceArn=arn:aws:waf::123456789012:webacl/3bffd3ed-  
fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-  
east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[]
```

Ausgabe:

```
{  
  "LoggingConfiguration": {  
    "ResourceArn": "arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-  
a6a7cf153fd3",  
    "LogDestinationConfigs": [  
      "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-  
firehose-stream"  
    ]  
  }  
}
```

- API-Einheiten finden Sie [PutLoggingConfiguration](#) in der AWS CLI Befehlsreferenz.

update-byte-match-set

Das folgende Codebeispiel zeigt die Verwendung `update-byte-match-set`.

AWS CLI

Um ein Byte-Match-Set zu aktualisieren

Der folgende `update-byte-match-set` Befehl löscht ein `ByteMatchTuple` Objekt (Filter) in einem `ByteMatchSet`:

```
aws waf update-byte-match-set --byte-match-set-id a123fae4-  
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --  
updates  
Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="b
```

Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch unter [Working with String Match Conditions](#).

- API Einzelheiten finden Sie [UpdateByteMatchSet](#) in der AWS CLI Befehlsreferenz.

update-ip-set

Das folgende Codebeispiel zeigt die Verwendung `update-ip-set`.

AWS CLI

Um einen IP-Satz zu aktualisieren

Der folgende `update-ip-set` Befehl aktualisiert eine IPSet mit einer IPv4 Adresse und löscht eine IPv6 Adresse:

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90  
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates  
Action="INSERT",IPSetDescriptor={Type="IPV4",Value="12.34.56.78/16"},Action="DELETE",IPSetD
```

Alternativ können Sie eine JSON Datei verwenden, um die Eingabe zu spezifizieren.

Beispielsweise:

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-  
token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates file://change.json
```

Wo der Inhalt der JSON Datei ist:

```
[  
{  
  "Action": "INSERT",  
  "IPSetDescriptor":
```

```
{
  "Type": "IPV4",
  "Value": "12.34.56.78/16"
},
{
  "Action": "DELETE",
  "IPSetDescriptor": {
    "Type": "IPV6",
    "Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
  }
}
]
```

Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch unter Arbeiten mit IP-Match-Bedingungen.

- API-Einheiten finden Sie [UpdateIPSet](#) in der AWS CLI Befehlsreferenz.

update-rule

Das folgende Codebeispiel zeigt die Verwendung `update-rule`.

AWS CLI

Um eine Regel zu aktualisieren

Mit dem folgenden `update-rule` Befehl wird ein Predicate-Objekt in einer Regel gelöscht:

```
aws waf update-rule --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}
```

Weitere Informationen finden Sie unter Arbeiten mit Regeln im AWS WAF Entwicklerhandbuch.

- API-Einheiten finden Sie [UpdateRule](#) in der AWS CLI Befehlsreferenz.

update-size-constraint-set

Das folgende Codebeispiel zeigt die Verwendung `update-size-constraint-set`.

AWS CLI

Um einen Größenbeschränkungssatz zu aktualisieren

Der folgende `update-size-constraint-set` Befehl löscht ein `SizeConstraint` Objekt (Filter) in einem Größenbeschränkungssatz:

```
aws waf update-size-constraint-set --size-constraint-set-id a123fae4-  
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --  
updates  
Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE"
```

Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch unter Arbeiten mit Größenbeschränkungsbedingungen.

- API Einzelheiten finden Sie [UpdateSizeConstraintSet](#) in der AWS CLI Befehlsreferenz.

update-sql-injection-match-set

Das folgende Codebeispiel zeigt die Verwendung `update-sql-injection-match-set`.

AWS CLI

Um ein SQL Injection Match Set zu aktualisieren

Der folgende `update-sql-injection-match-set` Befehl löscht ein `SqlInjectionMatchTuple` Objekt (Filter) in einem SQL Injection-Match-Set:

```
aws waf update-sql-injection-match-set --sql-injection-  
match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --  
change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates  
Action="DELETE",SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE"
```

Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch unter Working with SQL Injection Match Conditions.

- API Einzelheiten finden Sie [UpdateSqlInjectionMatchSet](#) in der AWS CLI Befehlsreferenz.

update-web-acl

Das folgende Codebeispiel zeigt die Verwendung `update-web-acl`.

AWS CLI

Um ein Web zu aktualisieren ACL

Der folgende `update-web-acl` Befehl löscht ein `ActivatedRule` Objekt in einem WebACL.

```
aws waf update-web-acl -- web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token
12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates Action=",, =' {Priority=1, =" -1-Beispiel",
Action= {Type=" „}, Type=" „}, Type=" „} 'DELETEActivatedRuleRuleIdWAFRuleALLOWREGULAR
```

Ausgabe:

```
{
  "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Web ACLs](#) im AWS WAF AWS Firewall Manager und AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [UpdateWebAcl](#) unter AWS CLI Befehlsreferenz.

update-xss-match-set

Das folgende Codebeispiel zeigt die Verwendung `update-xss-match-set`.

AWS CLI

Um ein zu aktualisieren XSSMatchSet

Der folgende `update-xss-match-set` Befehl löscht ein `XssMatchTuple` Objekt (Filter) in einem `XssMatchSet`:

```
aws waf update-xss-match-set --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_D
```

Weitere Informationen finden Sie im Entwicklerhandbuch unter Working with Cross-Site Scripting Match Conditions. AWS WAF

- API-Einheiten finden Sie [UpdateXssMatchSet](#) in der AWS CLI Befehlsreferenz.

AWS WAF Classic Regional Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS WAF Classic Regional.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-web-acl

Das folgende Codebeispiel zeigt die Verwendung `associate-web-acl`.

AWS CLI

Um ein Web ACL mit einer Ressource zu verknüpfen

Der folgende `associate-web-acl` Befehl verknüpft ein durch den `web-acl-id` spezifiziertes Web ACL mit einer durch den `resource-arn` angegebenen Ressource. Die Ressource ARN kann sich entweder auf einen Application Load Balancer oder ein Gateway beziehen: API

```
aws waf-regional associate-web-acl \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Web ACLs](#) im AWS WAFEntwicklerhandbuch.

- APIEinzelheiten finden Sie [AssociateWebAcl](#) in der AWS CLI Befehlsreferenz.

put-logging-configuration

Das folgende Codebeispiel zeigt die Verwendung `put-logging-configuration`.

AWS CLI

Um eine Logging-Konfiguration für das Web ACL ARN mit dem angegebenen Kinesis Firehose-Stream zu erstellen ARN

Das folgende `put-logging-configuration` Beispiel zeigt die Logging-Konfiguration für WAF mit ALB/API Gateway in Region `us-east-1`

```
aws waf-regional put-logging-configuration \
  --logging-configuration ResourceArn=arn:aws:waf-regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[] \
  --region us-east-1
```

Ausgabe:

```
{
  "LoggingConfiguration": {
    "ResourceArn": "arn:aws:waf-regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3",
    "LogDestinationConfigs": [
      "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream"
    ]
  }
}
```

- API Einzelheiten finden Sie [PutLoggingConfiguration](#) in der AWS CLI Befehlsreferenz.

update-byte-match-set

Das folgende Codebeispiel zeigt die Verwendung `update-byte-match-set`.

AWS CLI

Um ein Byte-Match-Set zu aktualisieren

Der folgende `update-byte-match-set` Befehl löscht ein `ByteMatchTuple` Objekt (Filter) in einem `ByteMatchSet`. Da der `updates` Wert doppelte Anführungszeichen enthält, müssen Sie den Wert in einfache Anführungszeichen setzen.


```
aws waf-regional update-byte-match-set \
  --byte-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
  --updates
'Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="
```

Weitere Informationen finden Sie im AWS WAF Entwicklerhandbuch [unter Working with String Match Conditions](#).

- API Einzelheiten finden Sie [UpdateByteMatchSet](#) in der AWS CLI Befehlsreferenz.

update-ip-set

Das folgende Codebeispiel zeigt die Verwendung `update-ip-set`.

AWS CLI

Um einen IP-Satz zu aktualisieren

Der folgende `update-ip-set` Befehl aktualisiert eine IPSet mit einer IPv4 Adresse und löscht eine IPv6 Adresse. Rufen Sie den Wert für `ab`, `change-token` indem Sie den `get-change-token` Befehl ausführen. Da der Wert für Aktualisierungen eingebettete doppelte Anführungszeichen enthält, müssen Sie den Wert in einfache Anführungszeichen setzen.

```
aws waf update-ip-set \
  --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
  --updates
'Action="INSERT",IPSetDescriptor={Type="IPV4",Value="12.34.56.78/16"},Action="DELETE",IPSet
```

Alternativ können Sie eine JSON Datei verwenden, um die Eingabe anzugeben. Beispielsweise:

```
aws waf-regional update-ip-set \
  --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
  --updates file://change.json
```

Inhalt des `change.json`

```
[
  {
```

```

    "Action": "INSERT",
    "IPSetDescriptor":
    {
        "Type": "IPV4",
        "Value": "12.34.56.78/16"
    }
  },
  {
    "Action": "DELETE",
    "IPSetDescriptor":
    {
        "Type": "IPV6",
        "Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
    }
  }
]

```

Weitere Informationen finden Sie unter [Arbeiten mit IP-Übereinstimmungsbedingungen](#) im AWS WAFEntwicklerhandbuch.

- APIEinzelheiten finden Sie [UpdateIpSet](#) in der AWS CLI Befehlsreferenz.

update-rule

Das folgende Codebeispiel zeigt die Verwendung `update-rule`.

AWS CLI

Um eine Regel zu aktualisieren

Der folgende `update-rule` Befehl löscht ein Predicate Objekt in einer Regel. Da der `updates` Wert doppelte Anführungszeichen enthält, müssen Sie den gesamten Wert in einfache Anführungszeichen setzen.

```

aws waf-regional update-rule \
  --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
  --updates
  'Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}'

```

Weitere Informationen finden Sie unter [Arbeiten mit Regeln](#) im AWS WAFEntwicklerhandbuch.

- APIEinzelheiten finden Sie [UpdateRule](#) in der AWS CLI Befehlsreferenz.

update-size-constraint-set

Das folgende Codebeispiel zeigt die Verwendung `update-size-constraint-set`.

AWS CLI

Um einen Größenbeschränkungssatz zu aktualisieren

Der folgende `update-size-constraint-set` Befehl löscht ein `SizeConstraint` -Objekt (Filter) in einem Größenbeschränkungssatz. Da der `updates` Wert eingebettete doppelte Anführungszeichen enthält, müssen Sie den gesamten Wert in einfache Anführungszeichen setzen.

```
aws waf-regional update-size-constraint-set \
  --size-constraint-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
  --updates
  'Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE"}
```

Weitere Informationen finden Sie unter [Arbeiten mit Größenbeschränkungsbedingungen](#) im AWS WAF Entwicklerhandbuch.

- API Einzelheiten finden Sie [UpdateSizeConstraintSet](#) in der AWS CLI Befehlsreferenz.

update-sql-injection-match-set

Das folgende Codebeispiel zeigt die Verwendung `update-sql-injection-match-set`.

AWS CLI

Um ein SQL Injection Match Set zu aktualisieren

Der folgende `update-sql-injection-match-set` Befehl löscht ein `SqlInjectionMatchTuple` Objekt (Filter) in einem SQL Injection-Match-Set. Da der `updates` Wert eingebettete doppelte Anführungszeichen enthält, müssen Sie den gesamten Wert in einfache Anführungszeichen setzen. :

```
aws waf-regional update-sql-injection-match-set -- sql-injection-match-set id
a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-
c3a4567d89f1 --updates 'Action="DELETE",SqlInjectionMatchTupleFieldToMatch=QUERY_STRING,TextTransformation=URLDECODE'
```

Weitere Informationen finden Sie unter [Sie unter Working with Injection Match SQL Conditions im Entwicklerhandbuch](#).AWS WAF

- APIEinzelheiten finden Sie [UpdateSqlInjectionMatchSet](#) in der AWS CLI Befehlsreferenz.

update-web-acl

Das folgende Codebeispiel zeigt die Verwendung `update-web-acl`.

AWS CLI

Um ein Web zu aktualisieren ACL

Der folgende `update-web-acl` Befehl löscht ein `ActivatedRule` Objekt in einem WebACL. Da der `updates` Wert eingebettete doppelte Anführungszeichen enthält, müssen Sie den gesamten Wert in einfache Anführungszeichen setzen.

```
aws waf-regional update-web-acl \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
  --updates Action="DELETE",ActivatedRule='{Priority=1,RuleId="WAFRule-1-Example",Action={Type="ALLOW"},Type="ALLOW"}'
```

Weitere Informationen finden Sie unter [Arbeiten mit dem Web ACLs](#) im AWS WAFEntwicklerhandbuch.

- APIEinzelheiten finden Sie [UpdateWebAcl](#) in der AWS CLI Befehlsreferenz.

update-xss-match-set

Das folgende Codebeispiel zeigt die Verwendung `update-xss-match-set`.

AWS CLI

Um ein zu aktualisieren XSSMatchSet

Der folgende `update-xss-match-set` Befehl löscht ein `XssMatchTuple` Objekt (Filter) in einem `XssMatchSet`. Da der `updates` Wert eingebettete doppelte Anführungszeichen enthält, müssen Sie den gesamten Wert in einfache Anführungszeichen setzen.

```
aws waf-regional update-xss-match-set \  
  --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --updates '{XssMatchTuple={Filter={Type="SQL_INJECTION_MATCHING"},Type="ALLOW"},Type="ALLOW"}'
```

```
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
--updates  
'Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_
```

Weitere Informationen finden Sie im [AWS WAFEntwicklerhandbuch unter Working with Cross-Site Scripting Match Conditions](#).

- APIEinzelheiten finden Sie [UpdateXssMatchSet](#) in der AWS CLI Befehlsreferenz.

AWS WAFV2 Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren AWS WAFV2.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-web-acl

Das folgende Codebeispiel zeigt die Verwendung `associate-web-acl`.

AWS CLI

Um ein Web ACL mit einer regionalen AWS Ressource zu verknüpfen

Das folgende `associate-web-acl` Beispiel verknüpft das angegebene Web ACL mit einem Application Load Balancer.

```
aws wafv2 associate-web-acl \  
--web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  

```

```
--resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/waf-cli-alb/1ea17125f8b25a2a \  
--region us-west-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Zuordnen oder Aufheben der Zuordnung eines Webs zu ACL einer AWS Ressource](#) im Entwicklerhandbuch AWS WAF zu AWS Firewall Manager und AWS Shield Advanced.

- API-Einheiten finden Sie [AssociateWebAcl](#) in AWS CLI der Befehlsreferenz.

check-capacity

Das folgende Codebeispiel zeigt die Verwendung `check-capacity`.

AWS CLI

Um die Kapazität zu ermitteln, die nach einer Reihe von Regeln genutzt wird

Im Folgenden werden die Kapazitätsanforderungen für einen Regelsatz `check-capacity` abgerufen, der eine ratenbasierte Regelanweisung und eine AND Regelanweisung enthält, die verschachtelte Regeln enthält.

```
aws wafv2 check-capacity \  
  --scope REGIONAL \  
  --rules file://waf-rule-list.json \  
  --region us-west-2
```

Inhalt von `file://.json: waf-rule-list`

```
[  
  {  
    "Name": "basic-rule",  
    "Priority": 0,  
    "Statement": {  
      "AndStatement": {  
        "Statements": [  
          {  
            "ByteMatchStatement": {  
              "SearchString": "example.com",  
              "FieldToMatch": {
```

```
        "SingleHeader":{
            "Name":"host"
        }
    },
    "TextTransformations":[
        {
            "Priority":0,
            "Type":"LOWERCASE"
        }
    ],
    "PositionalConstraint":"EXACTLY"
}
},
{
    "GeoMatchStatement":{
        "CountryCodes":[
            "US",
            "IN"
        ]
    }
}
]
}
},
"Action":{
    "Allow":{
    }
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"basic-rule"
}
},
{
    "Name":"rate-rule",
    "Priority":1,
    "Statement":{
        "RateBasedStatement":{
            "Limit":1000,
            "AggregateKeyType":"IP"
        }
    }
},
},
```

```
    "Action":{
      "Block":{

      }
    },
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"rate-rule"
    }
  }
]
```

Ausgabe:

```
{
  "Capacity":15
}
```

Weitere Informationen finden Sie unter [AWS WAF Web ACL Capacity Units \(WCU\)](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API Einzelheiten finden Sie [CheckCapacity](#) in der AWS CLI Befehlsreferenz.

create-ip-set

Das folgende Codebeispiel zeigt die Verwendung `create-ip-set`.

AWS CLI

Um einen IP-Satz für die Verwendung in Ihren Web ACLs - und Regelgruppen zu erstellen

Der folgende `create-ip-set` Befehl erstellt einen IP-Satz mit einer einzigen Adressbereichsspezifikation.

```
aws wafv2 create-ip-set \
  --name testip \
  --scope REGIONAL \
  --ip-address-version IPV4 \
  --addresses 198.51.100.0/16
```

Ausgabe:


```
{
  "Summary":{
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description":"",
    "Name":"testip",
    "LockToken":"447e55ac-0000-0000-0000-86b67c17f8b5",
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}
```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [CreateIpSet](#) in der AWS CLI Befehlsreferenz.

create-regex-pattern-set

Das folgende Codebeispiel zeigt die Verwendung `create-regex-pattern-set`.

AWS CLI

Um einen Regex-Mustersatz für die Verwendung in Ihren Web ACLs - und Regelgruppen zu erstellen

Der folgende `create-regex-pattern-set` Befehl erstellt einen Regex-Mustersatz mit zwei angegebenen Regex-Mustern.

```
aws wafv2 create-regex-pattern-set \
  --name regexPatterSet01 \
  --scope REGIONAL \
  --description 'Test web-acl' \
  --regular-expression-list '["RegexString": "[0-9]*"/],{"RegexString": "[a-
z]*"/}]'
```

Ausgabe:

```
{
  "Summary":{
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description":"Test web-acl",
```

```

    "Name": "regexPatterSet01",
    "LockToken": "0bc01e21-03c9-4b98-9433-6229cbf1ef1c",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [CreateRegexPatternSet](#) in der AWS CLI Befehlsreferenz.

create-rule-group

Das folgende Codebeispiel zeigt die Verwendung `create-rule-group`.

AWS CLI

Um eine benutzerdefinierte Regelgruppe für die Verwendung in Ihrem Web zu erstellen ACLs

Mit dem folgenden `create-rule-group` Befehl wird eine benutzerdefinierte Regelgruppe für den regionalen Gebrauch erstellt. Die Regelanweisungen für die Gruppe werden in einer Datei im JSON -Format bereitgestellt.

```

aws wafv2 create-rule-group \
  --name "TestRuleGroup" \
  --scope REGIONAL \
  --capacity 250 \
  --rules file://waf-rule.json \
  --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestRuleGroupMet
\
  --region us-west-2

```

Inhalt der Datei: `//waf-rule.json`:

```

[
  {
    "Name": "basic-rule",
    "Priority": 0,
    "Statement": {
      "AndStatement": {
        "Statements": [
          {

```

```

        "ByteMatchStatement":{
            "SearchString":"example.com",
            "FieldToMatch":{
                "SingleHeader":{
                    "Name":"host"
                }
            },
            "TextTransformations":[
                {
                    "Priority":0,
                    "Type":"LOWERCASE"
                }
            ],
            "PositionalConstraint":"EXACTLY"
        },
        {
            "GeoMatchStatement":{
                "CountryCodes":[
                    "US",
                    "IN"
                ]
            }
        }
    ]
},
"Action":{
    "Allow":{

    }
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"basic-rule"
}
}
]

```

Ausgabe:

```
{
```

```

    "Summary":{
      "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Description":"","
      "Name":"TestRuleGroup",
      "LockToken":"7b3bcec2-374e-4c5a-b2b9-563bf47249f0",
      "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
  }
}

```

Weitere Informationen finden Sie unter [Managing Your Own Rule Groups](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [CreateRuleGroup](#) in der AWS CLI Befehlsreferenz.

create-web-acl

Das folgende Codebeispiel zeigt die Verwendung `create-web-acl`.

AWS CLI

Um ein Web zu erstellen ACL

Mit dem folgenden `create-web-acl` Befehl wird ein Web ACL für den regionalen Gebrauch erstellt. Die Regelanweisungen für das Web ACL werden in einer Datei im JSON -Format bereitgestellt.

```

aws wafv2 create-web-acl \
  --name TestWebAcl \
  --scope REGIONAL \
  --default-action Allow={} \
  --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestWebAclMetric
\
  --rules file://waf-rule.json \
  --region us-west-2

```

Inhalt der Datei: `//waf-rule.json`:

```

[
  {
    "Name":"basic-rule",
    "Priority":0,

```

```
"Statement":{
  "AndStatement":{
    "Statements":[
      {
        "ByteMatchStatement":{
          "SearchString":"example.com",
          "FieldToMatch":{
            "SingleHeader":{
              "Name":"host"
            }
          },
          "TextTransformations":[
            {
              "Priority":0,
              "Type":"LOWERCASE"
            }
          ],
          "PositionalConstraint":"EXACTLY"
        },
        {
          "GeoMatchStatement":{
            "CountryCodes":[
              "US",
              "IN"
            ]
          }
        }
      ]
    },
    "Action":{
      "Allow":{
      }
    },
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"basic-rule"
    }
  }
}
```

Ausgabe:

```
{
  "Summary": {
    "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description": "",
    "Name": "TestWebAcl",
    "LockToken": "2294b3a1-eb60-4aa0-a86f-a3ae04329de9",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}
```

Weitere Informationen finden Sie unter [Verwalten und Verwenden einer Web-Zugriffskontrollliste \(WebACL\)](#) im AWS WAF Entwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- API Einzelheiten finden Sie [CreateWebAcl](#) in der AWS CLI Befehlsreferenz.

delete-ip-set

Das folgende Codebeispiel zeigt die Verwendung `delete-ip-set`.

AWS CLI

Um einen IP-Satz zu löschen

Im Folgenden `delete-ip-set` wird der angegebene IP-Satz gelöscht. Dieser Aufruf erfordert eine ID, die Sie aus dem Aufruf erhalten können `list-ip-sets`, und ein Sperrtoken, das Sie aus den Aufrufen erhalten können, `list-ip-sets` und `get-ip-set`.

```
aws wafv2 delete-ip-set \
  --name test1 \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --lock-token 46851772-db6f-459d-9385-49428812e357
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API Einzelheiten finden Sie [DeleteIpSet](#) in der AWS CLI Befehlsreferenz.

delete-logging-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-logging-configuration`.

AWS CLI

Um die Protokollierung für ein Web zu deaktivieren ACL

Im Folgenden `delete-logging-configuration` wird jegliche Protokollierungskonfiguration aus dem angegebenen Web entfernt ACL.

```
aws wafv2 delete-logging-configuration \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Logging Web ACL Traffic Information](#) im Advanced Developer Guide AWS WAF, AWS Firewall Manager und AWS Shield.

- API Einzelheiten finden Sie [DeleteLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

delete-regex-pattern-set

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-regex-pattern-set`.

AWS CLI

Um einen Regex-Mustersatz zu löschen

Im Folgenden werden die Einstellungen für den angegebenen Regex-Mustersatz `delete-regex-pattern-set` aktualisiert. Dieser Aufruf erfordert eine ID, die Sie aus dem Anruf erhalten können `list-regex-pattern-sets`, und ein Sperrtoken, das Sie aus dem Anruf `list-regex-pattern-sets` oder dem Anruf erhalten können. `get-regex-pattern-set`

```
aws wafv2 delete-regex-pattern-set \  
  --name regexPatterSet01 \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token 0bc01e21-03c9-4b98-9433-6229cbf1ef1c
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DeleteRegexPatternSet](#) in der AWS CLI Befehlsreferenz.

delete-rule-group

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-rule-group`.

AWS CLI

Um eine benutzerdefinierte Regelgruppe zu löschen

Im Folgenden `delete-rule-group` wird die angegebene benutzerdefinierte Regelgruppe gelöscht. Dieser Aufruf erfordert eine ID, die Sie aus dem Aufruf erhalten können `list-rule-groups`, und ein Sperrtoken, das Sie aus dem Aufruf `list-rule-groups` oder dem Aufruf `get-rule-group` erhalten können.

```
aws wafv2 delete-rule-group \
  --name TestRuleGroup \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Managing Your Own Rule Groups](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [DeleteRuleGroup](#) in der AWS CLI Befehlsreferenz.

delete-web-acl

Das folgende Codebeispiel zeigt, wie man es benutzt `delete-web-acl`.

AWS CLI

Um ein Web zu löschen ACL

Im Folgenden `delete-web-acl` wird die angegebene Website ACL aus Ihrem Konto gelöscht. Eine Website ACL kann nur gelöscht werden, wenn sie keinen Ressourcen zugeordnet ist. Dieser Aufruf erfordert eine ID, die Sie aus dem Aufruf erhalten können `list-web-acls`, und ein

Sperrtoken, das Sie aus dem Anruf `list-web-acls` oder dem Anruf erhalten können `get-web-acl`.

```
aws wafv2 delete-web-acl \  
  --name test \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token ebab4ed2-155e-4c9a-9efb-e4c45665b1f5
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verwalten und Verwenden einer Web-Zugriffskontrollliste \(WebACL\)](#) im AWS WAF Entwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- API-Einheiten finden Sie [DeleteWebAcl](#) in der AWS CLI Befehlsreferenz.

describe-managed-rule-group

Das folgende Codebeispiel zeigt, wie man es benutzt `describe-managed-rule-group`.

AWS CLI

Um die Beschreibung für eine verwaltete Regelgruppe abzurufen

Im Folgenden wird die Beschreibung für eine AWS verwaltete Regelgruppe `describe-managed-rule-group` abgerufen.

```
aws wafv2 describe-managed-rule-group \  
  --vendor-name AWS \  
  --name AWSManagedRulesCommonRuleSet \  
  --scope REGIONAL
```

Ausgabe:

```
{  
  "Capacity": 700,  
  "Rules": [  
    {  
      "Name": "NoUserAgent_HEADER",  
      "Action": {  
        "Block": {}  
      }  
    }  
  ]  
}
```

```
    }
  },
  {
    "Name": "UserAgent_BadBots_HEADER",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_QUERYSTRING",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_Cookie_HEADER",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_COOKIE",
    "Action": {
      "Block": {}
    }
  },
},
```

```
{
  "Name": "EC2MetaDataSSRF_URIPATH",
  "Action": {
    "Block": {}
  }
},
{
  "Name": "EC2MetaDataSSRF_QUERYARGUMENTS",
  "Action": {
    "Block": {}
  }
},
{
  "Name": "GenericLFI_QUERYARGUMENTS",
  "Action": {
    "Block": {}
  }
},
{
  "Name": "GenericLFI_URIPATH",
  "Action": {
    "Block": {}
  }
},
{
  "Name": "GenericLFI_BODY",
  "Action": {
    "Block": {}
  }
},
{
  "Name": "RestrictedExtensions_URIPATH",
  "Action": {
    "Block": {}
  }
},
{
  "Name": "RestrictedExtensions_QUERYARGUMENTS",
  "Action": {
    "Block": {}
  }
},
{
```

```
    "Name": "GenericRFI_QUERYARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericRFI_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericRFI_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_COOKIE",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_QUERYARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_URI_PATH",
    "Action": {
      "Block": {}
    }
  }
]
}
```

Weitere Informationen finden Sie unter [Verwaltete Regelgruppen](#) im AWS WAFEntwicklerhandbuch von AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [DescribeManagedRuleGroup](#) in der AWS CLI Befehlsreferenz.

disassociate-web-acl

Das folgende Codebeispiel zeigt, wie man es benutzt `disassociate-web-acl`.

AWS CLI

Um ein Web ACL von einer regionalen AWS Ressource zu trennen

Im folgenden `disassociate-web-acl` Beispiel werden alle vorhandenen ACL Webzuordnungen aus dem angegebenen Application Load Balancer entfernt.

```
aws wafv2 disassociate-web-acl \  
  --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/waf-cli-alb/1ea17125f8b25a2a \  
  --region us-west-2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Zuordnen oder Aufheben der Zuordnung eines Webs zu ACL einer AWS Ressource](#) im Entwicklerhandbuch AWS WAF zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [DisassociateWebAcl](#) in AWS CLI der Befehlsreferenz.

get-ip-set

Das folgende Codebeispiel zeigt, wie man es benutzt `get-ip-set`.

AWS CLI

Um einen bestimmten IP-Satz abzurufen

Im Folgenden wird der IP-Satz mit dem angegebenen Namen, Bereich und ID `get-ip-set` abgerufen. Sie können die ID für einen IP-Satz mit den Befehlen `create-ip-set` und `list-ip-sets` abrufen.

```
aws wafv2 get-ip-set \  
  --name example-ip-set \  
  --region us-west-2 \  
  --id example-id
```

```
--name testip \  
--scope REGIONAL \  
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "IPSet":{  
    "Description": "",  
    "Name": "testip",  
    "IPAddressVersion": "IPV4",  
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Addresses": [  
      "192.0.2.0/16"  
    ]  
  },  
  "LockToken": "447e55ac-2396-4c6d-b9f9-86b67c17f8b5"  
}
```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [GetIpSet](#) in der AWS CLI Befehlsreferenz.

get-logging-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `get-logging-configuration`.

AWS CLI

Um die Protokollierungskonfigurationen für ein Web abzurufen ACL

Im Folgenden wird die Protokollierungskonfiguration für das angegebene Web `get-logging-configuration` ACL abgerufen.

```
aws wafv2 get-logging-configuration \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \  
  --region us-west-2
```

Ausgabe:

```
{
  "LoggingConfiguration":{
    "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "RedactedFields":[
      {
        "Method":{
        }
      }
    ],
    "LogDestinationConfigs":[
      "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-
custom-transformation"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Logging Web ACL Traffic Information](#) im Advanced Developer Guide AWS WAF, AWS Firewall Manager und AWS Shield.

- API-Einheiten finden Sie [GetLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

get-rate-based-statement-managed-keys

Das folgende Codebeispiel zeigt, wie man es benutzt `get-rate-based-statement-managed-keys`.

AWS CLI

Um eine Liste von IP-Adressen abzurufen, die durch eine ratenbasierte Regel blockiert werden

Im Folgenden werden die IP-Adressen `get-rate-based-statement-managed-keys` abgerufen, die derzeit durch eine ratenbasierte Regel blockiert sind, die für eine regionale Anwendung verwendet wird.

```
aws wafv2 get-rate-based-statement-managed-keys \
  --scope REGIONAL \
  --web-acl-name testwebacl2 \
  --web-acl-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
```

```
--rule-name ratebasedtest
```

Ausgabe:

```
{
  "ManagedKeysIPV4":{
    "IPAddressVersion":"IPV4",
    "Addresses":[
      "198.51.100.0/32"
    ]
  },
  "ManagedKeysIPV6":{
    "IPAddressVersion":"IPV6",
    "Addresses":[
    ]
  }
}
```

Weitere Informationen finden Sie unter [Rate-Based Rule Statement](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie unter [GetRateBasedStatementManagedKeys AWS CLI](#) Befehlsreferenz.

get-regex-pattern-set

Das folgende Codebeispiel zeigt, wie man es benutzt `get-regex-pattern-set`.

AWS CLI

Um einen bestimmten Regex-Mustersatz abzurufen

Im Folgenden wird der Regex-Mustersatz mit dem angegebenen Namen, Bereich, Region und ID `get-regex-pattern-set` abgerufen. Sie können die ID für einen Regex-Mustersatz mit den Befehlen `create-regex-pattern-set` `list-regex-pattern-sets`

```
aws wafv2 get-regex-pattern-set \
  --name regexPatterSet01 \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --region us-west-2
```


Ausgabe:

```
{
  "RegexPatternSet":{
    "Description":"Test web-acl",
    "RegularExpressionList":[
      {
        "RegexString":"/[0-9]*/"
      },
      {
        "RegexString":"/[a-z]*/"
      }
    ],
    "Name":"regexPatterSet01",
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "LockToken":"c8abf33f-b6fc-46ae-846e-42f994d57b29"
}
```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [GetRegexPatternSet](#) in der AWS CLI Befehlsreferenz.

get-rule-group

Das folgende Codebeispiel zeigt, wie man es benutzt `get-rule-group`.

AWS CLI

Um eine bestimmte benutzerdefinierte Regelgruppe abzurufen

Im Folgenden wird die benutzerdefinierte Regelgruppe mit dem angegebenen Namen, Bereich und ID `get-rule-group` abgerufen. Sie können die ID für eine Regelgruppe mit den Befehlen `create-rule-group` und `list-rule-groups` abrufen.

```
aws wafv2 get-rule-group \
  --name ff \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{
  "RuleGroup":{
    "Capacity":1,
    "Description":"",
    "Rules":[
      {
        "Priority":0,
        "Action":{
          "Block":{

          }
        },
        "VisibilityConfig":{
          "SampledRequestsEnabled":true,
          "CloudWatchMetricsEnabled":true,
          "MetricName":"jj"
        },
        "Name":"jj",
        "Statement":{
          "SizeConstraintStatement":{
            "ComparisonOperator":"LE",
            "TextTransformations":[
              {
                "Priority":0,
                "Type":"NONE"
              }
            ],
            "FieldToMatch":{
              "UriPath":{

              }
            },
            "Size":7
          }
        }
      }
    ],
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"ff"
    },
  },
}
```

```

    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/ff/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "ff"
  },
  "LockToken": "485458c9-1830-4234-af31-ec4d52ced1b3"
}

```

Weitere Informationen finden Sie unter [Managing Your Own Rule Groups](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [GetRuleGroup](#) in der AWS CLI Befehlsreferenz.

get-sampled-requests

Das folgende Codebeispiel zeigt, wie man es benutzt `get-sampled-requests`.

AWS CLI

Um ein Beispiel von Webanfragen für ein Web abzurufen ACL

Im Folgenden werden die Stichproben-Webanfragen für das angegebene WebACL, die angegebene Regelmetrik und den angegebenen Zeitrahmen `get-sampled-requests` abgerufen.

```

aws wafv2 get-sampled-requests \
  --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --rule-metric-name AWS-AWSManagedRulesSQLiRuleSet \
  --scope=REGIONAL \
  --time-window StartTime=2020-02-12T20:00Z,EndTime=2020-02-12T21:10Z \
  --max-items 100

```

Ausgabe:

```

{
  "TimeWindow": {
    "EndTime": 1581541800.0,
    "StartTime": 1581537600.0
  },
  "SampledRequests": [
    {
      "Action": "BLOCK",

```

```
"Timestamp": 1581541799.564,
"RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
"Request": {
  "Country": "US",
  "URI": "/",
  "Headers": [
    {
      "Name": "Host",
      "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
    },
    {
      "Name": "Content-Length",
      "Value": "7456"
    },
    {
      "Name": "User-Agent",
      "Value": "curl/7.53.1"
    },
    {
      "Name": "Accept",
      "Value": "/"
    },
    {
      "Name": "Content-Type",
      "Value": "application/x-www-form-urlencoded"
    }
  ],
  "ClientIP": "198.51.100.08",
  "Method": "POST",
  "HTTPVersion": "HTTP/1.1"
},
"Weight": 1
},
{
  "Action": "BLOCK",
  "Timestamp": 1581541799.988,
  "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
  "Request": {
    "Country": "US",
    "URI": "/",
    "Headers": [
      {
        "Name": "Host",
        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
```

```
    },
    {
      "Name": "Content-Length",
      "Value": "7456"
    },
    {
      "Name": "User-Agent",
      "Value": "curl/7.53.1"
    },
    {
      "Name": "Accept",
      "Value": "/"
    },
    {
      "Name": "Content-Type",
      "Value": "application/x-www-form-urlencoded"
    }
  ],
  "ClientIP": "198.51.100.08",
  "Method": "POST",
  "HTTPVersion": "HTTP/1.1"
},
"Weight": 3
},
{
  "Action": "BLOCK",
  "Timestamp": 1581541799.846,
  "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
  "Request": {
    "Country": "US",
    "URI": "/",
    "Headers": [
      {
        "Name": "Host",
        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
      },
      {
        "Name": "Content-Length",
        "Value": "7456"
      },
      {
        "Name": "User-Agent",
        "Value": "curl/7.53.1"
      }
    ]
  }
},
```

```
        {
            "Name": "Accept",
            "Value": "/"
        },
        {
            "Name": "Content-Type",
            "Value": "application/x-www-form-urlencoded"
        }
    ],
    "ClientIP": "198.51.100.08",
    "Method": "POST",
    "HTTPVersion": "HTTP/1.1"
},
"Weight": 1
},
{
    "Action": "BLOCK",
    "Timestamp": 1581541799.4,
    "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
    "Request": {
        "Country": "US",
        "URI": "/",
        "Headers": [
            {
                "Name": "Host",
                "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
            },
            {
                "Name": "Content-Length",
                "Value": "7456"
            },
            {
                "Name": "User-Agent",
                "Value": "curl/7.53.1"
            },
            {
                "Name": "Accept",
                "Value": "/"
            },
            {
                "Name": "Content-Type",
                "Value": "application/x-www-form-urlencoded"
            }
        ]
    }
},
],
```

```

        "ClientIP": "198.51.100.08",
        "Method": "POST",
        "HTTPVersion": "HTTP/1.1"
    },
    "Weight": 1
}
],
"PopulationSize": 4
}

```

Weitere Informationen finden Sie unter [Ein Beispiel für Webanfragen anzeigen](#) im AWS WAF, AWS Firewall Manager und AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [GetSampledRequests](#) in der AWS CLI Befehlsreferenz.

get-web-acl-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `get-web-acl-for-resource`.

AWS CLI

Um das Web abzurufen ACL, das einer AWS Ressource zugeordnet ist

Im Folgenden wird das JSON für das Web `get-web-acl-for-resource` abgerufen ACL, das der angegebenen Ressource zugeordnet ist.

```

aws wafv2 get-web-acl-for-resource \
  --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a

```

Ausgabe:

```

{
  "WebACL": {
    "Capacity": 3,
    "Description": "",
    "Rules": [
      {
        "Priority": 1,
        "Action": {
          "Block": {

```

```
    },
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"testrule01"
    },
    "Name":"testrule01",
    "Statement":{
      "AndStatement":{
        "Statements":[
          {
            "ByteMatchStatement":{
              "PositionalConstraint":"EXACTLY",
              "TextTransformations":[
                {
                  "Priority":0,
                  "Type":"NONE"
                }
              ],
              "SearchString":"dGVzdHN0cm1uZw==",
              "FieldToMatch":{
                "UriPath":{
                }
              }
            }
          },
          {
            "SizeConstraintStatement":{
              "ComparisonOperator":"EQ",
              "TextTransformations":[
                {
                  "Priority":0,
                  "Type":"NONE"
                }
              ],
              "FieldToMatch":{
                "QueryString":{
                }
              }
            }
          }
        ],
        "Size":0
      }
    }
  }
}
```



```

    ],
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"test01"
    },
    "DefaultAction":{
      "Allow":{

      }
    },
    "Id":"9a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
    "Name":"test01"
  }
}

```

Weitere Informationen finden Sie unter [Zuordnen oder Aufheben der Zuordnung eines Webs zu ACL einer AWS Ressource](#) im Entwicklerhandbuch AWS WAF zu AWS Firewall Manager und AWS Shield Advanced.

- API-Einheiten finden Sie [GetWebAclForResource](#) in AWS CLI der Befehlsreferenz.

get-web-acl

Das folgende Codebeispiel zeigt, wie man es benutzt `get-web-acl`.

AWS CLI

Um ein Web abzurufen ACL

Im Folgenden wird das Web ACL mit dem angegebenen Namen, Bereich und ID `get-web-acl` abgerufen. Sie können die ID für ein Web mit den Befehlen `create-web-acl` und `ACL list-web-acls` abrufen.

```

aws wafv2 get-web-acl \
  --name test01 \

```

```
--scope REGIONAL \  
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Ausgabe:

```
{  
  "WebACL":{  
    "Capacity":3,  
    "Description":"","  
    "Rules":[  
      {  
        "Priority":1,  
        "Action":{  
          "Block":{  
  
          }  
        },  
        "VisibilityConfig":{  
          "SampledRequestsEnabled":true,  
          "CloudWatchMetricsEnabled":true,  
          "MetricName":"testrule01"  
        },  
        "Name":"testrule01",  
        "Statement":{  
          "AndStatement":{  
            "Statements":[  
              {  
                "ByteMatchStatement":{  
                  "PositionalConstraint":"EXACTLY",  
                  "TextTransformations":[  
                    {  
                      "Priority":0,  
                      "Type":"NONE"  
                    }  
                  ],  
                  "SearchString":"dGVzdHN0cm1uZw==",  
                  "FieldToMatch":{  
                    "UriPath":{  
  
                    }  
                  }  
                }  
              }  
            ]  
          }  
        }  
      },  
    ]  
  }  
}
```

```

        {
            "SizeConstraintStatement":{
                "ComparisonOperator":"EQ",
                "TextTransformations":[
                    {
                        "Priority":0,
                        "Type":"NONE"
                    }
                ],
                "FieldToMatch":{
                    "QueryString":{

                    }
                },
                "Size":0
            }
        }
    ]
}
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"test01"
},
"DefaultAction":{
    "Allow":{

    }
},
"Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"Name":"test01"
},
"LockToken":"e3db7e2c-d58b-4ee6-8346-6aec5511c6fb"
}

```

Weitere Informationen finden Sie unter [Verwalten und Verwenden einer Web-Zugriffskontrollliste \(WebACL\)](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- API-Einheiten finden Sie [GetWebACL](#) in der AWS CLI Befehlsreferenz.

list-available-managed-rule-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `list-available-managed-rule-groups`.

AWS CLI

Um die verwalteten Regelgruppen abzurufen

Im Folgenden `list-available-managed-rule-groups` finden Sie eine Liste aller verwalteten Regelgruppen, die derzeit für die Verwendung in Ihrer Website verfügbar sind ACLs.

```
aws wafv2 list-available-managed-rule-groups \
  --scope REGIONAL
```

Ausgabe:

```
{
  "ManagedRuleGroups": [
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesCommonRuleSet",
      "Description": "Contains rules that are generally applicable to web applications. This provides protection against exploitation of a wide range of vulnerabilities, including those described in OWASP publications and common Common Vulnerabilities and Exposures (CVE).",
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesAdminProtectionRuleSet",
      "Description": "Contains rules that allow you to block external access to exposed admin pages. This may be useful if you are running third-party software or would like to reduce the risk of a malicious actor gaining administrative access to your application."
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesKnownBadInputsRuleSet",
      "Description": "Contains rules that allow you to block request patterns that are known to be invalid and are associated with exploitation or discovery of vulnerabilities. This can help reduce the risk of a malicious actor discovering a vulnerable application."
    }
  ]
}
```

```
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesSQLiRuleSet",
      "Description": "Contains rules that allow you to block request patterns
associated with exploitation of SQL databases, like SQL injection attacks. This can
help prevent remote injection of unauthorized queries."
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesLinuxRuleSet",
      "Description": "Contains rules that block request patterns associated
with exploitation of vulnerabilities specific to Linux, including LFI attacks. This
can help prevent attacks that expose file contents or execute code for which the
attacker should not have had access."
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesUnixRuleSet",
      "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to POSIX/POSIX-like OS, including LFI
attacks. This can help prevent attacks that expose file contents or execute code
for which access should not been allowed."
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesWindowsRuleSet",
      "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to Windows, (e.g., PowerShell commands).
This can help prevent exploits that allow attacker to run unauthorized commands or
execute malicious code."
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesPHPRuleSet",
      "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to the use of the PHP, including injection
of unsafe PHP functions. This can help prevent exploits that allow an attacker to
remotely execute code or commands."
    },
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesWordPressRuleSet",
```

```
    "Description": "The WordPress Applications group contains rules that
block request patterns associated with the exploitation of vulnerabilities specific
to WordPress sites."
  },
  {
    "VendorName": "AWS",
    "Name": "AWSManagedRulesAmazonIpReputationList",
    "Description": "This group contains rules that are based on Amazon
threat intelligence. This is useful if you would like to block sources associated
with bots or other threats."
  }
]
}
```

Weitere Informationen finden Sie unter [Verwaltete Regelgruppen](#) im AWS WAFEntwicklerhandbuch von AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [ListAvailableManagedRuleGroups](#) in der AWS CLI Befehlsreferenz.

list-ip-sets

Das folgende Codebeispiel zeigt, wie man es benutzt `list-ip-sets`.

AWS CLI

Um eine Liste von IP-Sätzen abzurufen

Im Folgenden werden alle IP-Sets für das Konto `list-ip-sets` abgerufen, die einen regionalen Geltungsbereich haben.

```
aws wafv2 list-ip-sets \
  --scope REGIONAL
```

Ausgabe:

```
{
  "IPSets": [
    {
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Description": "",
```

```

        "Name": "testip",
        "LockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc",
        "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  "
    }
],
"NextMarker": "testip"
}

```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [ListIpSets](#) in der AWS CLI Befehlsreferenz.

list-logging-configurations

Das folgende Codebeispiel zeigt, wie man es benutzt `list-logging-configurations`.

AWS CLI

Um eine Liste aller Protokollierungskonfigurationen für eine Region abzurufen

Im Folgenden werden alle Protokollierungskonfigurationen für das Web `list-logging-configurations` abgerufen. ACLs, die für die regionale Verwendung in der Region vorgesehen sind. `us-west-2`

```

aws wafv2 list-logging-configurations \
  --scope REGIONAL \
  --region us-west-2

```

Ausgabe:

```

{
  "LoggingConfigurations": [
    {
      "ResourceArn": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "RedactedFields": [
        {
          "QueryString": {

```

```

    }
  ],
  "LogDestinationConfigs": [
    "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-test"
  ],
  },
  {
    "ResourceArn": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "RedactedFields": [
      {
        "Method": {
          }
        }
      ],
    "LogDestinationConfigs": [
      "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-custom-transformation"
    ]
  }
]
}

```

Weitere Informationen finden Sie unter [Logging Web ACL Traffic Information](#) im Advanced Developer Guide AWS WAF, AWS Firewall Manager und AWS Shield.

- API-Einheiten finden Sie [ListLoggingConfigurations](#) unter AWS CLI Befehlsreferenz.

list-regex-pattern-sets

Das folgende Codebeispiel zeigt, wie man es benutzt `list-regex-pattern-sets`.

AWS CLI

Um eine Liste von Regex-Mustersätzen abzurufen

Im Folgenden werden alle Regex-Mustersätze für das Konto `list-regex-pattern-sets` abgerufen, die in der Region definiert sind. `us-west-2`

```
aws wafv2 list-regex-pattern-sets \
```



```
--scope REGIONAL \
--region us-west-2
```

Ausgabe:

```
{
  "NextMarker": "regexPatterSet01",
  "RegexPatternSets": [
    {
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Description": "Test web-acl",
      "Name": "regexPatterSet01",
      "LockToken": "f17743f7-0000-0000-0000-19a8b93bfb01",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
  ]
}
```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [ListRegexPatternSets](#) in der AWS CLI Befehlsreferenz.

list-resources-for-web-acl

Das folgende Codebeispiel zeigt, wie man es benutzt `list-resources-for-web-acl`.

AWS CLI

Um die mit einem Web verknüpften Ressourcen abzurufen ACL

Im Folgenden werden die API REST API Gateway-Ressourcen `list-resources-for-web-acl` abgerufen, die derzeit dem angegebenen Web ACL in der Region `us-west-2` zugeordnet sind.

```
aws wafv2 list-resources-for-web-acl \
  --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --resource-type API_GATEWAY \
  --region us-west-2
```

Ausgabe:

```
{
  "ResourceArns": [
    "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
  ]
}
```

Weitere Informationen finden Sie unter [Zuordnen oder Aufheben der Zuordnung eines Webs zu ACL einer AWS Ressource](#) im Entwicklerhandbuch AWS WAF zu AWS Firewall Manager und AWS Shield Advanced.

- API-Einheiten finden Sie [ListResourcesForWebAcl](#) in AWS CLI der Befehlsreferenz.

list-rule-groups

Das folgende Codebeispiel zeigt, wie man es benutzt `list-rule-groups`.

AWS CLI

Um eine Liste von benutzerdefinierten Regelgruppen abzurufen

Im Folgenden werden alle benutzerdefinierten Regelgruppen `list-rule-groups` abgerufen, die für das Konto für den angegebenen Bereich und die angegebene Region definiert sind.

```
aws wafv2 list-rule-groups \
  --scope REGIONAL \
  --region us-west-2
```

Ausgabe:

```
{
  "RuleGroups": [
    {
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Description": "",
      "Name": "TestRuleGroup",
      "LockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
  ],
}
```

```

    {
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "Description": "",
      "Name": "test",
      "LockToken": "b0f4583e-998b-4880-9069-3fbe45738b43",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    }
  ],
  "NextMarker": "test"
}

```

Weitere Informationen finden Sie unter [Managing Your Own Rule Groups](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [ListRuleGroups](#) in der AWS CLI Befehlsreferenz.

list-tags-for-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `list-tags-for-resource`.

AWS CLI

Um alle Tags für eine AWS WAF Ressource abzurufen

Im Folgenden wird die Liste aller Tag-Schlüssel-Wert-Paare für das angegebene Web `list-tags-for-resource` ACL abgerufen.

```

aws wafv2 list-tags-for-resource \
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl2/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Ausgabe:

```

{
  "NextMarker": "",
  "TagInfoForResource": {
    "ResourceARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
testwebacl2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "TagList": [
      ]
    }
  ]
}

```

```
}  
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit AWS WAF](#) im AWS WAF AWS Firewall Manager und AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [ListTagsForResource](#) unter AWS CLI Befehlsreferenz.

list-web-acls

Das folgende Codebeispiel zeigt, wie man es benutzt `list-web-acls`.

AWS CLI

Um das Web ACLs für einen Bereich abzurufen

Im Folgenden werden alle Websites `list-web-acls` abgerufen ACLs, die für das Konto für den angegebenen Bereich definiert sind.

```
aws wafv2 list-web-acls \  
  --scope REGIONAL
```

Ausgabe:

```
{  
  "NextMarker": "Testt",  
  "WebACLs": [  
    {  
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/Testt/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Description": "sssss",  
      "Name": "Testt",  
      "LockToken": "7f36cb30-74ef-4cff-8cd4-a77e1aba1746",  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Verwalten und Verwenden einer Web-Zugriffskontrollliste \(WebACL\)](#) im AWS WAF-Entwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- API-Einheiten finden Sie [ListWebAcls](#) in der AWS CLI Befehlsreferenz.

put-logging-configuration

Das folgende Codebeispiel zeigt, wie man es benutzt `put-logging-configuration`.

AWS CLI

Um eine Logging-Konfiguration zu einem Web hinzuzufügen ACL

Im Folgenden wird die Amazon Kinesis Data Firehose Firehose-Protokollierungskonfiguration `aws-waf-logs-custom-transformation` zum angegebenen Web `put-logging-configuration` hinzugefügt ACL, ohne dass Felder aus den Protokollen gelöscht werden.

```
aws wafv2 put-logging-configuration \
  --logging-configuration ResourceArn=arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111,LogDestinationConfigs=arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-custom-transformation \
  --region us-west-2
```

Ausgabe:

```
{
  "LoggingConfiguration": {
    "ResourceArn": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "LogDestinationConfigs": [
      "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-custom-transformation"
    ]
  }
}
```

Weitere Informationen finden Sie unter [Logging Web ACL Traffic Information](#) im Advanced Developer Guide AWS WAF, AWS Firewall Manager und AWS Shield.

- API-Einheiten finden Sie [PutLoggingConfiguration](#) unter AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `tag-resource`.

AWS CLI

Um einer AWS WAF Ressource Tags hinzuzufügen

Im folgenden `tag-resource` Beispiel wird dem angegebenen Web ein Tag hinzugefügt, dessen Schlüssel Name und Wert AWSWAF auf gesetzt sindACL.

```
aws wafv2 tag-resource \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tags Key=Name, Value=AWSWAF
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit AWS WAF](#) im AWS WAF AWS Firewall Manager und AWS Shield Advanced Developer Guide.

- APIEinzelheiten finden Sie [TagResource](#)unter AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt, wie man es benutzt `untag-resource`.

AWS CLI

Um Tags aus einer AWS WAF Ressource zu entfernen

Im folgenden `untag-resource` Beispiel wird das Tag mit dem Schlüssel KeyName aus dem angegebenen Web entferntACL.

```
aws wafv2 untag-resource \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tag-keys "KeyName"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Erste Schritte mit AWS WAF](#) im AWS WAF AWS Firewall Manager und AWS Shield Advanced Developer Guide.

- APIEinzelheiten finden Sie [UntagResource](#)unter AWS CLI Befehlsreferenz.

update-ip-set

Das folgende Codebeispiel zeigt, wie man es benutzt `update-ip-set`.

AWS CLI

Um die Einstellungen für einen vorhandenen IP-Satz zu ändern

Im Folgenden werden die Einstellungen für den angegebenen IP-Satz `update-ip-set` aktualisiert. Dieser Aufruf erfordert eine ID, die Sie aus dem Aufruf erhalten können `list-ip-sets`, und ein Sperrtoken, das Sie aus den Aufrufen erhalten können, `list-ip-sets` und `get-ip-set`. Dieser Aufruf gibt auch ein Sperrtoken zurück, das Sie für ein späteres Update verwenden können.

```
aws wafv2 update-ip-set \  
  --name testip \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --addresses 198.51.100.0/16 \  
  --lock-token 447e55ac-2396-4c6d-b9f9-86b67c17f8b5
```

Ausgabe:

```
{  
  "NextLockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc"  
}
```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [UpdateIpSet](#) in der AWS CLI Befehlsreferenz.

update-regex-pattern-set

Das folgende Codebeispiel zeigt, wie man es benutzt `update-regex-pattern-set`.

AWS CLI

Um die Einstellungen für einen vorhandenen Regex-Mustersatz zu ändern

Im Folgenden werden die Einstellungen für den angegebenen Regex-Mustersatz `update-regex-pattern-set` aktualisiert. Dieser Aufruf erfordert eine ID, die Sie aus dem Aufruf

erhalten können, und ein Sperrtoken `list-regex-pattern-sets`, das Sie aus den Aufrufen erhalten können, `list-regex-pattern-sets` und `get-regex-pattern-set`. Dieser Aufruf gibt auch ein Sperrtoken zurück, das Sie für ein späteres Update verwenden können.

```
aws wafv2 update-regex-pattern-set \  
  --name ExampleRegex \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --regular-expression-list RegexString="^.+ $" \  
  --lock-token ed207e9c-82e9-4a77-aadd-81e6173ab7eb
```

Ausgabe:

```
{  
  "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"  
}
```

Weitere Informationen finden Sie unter [IP-Sets und Regex-Pattern-Sets](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [UpdateRegexPatternSet](#) in der AWS CLI Befehlsreferenz.

update-rule-group

Das folgende Codebeispiel zeigt, wie man es benutzt `update-rule-group`.

AWS CLI

Um eine benutzerdefinierte Regelgruppe zu aktualisieren

Im Folgenden wird die Sichtbarkeitskonfiguration für eine bestehende benutzerdefinierte Regelgruppe `update-rule-group` geändert. Dieser Aufruf erfordert eine ID, die Sie aus dem Aufruf erhalten können `list-rule-groups`, und ein Sperrtoken, das Sie aus den Aufrufen erhalten können, `list-rule-groups` und `get-rule-group`. Dieser Aufruf gibt auch ein Sperrtoken zurück, das Sie für ein späteres Update verwenden können.

```
aws wafv2 update-rule-group \  
  --name TestRuleGroup \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token ed207e9c-82e9-4a77-aadd-81e6173ab7eb
```



```

--lock-token 7b3bcec2-0000-0000-0000-563bf47249f0 \
--visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=TestMetricsFor
\
--region us-west-2

```

Ausgabe:

```

{
  "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"
}

```

Weitere Informationen finden Sie unter [Managing Your Own Rule Groups](#) im AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- API-Einheiten finden Sie [UpdateRuleGroup](#) in der AWS CLI Befehlsreferenz.

update-web-acl

Das folgende Codebeispiel zeigt, wie man es benutzt `update-web-acl`.

AWS CLI

Um ein Web zu aktualisieren ACL

Im Folgenden werden die Einstellungen für ein vorhandenes Web `update-web-acl` geändert ACL. Für diesen Aufruf sind eine ID erforderlich, die Sie aus dem Aufruf abrufen können `list-web-acls`, sowie ein Sperrtoken und andere Einstellungen, die Sie aus dem Aufruf abrufen können `get-web-acl`. Dieser Aufruf gibt auch ein Sperrtoken zurück, das Sie für ein späteres Update verwenden können.

```

aws wafv2 update-web-acl \
  --name TestWebAcl \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --lock-token 2294b3a1-0000-0000-0000-a3ae04329de9 \
  --default-action Block={} \
  --visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=NewMetricTestW
\
  --rules file://waf-rule.json \

```

```
--region us-west-2
```

Ausgabe:

```
{  
  "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"  
}
```

Weitere Informationen finden Sie unter [Verwalten und Verwenden einer Web-Zugriffskontrollliste \(WebACL\)](#) im AWS WAFEntwicklerhandbuch zu AWS Firewall Manager und AWS Shield Advanced.

- APIEinzelheiten finden Sie [UpdateWebAcl](#)in der AWS CLI Befehlsreferenz.

WorkDocs Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren WorkDocs.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

abort-document-version-upload

Das folgende Codebeispiel zeigt die Verwendung `abort-document-version-upload`.

AWS CLI

Um den Upload einer Dokumentversion zu beenden

In diesem Beispiel wird der Upload einer zuvor initiierten Dokumentversion gestoppt.

Befehl:

```
aws workdocs abort-document-version-upload --document-id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2 --version-id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dccc417da9313
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [AbortDocumentVersionUpload](#) in der AWS CLI Befehlsreferenz.

activate-user

Das folgende Codebeispiel zeigt die Verwendung `activate-user`.

AWS CLI

Um einen Benutzer zu aktivieren

In diesem Beispiel wird ein inaktiver Benutzer aktiviert.

Befehl:

```
aws workdocs activate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Ausgabe:

```
{
  "User": {
    "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Username": "exampleUser",
    "EmailAddress": "exampleUser@site.awsapps.com",
    "GivenName": "Example",
    "Surname": "User",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
      "75f67c183aa1217409ac87576a45c03a5df5e6d8c51c35c01669970538e86cd0",
    "RecycleBinFolderId":
      "642b7dd3e60b14204534f3df7b1959e01b5d170f8c2707f410e40a8149120a57",
```

```

    "Status": "ACTIVE",
    "Type": "MINIMALUSER",
    "CreatedTimestamp": 1521226107.747,
    "ModifiedTimestamp": 1525297406.462,
    "Storage": {
      "StorageUtilizedInBytes": 0,
      "StorageRule": {
        "StorageAllocatedInBytes": 0,
        "StorageType": "QUOTA"
      }
    }
  }
}

```

- API-Einheiten finden Sie [ActivateUser](#) in der AWS CLI Befehlsreferenz.

add-resource-permissions

Das folgende Codebeispiel zeigt die Verwendung `add-resource-permissions`.

AWS CLI

Um Berechtigungen für eine Ressource hinzuzufügen

In diesem Beispiel werden der Ressource Berechtigungen für die angegebenen Prinzipale hinzugefügt.

Befehl:

```

aws workdocs add-resource-permissions --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
principals Id=anonymous, Type=ANONYMOUS, Role=VIEWER

```

Ausgabe:

```

{
  "ShareResults": [
    {
      "PrincipalId": "anonymous",
      "Role": "VIEWER",
      "Status": "SUCCESS",
    }
  ]
}

```

```

    "ShareId":
      "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "StatusMessage": ""
    }
  ]
}

```

- API-Einheiten finden Sie unter [AddResourcePermissions AWS CLI](#) Befehlsreferenz.

create-comment

Das folgende Codebeispiel zeigt die Verwendung `create-comment`.

AWS CLI

Um einen neuen Kommentar hinzuzufügen

In diesem Beispiel wird der angegebenen Dokumentversion ein neuer Kommentar hinzugefügt.

Befehl:

```

aws workdocs create-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
text "This is a comment."

```

Ausgabe:

```

{
  "Comment": {
    "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
    "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
    "Text": "This is a comment.",
    "Contributor": {
      "Id": "arn:aws:iam::123456789123:user/exampleUser",
      "Username": "exampleUser",
      "GivenName": "Example",
      "Surname": "User",
      "Status": "ACTIVE"
    }
  },

```

```
"CreatedTimestamp": 1534799058.197,  
"Status": "PUBLISHED",  
"Visibility": "PUBLIC"  
}  
}
```

- API-Einheiten finden Sie [CreateComment](#) unter AWS CLI Befehlsreferenz.

create-custom-metadata

Das folgende Codebeispiel zeigt die Verwendung `create-custom-metadata`.

AWS CLI

Um benutzerdefinierte Metadaten zu erstellen

In diesem Beispiel werden benutzerdefinierte Metadaten für das angegebene Dokument erstellt.

Befehl:

```
aws workdocs create-custom-metadata --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-  
metadata KeyName1=example,KeyName2=example2
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [CreateCustomMetadata](#) unter AWS CLI Befehlsreferenz.

create-folder

Das folgende Codebeispiel zeigt die Verwendung `create-folder`.

AWS CLI

Um einen Ordner zu erstellen

In diesem Beispiel wird ein Ordner erstellt.

Befehl:

```
aws workdocs create-folder --name documents --parent-folder-  
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Ausgabe:

```
{  
  "Metadata": {  
    "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",  
    "Name": "documents",  
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",  
    "ParentFolderId":  
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",  
    "CreatedTimestamp": 1534450467.622,  
    "ModifiedTimestamp": 1534450467.622,  
    "ResourceState": "ACTIVE",  
    "Signature": "",  
    "Size": 0,  
    "LatestVersionSize": 0  
  }  
}
```

- API-Einheiten finden Sie [CreateFolder](#) unter AWS CLI Befehlsreferenz.

create-labels

Das folgende Codebeispiel zeigt die Verwendung `create-labels`.

AWS CLI

Um Labels zu erstellen

In diesem Beispiel wird eine Reihe von Etiketten für ein Dokument erstellt.

Befehl:

```
aws workdocs create-labels --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --  
labels "documents" "examples" "my_documents"
```

Ausgabe:

None

- API-Einheiten finden Sie [CreateLabels](#) unter AWS CLI Befehlsreferenz.

create-notification-subscription

Das folgende Codebeispiel zeigt die Verwendung `create-notification-subscription`.

AWS CLI

Um ein Benachrichtigungsabonnement zu erstellen

Im folgenden `create-notification-subscription` Beispiel wird ein Benachrichtigungsabonnement für die angegebene WorkDocs Amazon-Organisation konfiguriert.

```
aws workdocs create-notification-subscription \  
  --organization-id d-123456789c \  
  --protocol HTTPS \  
  --subscription-type ALL \  
  --notification-endpoint "https://example.com/example"
```

Ausgabe:

```
{  
  "Subscription": {  
    "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",  
    "EndPoint": "https://example.com/example",  
    "Protocol": "HTTPS"  
  }  
}
```

Weitere Informationen finden Sie unter [Benachrichtigungen abonnieren](#) im Amazon WorkDocs Developer Guide.

- API-Einheiten finden Sie [CreateNotificationSubscription](#) in der AWS CLI Befehlsreferenz.

create-user

Das folgende Codebeispiel zeigt die Verwendung `create-user`.

AWS CLI

Um einen neuen Benutzer zu erstellen

In diesem Beispiel wird ein neuer Benutzer in einem Simple AD- oder Microsoft AD-Verzeichnis erstellt.

Befehl:

```
aws workdocs create-user --organization-id d-926726012c --username exampleUser2
--email-address exampleUser2@site.awsapps.com --given-name example2Name --
surname example2Surname --password examplePa$$w0rd
```

Ausgabe:

```
{
  "User": {
    "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Username": "exampleUser2",
    "EmailAddress": "exampleUser2@site.awsapps.com",
    "GivenName": "example2Name",
    "Surname": "example2Surname",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
"35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
    "RecycleBinFolderId":
"9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
    "Status": "ACTIVE",
    "Type": "MINIMALUSER",
    "CreatedTimestamp": 1535478836.584,
    "ModifiedTimestamp": 1535478836.584,
    "Storage": {
      "StorageUtilizedInBytes": 0,
      "StorageRule": {
        "StorageAllocatedInBytes": 0,
        "StorageType": "QUOTA"
      }
    }
  }
}
```

- API-Einheiten finden Sie [CreateUser](#) in der AWS CLI Befehlsreferenz.

deactivate-user

Das folgende Codebeispiel zeigt die Verwendung `deactivate-user`.

AWS CLI

Um einen Benutzer zu deaktivieren

In diesem Beispiel wird ein aktiver Benutzer deaktiviert.

Befehl:

```
aws workdocs deactivate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [DeactivateUser](#) in der AWS CLI Befehlsreferenz.

delete-comment

Das folgende Codebeispiel zeigt die Verwendung `delete-comment`.

AWS CLI

Um einen bestimmten Kommentar aus einer Dokumentversion zu löschen

In diesem Beispiel wird der angegebene Kommentar aus der angegebenen Dokumentversion gelöscht.

Befehl:

```
aws workdocs delete-comment --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --comment-id 1534799058197-c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

Ausgabe:

None

- API-Einheiten finden Sie unter [DeleteComment AWS CLI](#) Befehlsreferenz.

delete-custom-metadata

Das folgende Codebeispiel zeigt die Verwendung `delete-custom-metadata`.

AWS CLI

Um benutzerdefinierte Metadaten aus einer Ressource zu löschen

In diesem Beispiel werden alle benutzerdefinierten Metadaten aus der angegebenen Ressource gelöscht.

Befehl:

```
aws workdocs delete-custom-metadata --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

Ausgabe:

None

- API-Einheiten finden Sie [DeleteCustomMetadata](#) in der AWS CLI Befehlsreferenz.

delete-document

Das folgende Codebeispiel zeigt die Verwendung `delete-document`.

AWS CLI

Um ein Dokument zu löschen

In diesem Beispiel wird das angegebene Dokument gelöscht.

Befehl:

```
aws workdocs delete-document --document-  
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

Ausgabe:

None

- API-Einheiten finden Sie unter [DeleteDocument AWS CLI](#) Befehlsreferenz.

delete-folder-contents

Das folgende Codebeispiel zeigt die Verwendung `delete-folder-contents`.

AWS CLI

Um den Inhalt eines Ordners zu löschen

In diesem Beispiel wird der Inhalt des angegebenen Ordners gelöscht.

Befehl:

```
aws workdocs delete-folder-contents --folder-id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Ausgabe:

None

- API-Einheiten finden Sie unter [DeleteFolderContents AWS CLI](#) Befehlsreferenz.

delete-folder

Das folgende Codebeispiel zeigt die Verwendung `delete-folder`.

AWS CLI

Um einen Ordner zu löschen

In diesem Beispiel wird der angegebene Ordner gelöscht.

Befehl:

```
aws workdocs delete-folder --folder-id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Ausgabe:

None

- API-Einheiten finden Sie unter [DeleteFolder AWS CLI](#) Befehlsreferenz.

delete-labels

Das folgende Codebeispiel zeigt die Verwendung `delete-labels`.

AWS CLI

Um Beschriftungen zu löschen

In diesem Beispiel werden die angegebenen Beschriftungen aus einem Dokument gelöscht.

Befehl:

```
aws workdocs delete-labels --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --labels "documents" "examples"
```

Ausgabe:

None

- API-Einheiten finden Sie unter [DeleteLabels AWS CLI](#) Befehlsreferenz.

delete-notification-subscription

Das folgende Codebeispiel zeigt die Verwendung `delete-notification-subscription`.

AWS CLI

Um ein Benachrichtigungsabonnement zu löschen

Im folgenden `delete-notification-subscription` Beispiel wird das angegebene Benachrichtigungsabonnement gelöscht.

```
aws workdocs delete-notification-subscription \
  --subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \
```

```
--organization-id d-123456789c
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden [Sie unter Benachrichtigungen abonnieren](#) im Amazon WorkDocs Developer Guide.

- APIEinzelheiten finden Sie [DeleteNotificationSubscription](#) in der AWS CLI Befehlsreferenz.

delete-user

Das folgende Codebeispiel zeigt die Verwendung `delete-user`.

AWS CLI

Benutzer löschen

In diesem Beispiel wird ein Benutzer gelöscht.

Befehl:

```
aws workdocs delete-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Ausgabe:

```
None
```

- APIEinzelheiten finden Sie [DeleteUser](#) in der AWS CLI Befehlsreferenz.

describe-activities

Das folgende Codebeispiel zeigt die Verwendung `describe-activities`.

AWS CLI

Um eine Liste der Benutzeraktivitäten zu erhalten

In diesem Beispiel wird eine Liste der letzten Benutzeraktivitäten für die angegebene Organisation zurückgegeben, wobei für die letzten beiden Aktivitäten ein Limit festgelegt wurde.

Befehl:

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

Ausgabe:

```
{
  "UserActivities": [
    {
      "Type": "DOCUMENT_VERSION_DOWNLOADED",
      "TimeStamp": 1534800122.17,
      "Initiator": {
        "Id": "arn:aws:iam::123456789123:user/exampleUser"
      },
      "ResourceMetadata": {
        "Type": "document",
        "Name": "updatedDoc",
        "Id":
"15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
        "Owner": {
          "Id":
"S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "GivenName": "exampleName",
          "Surname": "exampleSurname"
        }
      }
    },
    {
      "Type": "DOCUMENT_VERSION_VIEWED",
      "TimeStamp": 1534799079.207,
      "Initiator": {
        "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
        "GivenName": "exampleName",
        "Surname": "exampleSurname"
      },
      "ResourceMetadata": {
        "Type": "document",
        "Name": "updatedDoc",
        "Id":
"15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
        "Owner": {
          "Id":
"S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "GivenName": "exampleName",
          "Surname": "exampleSurname"
        }
      }
    }
  ]
}
```

```

    }
  }
},
"Marker":
"DnF1ZXJ5VGhlbkZldGNoAgAAAAAAS7Fm1TaU10d1FTU1h1UU00VVFibD1RWHcAAAAAAAJTRY3bWh5eUgzaVF1ZX
}

```

- API-Einheiten finden Sie [DescribeActivities](#) in der AWS CLI Befehlsreferenz.

describe-comments

Das folgende Codebeispiel zeigt die Verwendung von `describe-comments`.

AWS CLI

Um alle Kommentare für eine bestimmte Dokumentversion aufzulisten

In diesem Beispiel werden alle Kommentare für die angegebene Dokumentversion aufgelistet.

Befehl:

```

aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920

```

Ausgabe:

```

{
  "Comments": [
    {
      "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "Text": "This is a comment.",
      "Contributor": {
        "Username": "arn:aws:iam::123456789123:user/exampleUser",
        "Type": "USER"
      },
      "CreatedTimestamp": 1534799058.197,
      "Status": "PUBLISHED",
      "Visibility": "PUBLIC"
    }
  ]
}

```



```
    }  
  ]  
}
```

- API-Einheiten finden Sie [DescribeComments](#) unter AWS CLI Befehlsreferenz.

describe-document-versions

Das folgende Codebeispiel zeigt die Verwendung `describe-document-versions`.

AWS CLI

Um die Versionen eines Dokuments abzurufen

In diesem Beispiel werden die Dokumentversionen für das angegebene Dokument abgerufen, einschließlich der initialisierten Versionen und der Versionen URL für das Quelldokument.

Befehl:

```
aws workdocs describe-document-versions --document-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

Ausgabe:

```
{  
  "DocumentVersions": [  
    {  
      "Id":  
      "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",  
      "Name": "exampleDoc.docx",  
      "ContentType": "application/vnd.openxmlformats-  
officedocument.wordprocessingml.document",  
      "Size": 13922,  
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",  
      "Status": "ACTIVE",  
      "CreatedTimestamp": 1534452029.587,  
      "ModifiedTimestamp": 1534452029.849,  
      "CreatorId":  
      "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",  
      "Source": {  
        "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-  
west-2.amazonaws.com/  
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef"      }  
    }  
  ]  
}
```

```

response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.docx&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k23456781901234mno56pqr78EXAMPLE"
    }
  },
  {
    "Id": "1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59",
    "Name": "exampleDoc.pdf",
    "ContentType": "application/pdf",
    "Size": 425916,
    "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
    "Status": "ACTIVE",
    "CreatedTimestamp": 1529005196.082,
    "ModifiedTimestamp": 1529005196.796,
    "CreatorId":
    "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Source": {
      "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59?
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.pdf&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k23456781901234mno56pqr78EXAMPLE"
    }
  }
]
}

```

- API-Einheiten finden Sie unter [DescribeDocumentVersions AWS CLI](#) Befehlsreferenz.

describe-folder-contents

Das folgende Codebeispiel zeigt die Verwendung `describe-folder-contents`.

AWS CLI

Um den Inhalt eines Ordners zu beschreiben

Dieses Beispiel beschreibt den gesamten aktiven Inhalt des angegebenen Ordners, einschließlich seiner Dokumente und Unterordner, sortiert nach Datum in aufsteigender Reihenfolge.

Befehl:

```
aws workdocs describe-folder-contents --folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --order ASCENDING --type ALL
```

Ausgabe:

```
{
  "Folders": [
    {
      "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
      "Name": "testing",
      "CreatorId":
      "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
      "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1534450467.622,
      "ModifiedTimestamp": 1534451113.504,
      "ResourceState": "ACTIVE",
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Size": 23019,
      "LatestVersionSize": 11537
    }
  ],
  "Documents": [
    {
      "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "CreatorId":
      "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
      "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1529005196.082,
      "ModifiedTimestamp": 1534452483.01,
      "LatestVersionMetadata": {
        "Id":
        "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
        "Name": "exampleDoc.docx",
        "ContentType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
```

```

        "Size": 13922,
        "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
        "Status": "ACTIVE",
        "CreatedTimestamp": 1534452029.587,
        "ModifiedTimestamp": 1534452029.587,
        "CreatorId":
    "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
    },
    "ResourceState": "ACTIVE"
}
]
}

```

- API-Einheiten finden Sie [DescribeFolderContents](#) in der AWS CLI Befehlsreferenz.

describe-groups

Das folgende Codebeispiel zeigt die Verwendung `describe-groups`.

AWS CLI

Um eine Liste von Gruppen abzurufen

Das folgende `describe-groups` Beispiel listet die Gruppen auf, die der angegebenen WorkDocs Amazon-Organisation zugeordnet sind.

```

aws workdocs describe-groups \
  --search-query "e" \
  --organization-id d-123456789c

```

Ausgabe:

```

{
  "Groups": [
    {
      "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444&d-123456789c",
      "Name": "Example Group 1"
    },
    {
      "Id": "S-1-1-11-1122222222-2222233333-3333334444-5555&d-123456789c",
      "Name": "Example Group 2"
    }
  ]
}

```

```
]
}
```

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon WorkDocs im WorkDocs Amazon-Administratorhandbuch](#).

- API-Einheiten finden Sie [DescribeGroups](#) in der AWS CLI Befehlsreferenz.

describe-notification-subscriptions

Das folgende Codebeispiel zeigt die Verwendung `describe-notification-subscriptions`.

AWS CLI

Um eine Liste von Benachrichtigungsabonnements abzurufen

Im folgenden `describe-notification-subscriptions` Beispiel werden die Benachrichtigungsabonnements für die angegebene WorkDocs Amazon-Organisation abgerufen.

```
aws workdocs describe-notification-subscriptions \
  --organization-id d-123456789c
```

Ausgabe:

```
{
  "Subscriptions": [
    {
      "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
      "EndPoint": "https://example.com/example",
      "Protocol": "HTTPS"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Benachrichtigungen abonnieren](#) im Amazon WorkDocs Developer Guide.

- API-Einheiten finden Sie [DescribeNotificationSubscriptions](#) in der AWS CLI Befehlsreferenz.

describe-resource-permissions

Das folgende Codebeispiel zeigt die Verwendung `describe-resource-permissions`.

AWS CLI

Um eine Liste der Berechtigungen für eine Ressource abzurufen

Dieses Beispiel gibt eine Liste der Berechtigungen für die angegebene Ressource (Dokument oder Ordner) zurück.

Befehl:

```
aws workdocs describe-resource-permissions --resource-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

Ausgabe:

```
{
  "Principals": [
    {
      "Id": "anonymous",
      "Type": "ANONYMOUS",
      "Roles": [
        {
          "Role": "VIEWER",
          "Type": "DIRECT"
        }
      ]
    },
    {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Type": "USER",
      "Roles": [
        {
          "Role": "OWNER",
          "Type": "DIRECT"
        }
      ]
    },
    {
      "Id": "d-926726012c",
      "Type": "ORGANIZATION",
      "Roles": [
        {
          "Role": "VIEWER",
          "Type": "INHERITED"
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

- API-Einheiten finden Sie [DescribeResourcePermissions](#) unter AWS CLI Befehlsreferenz.

describe-users

Das folgende Codebeispiel zeigt die Verwendung `describe-users`.

AWS CLI

Um Details für bestimmte Benutzer abzurufen

In diesem Beispiel werden Details für alle Benutzer in der angegebenen Organisation abgerufen.

Befehl:

```
aws workdocs describe-users --organization-id d-926726012c
```

Ausgabe:

```

{
  "Users": [
    {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "example1User",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
"3c0e3f849dd20a9771d937b9bbcc97e18796150ae56c26d64a4fa0320a2dedc9",
      "RecycleBinFolderId":
"c277f4c4d647be1f5147b3184ffa96e1e2bf708278b696cacba68ba13b91f4fe",
      "Status": "INACTIVE",
      "Type": "USER",
      "CreatedTimestamp": 1535478999.452,
      "ModifiedTimestamp": 1535478999.452
    },
    {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-4444&d-926726012c",
      "Username": "example2User",

```

```

    "EmailAddress": "example2User@site.awsapps.com",
    "GivenName": "example2Name",
    "Surname": "example2Surname",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
"35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
    "RecycleBinFolderId":
"9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
    "Status": "ACTIVE",
    "Type": "MINIMALUSER",
    "CreatedTimestamp": 1535478836.584,
    "ModifiedTimestamp": 1535478836.584
  }
]
}

```

- API-Einheiten finden Sie unter [DescribeUsers AWS CLI-Befehlsreferenz](#).

get-document-path

Das folgende Codebeispiel zeigt die Verwendung von `get-document-path`.

AWS CLI

Um die Pfadinformationen eines Dokuments abzurufen

In diesem Beispiel werden die Pfadinformationen (Hierarchie aus dem Stammordner) für das angegebene Dokument abgerufen und die Namen der übergeordneten Ordner eingeschlossen.

Befehl:

```
aws workdocs get-document-path --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME
```

Ausgabe:

```

{
  "Path": {
    "Components": [
      {
        "Id":
"a43d29cbb8e7c4d25cfce8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",

```



```

        "Name": "/"
      },
      {
        "Id":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
"d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
        "Name": "exampleDoc.docx"
      }
    ]
  }
}

```

- API-Einheiten finden Sie unter [GetDocumentPath AWS CLI Befehlsreferenz](#).

get-document-version

Das folgende Codebeispiel zeigt die Verwendung `get-document-version`.

AWS CLI

Um Versionsmetadaten für ein bestimmtes Dokument abzurufen

In diesem Beispiel werden Versionsmetadaten für das angegebene Dokument abgerufen, einschließlich einer Quelle URL und benutzerdefinierter Metadaten.

Befehl:

```

aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata

```

Ausgabe:

```

{
  "Metadata": {
    "Id":
"1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920",

```

```

    "Name": "exampleDoc",
    "ContentType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
    "Size": 11537,
    "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
    "Status": "ACTIVE",
    "CreatedTimestamp": 1521672507.741,
    "ModifiedTimestamp": 1534451113.504,
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Source": {
      "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-west-2.amazonaws.com/15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3/1521672507.741-response-content-disposition=attachment%3B%20filename%2A%3DUTF-8%27%27exampleDoc&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-Date=20180820T212202Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20180820%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-Signature=01Ab2c34d567e8f90123g456hi78j901k23456781901234mno56pqr78EXAMPLE"
    }
  }
}

```

- API-Einheiten finden Sie unter [GetDocumentVersion AWS CLI Befehlsreferenz](#).

get-document

Das folgende Codebeispiel zeigt die Verwendung get-document.

AWS CLI

Um Dokumentdetails abzurufen

In diesem Beispiel werden die Details des angegebenen Dokuments abgerufen.

Befehl:

```
aws workdocs get-document --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

Ausgabe:

```
{
  "Metadata": {
```

```

    "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "ParentFolderId":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
    "CreatedTimestamp": 1529005196.082,
    "ModifiedTimestamp": 1534452483.01,
    "LatestVersionMetadata": {
        "Id":
"1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
        "Name": "exampleDoc.docx",
        "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
        "Size": 13922,
        "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
        "Status": "ACTIVE",
        "CreatedTimestamp": 1534452029.587,
        "ModifiedTimestamp": 1534452029.587,
        "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
    },
    "ResourceState": "ACTIVE"
}
}

```

- API-Einheiten finden Sie unter [GetDocument AWS CLI-Befehlsreferenz](#).

get-folder-path

Das folgende Codebeispiel zeigt die Verwendung `get-folder-path`.

AWS CLI

Um Pfadinformationen für einen Ordner abzurufen

In diesem Beispiel werden die Pfadinformationen (Hierarchie aus dem Stammordner) für den angegebenen Ordner abgerufen und die Namen der übergeordneten Ordner eingeschlossen.

Befehl:

```
aws workdocs get-folder-path --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

Ausgabe:

```
{
  "Path": {
    "Components": [
      {
        "Id":
"a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
        "Name": "/"
      },
      {
        "Id":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
"50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
        "Name": "Sublevel Folder"
      }
    ]
  }
}
```

- API-Einheiten finden Sie unter [GetFolderPath AWS CLI-Befehlsreferenz](#).

get-folder

Das folgende Codebeispiel zeigt die Verwendung von `get-folder`.

AWS CLI

Um die Metadaten für einen Ordner abzurufen

In diesem Beispiel werden die Metadaten für den angegebenen Ordner abgerufen.

Befehl:

```
aws workdocs get-folder --folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Ausgabe:

```
{
```

```

"Metadata": {
  "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
  "Name": "exampleFolder",
  "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
  "ParentFolderId":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
  "CreatedTimestamp": 1534450467.622,
  "ModifiedTimestamp": 1534451113.504,
  "ResourceState": "ACTIVE",
  "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
  "Size": 23019,
  "LatestVersionSize": 11537
}
}

```

- API-Einheiten finden Sie unter [GetFolder AWS CLI Befehlsreferenz](#).

get-resources

Das folgende Codebeispiel zeigt die Verwendung `get-resources`.

AWS CLI

Um gemeinsam genutzte Ressourcen abzurufen

Im folgenden `get-resources` Beispiel werden die Ressourcen abgerufen, die mit dem angegebenen WorkDocs Amazon-Benutzer geteilt wurden.

```

aws workdocs get-resources \
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \
  --collection-type SHARED_WITH_ME

```

Ausgabe:

```

{
  "Folders": [],
  "Documents": []
}

```

Weitere Informationen finden Sie unter [Dateien und Ordner teilen](#) im WorkDocs Amazon-Benutzerhandbuch.

- API-Einheiten finden Sie [GetResources](#) in der AWS CLI Befehlsreferenz.

initiate-document-version-upload

Das folgende Codebeispiel zeigt die Verwendung `initiate-document-version-upload`.

AWS CLI

Um den Upload einer Dokumentversion zu initiieren

Im folgenden `initiate-document-upload` Beispiel werden ein neues Dokumentobjekt und ein neues Versionsobjekt erstellt.

```
aws workdocs initiate-document-version-upload \  
  --name exampledocname \  
  --parent-folder-  
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

Ausgabe:

```
{  
  "Metadata": {  
    "Id": "feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2",  
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",  
    "ParentFolderId":  
"eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189",  
    "CreatedTimestamp": 1536773972.914,  
    "ModifiedTimestamp": 1536773972.914,  
    "LatestVersionMetadata": {  
      "Id": "1536773972914-  
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dccccd417da9313",  
      "Name": "exampledocname",  
      "ContentType": "application/octet-stream",  
      "Size": 0,  
      "Status": "INITIALIZED",  
      "CreatedTimestamp": 1536773972.914,  
      "ModifiedTimestamp": 1536773972.914,  
      "CreatorId": "arn:aws:iam::123456789123:user/EXAMPLE"  
    },  
    "ResourceState": "ACTIVE"  
  },  
  "UploadMetadata": {
```

```

    "UploadUrl": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2/1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dccccd417da9313?X-Amz-
Algorithm=AWS1-ABCD-EFG234&X-Amz-Date=20180912T173932Z&X-Amz-SignedHeaders=content-
type%3Bhost%3Bx-amz-server-side-encryption&X-Amz-Expires=899&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180912%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE",
    "SignedHeaders": {
        "Content-Type": "application/octet-stream",
        "x-amz-server-side-encryption": "ABC123"
    }
}
}

```

- API-Einheiten finden Sie [InitiateDocumentVersionUpload](#) unter AWS CLI Befehlsreferenz.

remove-all-resource-permissions

Das folgende Codebeispiel zeigt die Verwendung `remove-all-resource-permissions`.

AWS CLI

Um alle Berechtigungen von einer angegebenen Ressource zu entfernen

In diesem Beispiel werden alle Berechtigungen von der angegebenen Ressource entfernt.

Befehl:

```
aws workdocs remove-all-resource-permissions --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [RemoveAllResourcePermissions](#) in der AWS CLI Befehlsreferenz.

remove-resource-permission

Das folgende Codebeispiel zeigt die Verwendung `remove-resource-permission`.

AWS CLI

Um Berechtigungen von einer Ressource zu entfernen

In diesem Beispiel werden Berechtigungen für den angegebenen Prinzipal aus der Ressource entfernt.

Befehl:

```
aws workdocs remove-resource-permission --resource-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-id anonymous
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [RemoveResourcePermission](#) unter AWS CLI Befehlsreferenz.

update-document-version

Das folgende Codebeispiel zeigt die Verwendung `update-document-version`.

AWS CLI

Um den Versionsstatus eines Dokuments in Aktiv zu ändern

In diesem Beispiel wird der Status der Dokumentversion in Aktiv geändert.

Befehl:

```
aws workdocs update-document-version --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --version-status ACTIVE
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [UpdateDocumentVersion](#) unter AWS CLI Befehlsreferenz.

update-document

Das folgende Codebeispiel zeigt die Verwendung `update-document`.

AWS CLI

Um ein Dokument zu aktualisieren

In diesem Beispiel werden der Name und der übergeordnete Ordner eines Dokuments aktualisiert.

Befehl:

```
aws workdocs update-document --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --name updatedDoc --parent-folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Ausgabe:

```
None
```

- API-Einheiten finden Sie [UpdateDocument](#) unter AWS CLI Befehlsreferenz.

update-folder

Das folgende Codebeispiel zeigt die Verwendung `update-folder`.

AWS CLI

Um einen Ordner zu aktualisieren

In diesem Beispiel werden der Name und der übergeordnete Ordner eines Ordners aktualisiert.

Befehl:

```
aws workdocs update-folder --folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --name exampleFolder1 --parent-folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Ausgabe:

None

- API-Einheiten finden Sie [UpdateFolder](#) unter AWS CLI Befehlsreferenz.

update-user

Das folgende Codebeispiel zeigt die Verwendung `update-user`.

AWS CLI

Um einen Benutzer zu aktualisieren

In diesem Beispiel wird die Zeitzone für den angegebenen Benutzer aktualisiert.

Befehl:

```
aws workdocs update-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-id "America/Los_Angeles"
```

Ausgabe:

```
{
  "User": {
    "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Username": "exampleUser",
    "EmailAddress": "exampleUser@site.awsapps.com",
    "GivenName": "Example",
    "Surname": "User",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
    "c5eceb5e1a2d1d460c9d1af8330ae117fc8d39bb1d3ed6acd0992d5ff192d986",
    "RecycleBinFolderId":
    "6ca20102926ad15f04b1d248d6d6e44f2449944eda5c758f9a1e9df6a6b7fa66",
    "Status": "ACTIVE",
    "Type": "USER",
    "TimeZoneId": "America/Los_Angeles",
    "Storage": {
      "StorageUtilizedInBytes": 0,
      "StorageRule": {
        "StorageAllocatedInBytes": 53687091200,
        "StorageType": "QUOTA"
      }
    }
  }
}
```

```
}  
  }  
} }
```

- API-Einheiten finden Sie [UpdateUser](#) in der AWS CLI Befehlsreferenz.

WorkMail Amazon-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie AWS Command Line Interface mit Amazon Aktionen ausführen und allgemeine Szenarien implementieren WorkMail.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

associate-delegate-to-resource

Das folgende Codebeispiel zeigt die Verwendung `associate-delegate-to-resource`.

AWS CLI

Um einer Ressource einen Delegierten hinzuzufügen

Der folgende `associate-delegate-to-resource` Befehl fügt einer Ressource einen Delegierten hinzu.

```
aws workmail associate-delegate-to-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AssociateDelegateToResource](#) in der AWS CLI Befehlsreferenz.

associate-member-to-group

Das folgende Codebeispiel zeigt die Verwendung `associate-member-to-group`.

AWS CLI

Um ein Mitglied zu einer Gruppe hinzuzufügen

Der folgende `associate-member-to-group` Befehl fügt das angegebene Mitglied einer Gruppe hinzu.

```
aws workmail associate-member-to-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [AssociateMemberToGroup](#) unter AWS CLI Befehlsreferenz.

create-alias

Das folgende Codebeispiel zeigt die Verwendung `create-alias`.

AWS CLI

Um einen Alias zu erstellen

Der folgende `create-alias` Befehl erstellt einen Alias für die angegebene Entität (Benutzer oder Gruppe).

```
aws workmail create-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [CreateAlias](#) unter AWS CLI Befehlsreferenz.

create-group

Das folgende Codebeispiel zeigt die Verwendung `create-group`.

AWS CLI

Um eine neue Gruppe zu erstellen

Der folgende `create-group` Befehl erstellt eine neue Gruppe für die angegebene Organisation.

```
aws workmail create-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleGroup1
```

Ausgabe:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"  
}
```

- API-Einheiten finden Sie [CreateGroup](#) unter AWS CLI Befehlsreferenz.

create-resource

Das folgende Codebeispiel zeigt die Verwendung `create-resource`.

AWS CLI

Um eine neue Ressource zu erstellen

Der folgende `create-resource` Befehl erstellt eine neue Ressource (Besprechungsraum) für die angegebene Organisation.

```
aws workmail create-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleRoom1 \  
  --type ROOM
```

Ausgabe:

```
{
```

```
"ResourceId": "r-7afe0efbade843a58cdc10251fce992c"
}
```

- API-Einheiten finden Sie [CreateResource](#) unter AWS CLI Befehlsreferenz.

create-user

Das folgende Codebeispiel zeigt die Verwendung `create-user`.

AWS CLI

Um einen neuen Benutzer zu erstellen

Der folgende `create-user` Befehl erstellt einen neuen Benutzer.

```
aws workmail create-user \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --name exampleName \
  --display-name exampleDisplayName \
  --password examplePa$$w0rd
```

Ausgabe:

```
{
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"
}
```

- API-Einheiten finden Sie [CreateUser](#) in der AWS CLI Befehlsreferenz.

delete-access-control-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-access-control-rule`.

AWS CLI

Um eine Zugriffskontrollregel zu löschen

Im folgenden `delete-access-control-rule` Beispiel wird die angegebene Zugriffskontrollregel aus der angegebenen WorkMail Amazon-Organisation gelöscht.

```
aws workmail delete-access-control-rule \
```

```
--organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
--name "myRule"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Zugriffskontrollregeln](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteAccessControlRule](#) in der AWS CLI Befehlsreferenz.

delete-alias

Das folgende Codebeispiel zeigt die Verwendung `delete-alias`.

AWS CLI

Um einen Alias zu löschen

Der folgende `delete-alias` Befehl löscht den Alias für die angegebene Entität (Benutzer oder Gruppe).

```
aws workmail delete-alias \  
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
--entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
--alias exampleAlias@site.awsapps.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteAlias AWS CLI](#) Befehlsreferenz.

delete-group

Das folgende Codebeispiel zeigt die Verwendung `delete-group`.

AWS CLI

Um eine bestehende Gruppe zu löschen

Der folgende `delete-group` Befehl löscht eine bestehende Gruppe aus Amazon WorkMail.

```
aws workmail delete-group \  
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  

```

```
--group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteGroup](#) in der AWS CLI Befehlsreferenz.

delete-mailbox-permissions

Das folgende Codebeispiel zeigt die Verwendung `delete-mailbox-permissions`.

AWS CLI

Um Postfachberechtigungen zu löschen

Mit dem folgenden `delete-mailbox-permissions` Befehl werden Postfachberechtigungen gelöscht, die zuvor einem Benutzer oder einer Gruppe gewährt wurden. Die Entität stellt den Benutzer dar, dem das Postfach gehört, und der Empfänger steht für den Benutzer oder die Gruppe, für den bzw. die Berechtigungen gelöscht werden sollen.

```
aws workmail delete-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --grantee-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeleteMailboxPermissions AWS CLI](#) Befehlsreferenz.

delete-resource

Das folgende Codebeispiel zeigt die Verwendung `delete-resource`.

AWS CLI

Um eine bestehende Ressource zu löschen

Der folgende `delete-resource` Befehl löscht eine vorhandene Ressource aus Amazon WorkMail.

```
aws workmail delete-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id S-1-1-11-1122222222-2222233333-3333334444-4444
```



```
--resource-id r-7afe0efbade843a58cdc10251fce992c
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteResource](#) in der AWS CLI Befehlsreferenz.

delete-user

Das folgende Codebeispiel zeigt die Verwendung `delete-user`.

AWS CLI

Benutzer löschen

Der folgende `delete-user` Befehl löscht den angegebenen Benutzer aus Amazon WorkMail und allen nachfolgenden Systemen.

```
aws workmail delete-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DeleteUser](#) in der AWS CLI Befehlsreferenz.

deregister-from-work-mail

Das folgende Codebeispiel zeigt die Verwendung `deregister-from-work-mail`.

AWS CLI

Um eine bestehende Entität zu deaktivieren

Der folgende `deregister-from-work-mail` Befehl verhindert, dass eine bestehende Entität (Benutzer, Gruppe oder Ressource) Amazon WorkMail verwendet.

```
aws workmail deregister-from-work-mail \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie unter [DeregisterFromWorkMail AWS CLI](#) Befehlsreferenz.

describe-group

Das folgende Codebeispiel zeigt die Verwendung `describe-group`.

AWS CLI

Um Informationen für eine Gruppe abzurufen

Mit dem folgenden `describe-group` Befehl werden Informationen über die angegebene Gruppe abgerufen.

```
aws workmail describe-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Ausgabe:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",  
  "Name": "exampleGroup1",  
  "State": "ENABLED"  
}
```

- API-Einheiten finden Sie unter [DescribeGroup AWS CLI](#) Befehlsreferenz.

describe-organization

Das folgende Codebeispiel zeigt die Verwendung `describe-organization`.

AWS CLI

Um Informationen für eine Organisation abzurufen

Der folgende `describe-organization` Befehl ruft Informationen für die angegebene WorkMail Amazon-Organisation ab.

```
aws workmail describe-organization \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Ausgabe:

```
{
  "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
  "Alias": "alias",
  "State": "Active",
  "DirectoryId": "d-926726012c",
  "DirectoryType": "VpcDirectory",
  "DefaultMailDomain": "site.awsapps.com",
  "CompletedDate": 1522693605.468,
  "ARN": "arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza"
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Organizations](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeOrganization](#) in der AWS CLI Befehlsreferenz.

describe-resource

Das folgende Codebeispiel zeigt die Verwendung `describe-resource`.

AWS CLI

Um Informationen für eine Ressource abzurufen

Der folgende `describe-resource` Befehl ruft Informationen über die angegebene Ressource ab.

```
aws workmail describe-resource \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --resource-id r-7afe0efbade843a58cdc10251fce992c
```

Ausgabe:

```
{
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
  "Name": "exampleRoom1",
  "Type": "ROOM",
  "BookingOptions": {
    "AutoAcceptRequests": true,
  }
}
```

```
    "AutoDeclineRecurringRequests": false,  
    "AutoDeclineConflictingRequests": true  
  },  
  "State": "ENABLED"  
}
```

- API-Einheiten finden Sie unter [DescribeResource AWS CLI](#) Befehlsreferenz.

describe-user

Das folgende Codebeispiel zeigt die Verwendung `describe-user`.

AWS CLI

Um Benutzerinformationen abzurufen

Mit dem folgenden `describe-user` Befehl werden Informationen über den angegebenen Benutzer abgerufen.

```
aws workmail describe-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Ausgabe:

```
{  
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
  "Name": "exampleUser1",  
  "Email": "exampleUser1@site.awsapps.com",  
  "DisplayName": "",  
  "State": "ENABLED",  
  "UserRole": "USER",  
  "EnabledDate": 1532459261.827  
}
```

- API-Einheiten finden Sie unter [DescribeUser AWS CLI](#) Befehlsreferenz.

disassociate-delegate-from-resource

Das folgende Codebeispiel zeigt die Verwendung `disassociate-delegate-from-resource`.

AWS CLI

Um ein Mitglied aus einer Ressource zu entfernen

Der folgende `disassociate-delegate-from-resource` Befehl entfernt das angegebene Mitglied aus einer Ressource.

```
aws workmail disassociate-delegate-from-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DisassociateDelegateFromResource](#) unter AWS CLI Befehlsreferenz.

disassociate-member-from-group

Das folgende Codebeispiel zeigt die Verwendung `disassociate-member-from-group`.

AWS CLI

Um ein Mitglied aus einer Gruppe zu entfernen

Der folgende `disassociate-member-from-group` Befehl entfernt das angegebene Mitglied aus einer Gruppe.

```
aws workmail disassociate-member-from-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [DisassociateMemberFromGroup](#) unter AWS CLI Befehlsreferenz.

get-access-control-effect

Das folgende Codebeispiel zeigt die Verwendung `get-access-control-effect`.

AWS CLI

Um die Wirkung von Zugriffskontrollregeln zu nutzen

Im folgenden `get-access-control-effect` Beispiel werden die Auswirkungen der Zugriffskontrollregeln der angegebenen WorkMail Amazon-Organisation für die angegebene IP-Adresse, die Zugriffsprotokollaktion und die Benutzer-ID abgerufen.

```
aws workmail get-access-control-effect \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --ip-address "192.0.2.0" \  
  --action "WindowsOutlook" \  
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333"
```

Ausgabe:

```
{  
  "Effect": "DENY",  
  "MatchedRules": [  
    "myRule"  
  ]  
}
```

Weitere Informationen finden Sie unter [Arbeiten mit Zugriffskontrollregeln](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [GetAccessControlEffect](#) in der AWS CLI Befehlsreferenz.

get-mailbox-details

Das folgende Codebeispiel zeigt die Verwendung `get-mailbox-details`.

AWS CLI

Um die Postfachdetails eines Benutzers abzurufen

Mit dem folgenden `get-mailbox-details` Befehl werden Details zum Postfach des angegebenen Benutzers abgerufen.

```
aws workmail get-mailbox-details \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

```
--user-id S-1-1-11-1111111111-222222222-3333333333-3333
```

Ausgabe:

```
{
  "MailboxQuota": 51200,
  "MailboxSize": 0.03890800476074219
}
```

Weitere Informationen finden Sie unter [Benutzerkonten verwalten](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [GetMailboxDetails](#) in der AWS CLI Befehlsreferenz.

list-access-control-rules

Das folgende Codebeispiel zeigt die Verwendung `list-access-control-rules`.

AWS CLI

Um Zugriffskontrollregeln aufzulisten

Das folgende `list-access-control-rules` Beispiel listet die Zugriffskontrollregeln für die angegebene WorkMail Amazon-Organisation auf.

```
aws workmail list-access-control-rules \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

Ausgabe:

```
{
  "Rules": [
    {
      "Name": "default",
      "Effect": "ALLOW",
      "Description": "Default WorkMail Rule",
      "DateCreated": 0.0,
      "DateModified": 0.0
    },
    {
      "Name": "myRule",
```

```

    "Effect": "DENY",
    "Description": "my rule",
    "UserIds": [
      "S-1-1-11-1111111111-2222222222-3333333333-3333"
    ],
    "DateCreated": 1581635628.0,
    "DateModified": 1581635628.0
  }
]
}

```

Weitere Informationen finden Sie unter [Arbeiten mit Zugriffskontrollregeln](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [ListAccessControlRules](#) in der AWS CLI Befehlsreferenz.

list-aliases

Das folgende Codebeispiel zeigt die Verwendung `list-aliases`.

AWS CLI

Um Aliase für ein Mitglied aufzulisten

Der folgende `list-aliases` Befehl listet Aliase für das angegebene Mitglied (Benutzer oder Gruppe) auf.

```

aws workmail list-aliases \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333

```

Ausgabe:

```

{
  "Aliases": [
    "exampleAlias@site.awsapps.com",
    "exampleAlias1@site.awsapps.com"
  ]
}

```

- API-Einheiten finden Sie unter [ListAliases AWS CLI](#) Befehlsreferenz.

list-group-members

Das folgende Codebeispiel zeigt die Verwendung `list-group-members`.

AWS CLI

Um Gruppenmitglieder aufzulisten

Der folgende `list-group-members` Befehl listet die Mitglieder der angegebenen Gruppe auf.

```
aws workmail list-group-members \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Ausgabe:

```
{  
  "Members": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Name": "exampleUser1",  
      "Type": "USER",  
      "State": "ENABLED",  
      "EnabledDate": 1532459261.827  
    }  
  ]  
}
```

- API-Einheiten finden Sie [ListGroupMembers](#) unter AWS CLI Befehlsreferenz.

list-groups

Das folgende Codebeispiel zeigt die Verwendung `list-groups`.

AWS CLI

Um eine Liste von Gruppen abzurufen

Mit dem folgenden `list-groups` Befehl werden Zusammenfassungen der Gruppen in der angegebenen Organisation abgerufen.

```
aws workmail list-groups \  

```

```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Ausgabe:

```
{
  "Groups": [
    {
      "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",
      "Name": "exampleGroup1",
      "State": "DISABLED"
    },
    {
      "Id": "S-4-4-44-1122222222-2222233333-3333334444-4444",
      "Name": "exampleGroup2",
      "State": "ENABLED"
    }
  ]
}
```

- API-Einheiten finden Sie [ListGroups](#) in der AWS CLI Befehlsreferenz.

list-mailbox-permissions

Das folgende Codebeispiel zeigt die Verwendung `list-mailbox-permissions`.

AWS CLI

Um Postfachberechtigungen abzurufen

Mit dem folgenden `list-mailbox-permissions` Befehl werden die Postfachberechtigungen abgerufen, die dem Postfach der angegebenen Entität zugeordnet sind.

```
aws workmail list-mailbox-permissions \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Ausgabe:

```
{
  "Permissions": [
    {
```

```

        "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
        "GranteeType": "USER",
        "PermissionValues": [
            "FULL_ACCESS"
        ]
    }
]
}

```

- API-Einheiten finden Sie unter [ListMailboxPermissions AWS CLI Befehlsreferenz](#).

list-organizations

Das folgende Codebeispiel zeigt die Verwendung `list-organizations`.

AWS CLI

Um eine Liste von Organisationen abzurufen

Mit dem folgenden `list-organizations` Befehl werden Zusammenfassungen der nicht gelöschten Organisationen abgerufen.

```
aws workmail list-organizations
```

Ausgabe:

```

{
  "OrganizationSummaries": [
    {
      "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
      "Alias": "exampleAlias",
      "State": "Active"
    }
  ]
}

```

- API-Einheiten finden Sie [ListOrganizations](#) in AWS CLI der Befehlsreferenz.

list-resource-delegates

Das folgende Codebeispiel zeigt die Verwendung `list-resource-delegates`.

AWS CLI

Um die Delegierten für eine Ressource aufzulisten

Mit dem folgenden `list-resource-delegates` Befehl werden die Delegierten abgerufen, die der angegebenen Ressource zugeordnet sind.

```
aws workmail list-resource-delegates \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

Ausgabe:

```
{  
  "Delegates": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Type": "USER"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [ListResourceDelegates AWS CLI](#) Befehlsreferenz.

list-resources

Das folgende Codebeispiel zeigt die Verwendung `list-resources`.

AWS CLI

Um eine Liste von Ressourcen abzurufen

Mit dem folgenden `list-resources` Befehl werden Zusammenfassungen der Ressourcen für die angegebene Organisation abgerufen.

```
aws workmail list-resources \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Ausgabe:

```
{
```

```
"Resources": [  
  {  
    "Id": "r-7afe0efbade843a58cdc10251fce992c",  
    "Name": "exampleRoom1",  
    "Type": "ROOM",  
    "State": "ENABLED"  
  }  
]  
}
```

- API-Einheiten finden Sie unter [ListResources AWS CLI Befehlsreferenz](#).

list-tags-for-resource

Das folgende Codebeispiel zeigt die Verwendung `list-tags-for-resource`.

AWS CLI

Um die Tags für eine Ressource aufzulisten

Das folgende `list-tags-for-resource` Beispiel listet die Tags für die angegebene WorkMail Amazon-Organisation auf.

```
aws workmail list-tags-for-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza
```

Ausgabe:

```
{  
  "Tags": [  
    {  
      "Key": "priority",  
      "Value": "1"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [Organisation kennzeichnen](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [ListTagsForResource](#) in der AWS CLI Befehlsreferenz.

list-users

Das folgende Codebeispiel zeigt die Verwendung `list-users`.

AWS CLI

Um eine Liste von Benutzern abzurufen

Mit dem folgenden `list-users` Befehl werden Zusammenfassungen der Benutzer in der angegebenen Organisation abgerufen.

```
aws workmail list-users \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Ausgabe:

```
{  
  "Users": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Email": "exampleUser1@site.awsapps.com",  
      "Name": "exampleUser1",  
      "State": "ENABLED",  
      "UserRole": "USER",  
      "EnabledDate": 1532459261.827  
    },  
    {  
      "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",  
      "Name": "exampleGuestUser",  
      "State": "DISABLED",  
      "UserRole": "SYSTEM_USER"  
    }  
  ]  
}
```

- API-Einheiten finden Sie unter [ListUsers AWS CLI](#) Befehlsreferenz.

put-access-control-rule

Das folgende Codebeispiel zeigt die Verwendung `put-access-control-rule`.

AWS CLI

Um eine neue Zugriffskontrollregel einzufügen

Im folgenden `put-access-control-rule` Beispiel wird dem angegebenen Benutzer der Zugriff auf die angegebene WorkMail Amazon-Organisation verweigert.

```
aws workmail put-access-control-rule \  
  --name "myRule" \  
  --effect "DENY" \  
  --description "my rule" \  
  --user-ids "S-1-1-11-1111111111-2222222222-3333333333-3333" \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Arbeiten mit Zugriffskontrollregeln](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [PutAccessControlRule](#) in der AWS CLI Befehlsreferenz.

put-mailbox-permissions

Das folgende Codebeispiel zeigt die Verwendung `put-mailbox-permissions`.

AWS CLI

So legen Sie Postfachberechtigungen fest

Mit dem folgenden `put-mailbox-permissions` Befehl werden vollständige Zugriffsberechtigungen für den angegebenen Empfänger (Benutzer oder Gruppe) festgelegt. Die Entität stellt den Besitzer des Postfachs dar.

```
aws workmail put-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --grantee-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --permission-values FULL_ACCESS
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [PutMailboxPermissions](#) unter AWS CLI Befehlsreferenz.

register-to-work-mail

Das folgende Codebeispiel zeigt die Verwendung `register-to-work-mail`.

AWS CLI

Um eine bestehende oder deaktivierte Entität zu registrieren

Der folgende `register-to-work-mail` Befehl ermöglicht es der angegebenen vorhandenen Entität (Benutzer, Gruppe oder Ressource), Amazon zu verwenden WorkMail.

```
aws workmail register-to-work-mail \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --email exampleGroup1@site.awsapps.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [RegisterToWorkMail](#) unter AWS CLI Befehlsreferenz.

reset-password

Das folgende Codebeispiel zeigt die Verwendung `reset-password`.

AWS CLI

Um das Passwort eines Benutzers zurückzusetzen

Der folgende `reset-password` Befehl setzt das Passwort für den angegebenen Benutzer zurück.

```
aws workmail reset-password \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --password examplePa$$w0rd
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [ResetPassword](#) in der AWS CLI Befehlsreferenz.

tag-resource

Das folgende Codebeispiel zeigt die Verwendung `tag-resource`.

AWS CLI

Um ein Tag auf eine Ressource anzuwenden

Im folgenden `tag-resource` Beispiel wird der angegebenen WorkMail Amazon-Organisation ein Tag mit dem Schlüssel „Priorität“ und dem Wert „1“ zugewiesen.

```
aws workmail tag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tags "Key=priority, Value=1"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Organisation kennzeichnen](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [TagResource](#) in der AWS CLI Befehlsreferenz.

untag-resource

Das folgende Codebeispiel zeigt die Verwendung `untag-resource`.

AWS CLI

Um die Markierung einer Ressource aufzuheben

Im folgenden `untag-resource` Beispiel wird das angegebene Tag aus der angegebenen WorkMail Amazon-Organisation entfernt.

```
aws workmail untag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tag-keys "priority"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Organisation kennzeichnen](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [UntagResource](#) in der AWS CLI Befehlsreferenz.

update-mailbox-quota

Das folgende Codebeispiel zeigt die Verwendung `update-mailbox-quota`.

AWS CLI

Um das Postfachkontingent eines Benutzers zu aktualisieren

Der folgende `update-mailbox-quota` Befehl ändert das Postfachkontingent des angegebenen Benutzers.

```
aws workmail update-mailbox-quota \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --mailbox-quota 40000
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerkonten verwalten](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [UpdateMailboxQuota](#) in der AWS CLI Befehlsreferenz.

update-primary-email-address

Das folgende Codebeispiel zeigt die Verwendung `update-primary-email-address`.

AWS CLI

Um eine primäre E-Mail-Adresse zu aktualisieren

Der folgende `update-primary-email-address` Befehl aktualisiert die primäre E-Mail-Adresse der angegebenen Entität (Benutzer, Gruppe oder Ressource).

```
aws workmail update-primary-email-address \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --email exampleUser2@site.awsapps.com
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UpdatePrimaryEmailAddress](#) unter AWS CLI Befehlsreferenz.

update-resource

Das folgende Codebeispiel zeigt die Verwendung `update-resource`.

AWS CLI

Um eine Ressource zu aktualisieren

Der folgende `update-resource` Befehl aktualisiert den Namen der angegebenen Ressource.

```
aws workmail update-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c \  
  --name exampleRoom2
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

- API-Einheiten finden Sie [UpdateResource](#) unter AWS CLI Befehlsreferenz.

Amazon WorkMail Message Flow-Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von Amazon WorkMail Message Flow Aktionen ausführen und allgemeine Szenarien implementieren. AWS Command Line Interface

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

get-raw-message-content

Das folgende Codebeispiel zeigt die Verwendung `get-raw-message-content`.

AWS CLI

Um den Rohinhalt einer E-Mail-Nachricht abzurufen

Im folgenden `get-raw-message-content` Beispiel wird der Rohinhalt einer E-Mail-Nachricht abgerufen und an eine Textdatei mit dem Namen `test` gesendet.

```
aws workmailmessageflow get-raw-message-content \  
  --message-id a1b2cd34-ef5g-6h7j-k18m-npq9012345rs \  
  test
```

Inhalt der Datei `test` nach der Ausführung des Befehls:

```
Subject: Hello World  
From: =?UTF-8?Q?marymajor_marymajor?= <marymajor@example.com>  
To: =?UTF-8?Q?mateojackson=40example=2Enet?= <mateojackson@example.net>  
Date: Thu, 7 Nov 2019 19:22:46 +0000  
Mime-Version: 1.0  
Content-Type: multipart/alternative;  
  boundary="=_EXAMPLE+"  
References: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>  
X-Priority: 3 (Normal)  
X-Mailer: Amazon WorkMail  
Thread-Index: EXAMPLE  
Thread-Topic: Hello World  
Message-Id: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>
```

This is a multi-part message in MIME format. Your mail reader does not understand MIME message format.

--=_EXAMPLE+

Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

hello world

--=_EXAMPLE+

Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML><html>

<head>

<meta name=3D"Generator" content=3D"Amazon WorkMail v3.0-4510">

```
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Dutf-8">=  
  
<title>testing</title>  
</head>  
<body>  
<p style=3D"margin: 0px; font-family: Arial, Tahoma, Helvetica, sans-seri=  
f; font-size: small;">hello world</p>  
</body>  
</html>  
--=_EXAMPLE+--
```

Weitere Informationen finden Sie unter [Abrufen von Nachrichteninhalten mit AWS Lambda](#) im WorkMail Amazon-Administratorhandbuch.

- API-Einheiten finden Sie unter [GetRawMessageContent AWS CLI](#) Befehlsreferenz.

WorkSpaces Beispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface with Aktionen ausführen und allgemeine Szenarien implementieren WorkSpaces.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

create-tags

Das folgende Codebeispiel zeigt die Verwendung `create-tags`.

AWS CLI

Um Tags zu einem hinzuzufügen Workspace

Im folgenden `create-tags` Beispiel werden die angegebenen Tags zu den angegebenen hinzugefügt WorkSpace.

```
aws workspaces create-tags \  
  --resource-id ws-dk1xzzr417 \  
  --tags Key=Department,Value=Finance
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [WorkSpaces Tag-Ressourcen](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [CreateTags](#) in der AWS CLI Befehlsreferenz.

create-workspaces

Das folgende Codebeispiel zeigt die Verwendung `create-workspaces`.

AWS CLI

Beispiel 1: Um ein zu erstellen AlwaysOn WorkSpace

Das folgende `create-workspaces` Beispiel erstellt eine AlwaysOn WorkSpace für den angegebenen Benutzer unter Verwendung des angegebenen Verzeichnisses und Pakets.

```
aws workspaces create-workspaces \  
  --workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

Ausgabe:

```
{  
  "FailedRequests": [],  
  "PendingRequests": [  
    {  
      "WorkspaceId": "ws-kcqms853t",  
      "DirectoryId": "d-926722edaf",  
      "UserName": "Mateo",  
      "State": "PENDING",  
      "BundleId": "wsb-0zsvgp8fc"  
    }  
  ]  
}
```

Beispiel 2: Um ein zu erstellen AutoStop WorkSpace

Das folgende `create-workspaces` Beispiel erstellt eine AutoStop WorkSpace für den angegebenen Benutzer unter Verwendung des angegebenen Verzeichnisses und Pakets.

```
aws workspaces create-workspaces \
  --
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties
```

Ausgabe:

```
{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-dk1x zr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "State": "PENDING",
      "BundleId": "wsb-0zsvgp8fc"
    }
  ]
}
```

Beispiel 3: Um ein vom Benutzer entkoppeltes Objekt zu erstellen WorkSpace

Im folgenden `create-workspaces` Beispiel wird eine benutzerentkoppelte Datei erstellt, WorkSpace indem der Benutzername auf gesetzt und ein WorkSpace Name[UNDEFINED], eine Verzeichnis-ID und eine Bundle-ID angegeben werden.

```
aws workspaces create-workspaces \
  --workspaces
DirectoryId=d-926722edaf,UserName=''[UNDEFINED]'' ,WorkspaceName=MaryWorkspace1,BundleId=wsb
```

Ausgabe:

```
{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-abcd1234",
      "DirectoryId": "d-926722edaf",
```

```
    "UserName": "[UNDEFINED]",
    "State": "PENDING",
    "BundleId": "wsb-0zsvgp8fc",
    "WorkspaceName": "MaryWorkspace1"
  }
]
```

Weitere Informationen finden Sie unter [Starten eines virtuellen Desktops](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [CreateWorkspaces](#) in der AWS CLI Befehlsreferenz.

delete-tags

Das folgende Codebeispiel zeigt die Verwendung `delete-tags`.

AWS CLI

Um ein Tag aus einem zu löschen Workspace

Im folgenden `delete-tags` Beispiel wird das angegebene Tag aus dem angegebenen Workspace Tag gelöscht.

```
aws workspaces delete-tags \
  --resource-id ws-dk1xzr417 \
  --tag-keys Department
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [WorkSpaces Tag-Ressourcen](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DeleteTags](#) in der AWS CLI Befehlsreferenz.

deregister-workspace-directory

Das folgende Codebeispiel zeigt die Verwendung `deregister-workspace-directory`.

AWS CLI

Um die Registrierung eines Verzeichnisses aufzuheben

Im folgenden `deregister-workspace-directory` Beispiel wird die Registrierung des angegebenen Verzeichnisses aufgehoben.

```
aws workspaces deregister-workspace-directory \  
  --directory-id d-926722edaf
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Registrieren eines Verzeichnisses bei WorkSpaces](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DeregisterWorkspaceDirectory](#) in der AWS CLI Befehlsreferenz.

describe-tags

Das folgende Codebeispiel zeigt die Verwendung `describe-tags`.

AWS CLI

Um die Tags für ein zu beschreiben Workspace

Das folgende `describe-tags` Beispiel beschreibt die Tags für die angegebenen Workspace.

```
aws workspaces describe-tags \  
  --resource-id ws-dk1xzz417
```

Ausgabe:

```
{  
  "TagList": [  
    {  
      "Key": "Department",  
      "Value": "Finance"  
    }  
  ]  
}
```

Weitere Informationen finden Sie unter [WorkSpaces Tag-Ressourcen](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeTags](#) in der AWS CLI Befehlsreferenz.

describe-workspace-bundles

Das folgende Codebeispiel zeigt die Verwendung `describe-workspace-bundles`.

AWS CLI

Um die von Amazon bereitgestellten Bundles aufzulisten

Das folgende `describe-workspace-bundles` Beispiel listet die Namen und die IDs von Amazon bereitgestellten Bundles im Tabellenformat und sortiert nach Namen auf.

```
aws workspaces describe-workspace-bundles \  
  --owner AMAZON \  
  --query "Bundles[*].[Name, BundleId]"
```

Ausgabe:

```
[  
  [  
    "Standard with Amazon Linux 2",  
    "wsb-clj85qzj1"  
  ],  
  [  
    "Performance with Windows 10 (Server 2016 based)",  
    "wsb-gm4d5tx2v"  
  ],  
  [  
    "PowerPro with Windows 7",  
    "wsb-1pzkp0bx4"  
  ],  
  [  
    "Power with Amazon Linux 2",  
    "wsb-2bs6k5lgn"  
  ],  
  [  
    "Graphics with Windows 10 (Server 2019 based)",  
    "wsb-03gyjnfyy"  
  ],  
  ...  
]
```

Weitere Informationen finden Sie unter [WorkSpaces Bundles und Images](#) im Amazon WorkSpaces Administration Guide.

- API-Einheiten finden Sie [DescribeWorkspaceBundles](#) in der AWS CLI Befehlsreferenz.

describe-workspace-directories

Das folgende Codebeispiel zeigt die Verwendung `describe-workspace-directories`.

AWS CLI

Um ein registriertes Verzeichnis zu beschreiben

Das folgende `describe-workspace-directories` Beispiel beschreibt das angegebene registrierte Verzeichnis.

```
aws workspaces describe-workspace-directories \  
  --directory-ids d-926722edaf
```

Ausgabe:

```
{  
  "Directories": [  
    {  
      "DirectoryId": "d-926722edaf",  
      "Alias": "d-926722edaf",  
      "DirectoryName": "example.com",  
      "RegistrationCode": "WSpdx+9RJ8JT",  
      "SubnetIds": [  
        "subnet-9d19c4c6",  
        "subnet-500d5819"  
      ],  
      "DnsIpAddresses": [  
        "172.16.1.140",  
        "172.16.0.30"  
      ],  
      "CustomerUserName": "Administrator",  
      "IamRoleId": "arn:aws:iam::123456789012:role/workspaces_DefaultRole",  
      "DirectoryType": "SIMPLE_AD",  
      "WorkspaceSecurityGroupId": "sg-0d89e927e5645d7c5",  
      "State": "REGISTERED",  
      "WorkspaceCreationProperties": {  
        "EnableWorkDocs": false,  
        "EnableInternetAccess": false,  
        "UserEnabledAsLocalAdministrator": true,  
        "EnableMaintenanceMode": true  
      }  
    }  
  ]  
}
```

```

    },
    "WorkspaceAccessProperties": {
      "DeviceTypeWindows": "ALLOW",
      "DeviceTypeOsx": "ALLOW",
      "DeviceTypeWeb": "DENY",
      "DeviceTypeIos": "ALLOW",
      "DeviceTypeAndroid": "ALLOW",
      "DeviceTypeChromeOs": "ALLOW",
      "DeviceTypeZeroClient": "ALLOW",
      "DeviceTypeLinux": "DENY"
    },
    "Tenancy": "SHARED",
    "SelfservicePermissions": {
      "RestartWorkspace": "ENABLED",
      "IncreaseVolumeSize": "DISABLED",
      "ChangeComputeType": "DISABLED",
      "SwitchRunningMode": "DISABLED",
      "RebuildWorkspace": "DISABLED"
    }
  }
]
}

```

Weitere Informationen finden Sie unter [Verzeichnisse verwalten für WorkSpaces](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeWorkspaceDirectories](#) in der AWS CLI Befehlsreferenz.

describe-workspaces-connection-status

Das folgende Codebeispiel zeigt die Verwendung `describe-workspaces-connection-status`.

AWS CLI

Um den Verbindungsstatus eines zu beschreiben WorkSpace

Das folgende `describe-workspaces-connection-status` Beispiel beschreibt den Verbindungsstatus des angegebenen WorkSpace.

```
aws workspaces describe-workspaces-connection-status \
  --workspace-ids ws-dk1xzr417
```

Ausgabe:

```
{
  "WorkspacesConnectionStatus": [
    {
      "WorkspaceId": "ws-dk1x zr417",
      "ConnectionState": "CONNECTED",
      "ConnectionStateCheckTimestamp": 1662526214.744
    }
  ]
}
```

Weitere Informationen finden Sie unter [Verwalten Sie Ihre WorkSpaces](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeWorkspacesConnectionStatus](#) in der AWS CLI Befehlsreferenz.

describe-workspaces

Das folgende Codebeispiel zeigt die Verwendung `describe-workspaces`.

AWS CLI

Um einen zu beschreiben WorkSpace

Das folgende `describe-workspaces` Beispiel beschreibt den angegebenen WorkSpace.

```
aws workspaces describe-workspaces \
  --workspace-ids ws-dk1x zr417
```

Ausgabe:

```
{
  "Workspaces": [
    {
      "WorkspaceId": "ws-dk1x zr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "IpAddress": "172.16.0.175",
      "State": "STOPPED",
      "BundleId": "wsb-0zsvgp8fc",
      "SubnetId": "subnet-500d5819",
      "ComputerName": "WSAMZN-RBSLTTD9",
    }
  ]
}
```

```
    "WorkspaceProperties": {
      "RunningMode": "AUTO_STOP",
      "RunningModeAutoStopTimeoutInMinutes": 60,
      "RootVolumeSizeGib": 80,
      "UserVolumeSizeGib": 10,
      "ComputeTypeName": "VALUE"
    },
    "ModificationStates": []
  }
]
```

Weitere Informationen finden Sie unter [Verwalten Sie Ihre WorkSpaces](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [DescribeWorkspaces](#) in der AWS CLI Befehlsreferenz.

migrate-workspace

Das folgende Codebeispiel zeigt die Verwendung `migrate-workspace`.

AWS CLI

Um ein zu migrieren WorkSpace

Im folgenden `migrate-workspace` Beispiel wird das angegebene Paket WorkSpace zum angegebenen Paket migriert.

```
aws workspaces migrate-workspace \
  --source-workspace-id ws-dk1xzt417 \
  --bundle-id wsb-j4dky1gs4
```

Ausgabe:

```
{
  "SourceWorkspaceId": "ws-dk1xzt417",
  "TargetWorkspaceId": "ws-x5h11bcp5"
}
```

Weitere Informationen finden Sie unter [Migrate a Workspace](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [MigrateWorkspace](#) in der AWS CLI Befehlsreferenz.

modify-workspace-creation-properties

Das folgende Codebeispiel zeigt die Verwendung `modify-workspace-creation-properties`.

AWS CLI

Um eine Workspace-Erstellungseigenschaft eines Verzeichnisses zu ändern

Im folgenden `modify-workspace-creation-properties` Beispiel wird die `EnableInternetAccess` Eigenschaft für das angegebene Verzeichnis aktiviert. Dies ermöglicht die automatische Zuweisung von öffentlichen IP-Adressen für das Verzeichnis, das für das Verzeichnis WorkSpaces erstellt wurde.

```
aws workspaces modify-workspace-creation-properties \  
  --resource-id d-926722edaf \  
  --workspace-creation-properties EnableInternetAccess=true
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Verzeichnisdetails für Sie aktualisieren WorkSpaces](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [ModifyWorkspaceCreationProperties](#) unter AWS CLI Befehlsreferenz.

modify-workspace-properties

Das folgende Codebeispiel zeigt die Verwendung `modify-workspace-properties`.

AWS CLI

Um den Laufmodus eines zu ändern Workspace

Im folgenden `modify-workspace-properties` Beispiel wird der angegebene Laufmodus Workspace auf `gesetzt` `AUTO_STOP`.

```
aws workspaces modify-workspace-properties \  
  --workspace-id ws-dk1xzz417 \  
  --workspace-properties RunningMode=AUTO_STOP
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Modify a Workspace](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [ModifyWorkspaceProperties](#) in der AWS CLI Befehlsreferenz.

modify-workspace-state

Das folgende Codebeispiel zeigt die Verwendung `modify-workspace-state`.

AWS CLI

Um den Status eines zu ändern Workspace

Im folgenden `modify-workspace-state` Beispiel wird der angegebene Status Workspace auf `ADMIN_MAINTENANCE` gesetzt.

```
aws workspaces modify-workspace-state \  
  --workspace-id ws-dk1xzz417 \  
  --workspace-state ADMIN_MAINTENANCE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Workspace Wartung](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [ModifyWorkspaceState](#) in der AWS CLI Befehlsreferenz.

reboot-workspaces

Das folgende Codebeispiel zeigt die Verwendung `reboot-workspaces`.

AWS CLI

Um einen neu zu starten Workspace

Im folgenden `reboot-workspaces` Beispiel wird der angegebene Workspace neu gestartet.

```
aws workspaces reboot-workspaces \  
  --reboot-workspace-requests ws-dk1xzz417
```


Ausgabe:

```
{
  "FailedRequests": []
}
```

Weitere Informationen finden Sie unter [Reboot a WorkSpace](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [RebootWorkspaces](#) in der AWS CLI Befehlsreferenz.

rebuild-workspaces

Das folgende Codebeispiel zeigt die Verwendung `rebuild-workspaces`.

AWS CLI

Um einen neu zu erstellen WorkSpace

Im folgenden `rebuild-workspaces` Beispiel wird das angegebene WorkSpace neu erstellt.

```
aws workspaces rebuild-workspaces \
  --rebuild-workspace-requests ws-dk1xzz417
```

Ausgabe:

```
{
  "FailedRequests": []
}
```

Weitere Informationen finden Sie unter [Rebuild a WorkSpace](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [RebuildWorkspaces](#) in der AWS CLI Befehlsreferenz.

register-workspace-directory

Das folgende Codebeispiel zeigt die Verwendung `register-workspace-directory`.

AWS CLI

Um ein Verzeichnis zu registrieren

Im folgenden `register-workspace-directory` Beispiel wird das angegebene Verzeichnis für die Verwendung mit Amazon registriert WorkSpaces.

```
aws workspaces register-workspace-directory \  
  --directory-id d-926722edaf \  
  --no-enable-work-docs
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Registrieren eines Verzeichnisses bei WorkSpaces](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [RegisterWorkspaceDirectory](#) in der AWS CLI Befehlsreferenz.

restore-workspace

Das folgende Codebeispiel zeigt die Verwendung `restore-workspace`.

AWS CLI

Um eine wiederherzustellen WorkSpace

Das folgende `restore-workspace` Beispiel stellt die angegebene Datei wieder her WorkSpace.

```
aws workspaces restore-workspace \  
  --workspace-id ws-dk1xzr417
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Wiederherstellen a WorkSpace](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [RestoreWorkspace](#) in der AWS CLI Befehlsreferenz.

start-workspaces

Das folgende Codebeispiel zeigt die Verwendung `start-workspaces`.

AWS CLI

Um ein zu starten AutoStop WorkSpace

Im folgenden `start-workspaces` Beispiel wird der angegebene gestartet WorkSpace. Der WorkSpace muss den Betriebsmodus `habenAutoStop`.

```
aws workspaces start-workspaces \  
  --start-workspace-requests WorkspaceId=ws-dk1xzz417
```

Ausgabe:

```
{  
  "FailedRequests": []  
}
```

Weitere Informationen finden Sie unter [Stopp und Start AutoStop WorkSpace im WorkSpaces Amazon-Administratorhandbuch](#).

- API-Einheiten finden Sie [StartWorkspaces](#) in der AWS CLI Befehlsreferenz.

stop-workspaces

Das folgende Codebeispiel zeigt die Verwendung `stop-workspaces`.

AWS CLI

Um ein zu stoppen AutoStop WorkSpace

Im folgenden `stop-workspaces` Beispiel wird der angegebene Vorgang beendet WorkSpace. Der WorkSpace muss den Betriebsmodus `habenAutoStop`.

```
aws workspaces stop-workspaces \  
  --stop-workspace-requests WorkspaceId=ws-dk1xzz417
```

Ausgabe:

```
{  
  "FailedRequests": []  
}
```

Weitere Informationen finden Sie unter [Stopp und Start AutoStop WorkSpace im WorkSpaces Amazon-Administratorhandbuch](#).

- API-Einheiten finden Sie [StopWorkspaces](#) in der AWS CLI Befehlsreferenz.

terminate-workspaces

Das folgende Codebeispiel zeigt die Verwendung `terminate-workspaces`.

AWS CLI

Um einen zu beenden Workspace

Das folgende `terminate-workspaces` Beispiel beendet den angegebenen Workspace.

```
aws workspaces terminate-workspaces \  
  --terminate-workspace-requests ws-dk1xzz417
```

Ausgabe:

```
{  
  "FailedRequests": []  
}
```

Weitere Informationen finden Sie unter [Delete a Workspace](#) im WorkSpaces Amazon-Administratorhandbuch.

- API-Einheiten finden Sie [TerminateWorkspaces](#) in der AWS CLI Befehlsreferenz.

Röntgenbeispiele mit AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie mithilfe von AWS Command Line Interface mit X-Ray Aktionen ausführen und allgemeine Szenarien implementieren.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

batch-traces-get

Das folgende Codebeispiel zeigt die Verwendung `batch-traces-get`.

AWS CLI

Um eine Liste von Traces zu erhalten

Im folgenden `batch-get-traces` Beispiel wird eine Liste von Traces abgerufen, die durch eine ID angegeben sind. Die vollständige Ablaufverfolgung umfasst ein Dokument für jedes Segment, das aus allen Segmentdokumenten kompiliert wurde, die mit der gleichen Ablaufverfolgungs-ID erhalten wurden.

```
aws xray batch-get-traces \
  --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

Ausgabe:

```
{
  "Traces": [
    {
      "Id": "1-5d82881a-0a9126e92a73e971eed891b9",
      "Duration": 0.232,
      "Segments": [
        {
          "Id": "54aff5735b12dd28",
          "Document": "{\"id\":\"54aff5735b12dd28\",\"name\":\
\"Scorekeep\",\"start_time\":1.568835610432E9,\"end_time\":1.568835610664E9,\
\"http\":{\"request\":{\"url\":\"http://scorekeep-env-1.m4fg2pfzpv.us-\
east-2.elasticbeanstalk.com/api/user\",\"method\":\"POST\",\"user_agent\":\
\"curl/7.59.0\",\"client_ip\":\"52.95.4.28\",\"x_forwarded_for\":true},\
\"response\":{\"status\":200}},\"aws\":{\"elastic_beanstalk\":{\"version_label\
\":\"Sample Application-1\",\"deployment_id\":3,\"environment_name\":\"Scorekeep-\
env-1\"},\"ec2\":{\"availability_zone\":\"us-east-2b\",\"instance_id\":\
\"i-0e3cf4d2de0f3f37a\"},\"xray\":{\"sdk_version\":\"1.1.0\",\"sdk\":\"X-Ray for\
Java\"},\"service\":{\"runtime\":\"OpenJDK 64-Bit Server VM\",\"runtime_version\
\":\"1.8.0_222\"},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",\
\"origin\":\"AWS:ElasticBeanstalk:Environment\",\"subsegments\":[{\
\"id\":\"2d6900034ccfe558\",\"name\":\"DynamoDB\",\"start_time\":1.568835610658E9,\
\"end_time\":1.568835610664E9,\"http\":{\"response\":{\"status\":200,
```

```

{"content_length":61}},{"aws":{"table_name":"scorekeep-user","operation":
"UpdateItem","request_id":"TPEIDNDUROMLP0V17U4A79555NVV4KQNS05AEMVJF66Q9ASUAAJG
","resource_names":["scorekeep-user"],"namespace":"aws"}}}
    },
    {
      "Id": "0f278b6334c34e6b",
      "Document": "{\"id\":\"0f278b6334c34e6b\",\"name\":
\"DynamoDB\",\"start_time\":1.568835610658E9,\"end_time\":1.568835610664E9,
\"parent_id\":\"2d6900034ccfe558\",\"inferred\":true,\"http\":{\"response
\":{\"status\":200,\"content_length\":61}},\"aws\":{\"table_name
\":\"scorekeep-user\",\"operation\":\"UpdateItem\",\"request_id\":
\"TPEIDNDUROMLP0V17U4A79555NVV4KQNS05AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-user\"]},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",\"origin
\":\"AWS::DynamoDB::Table\"}"
    }
  ]
},
"UnprocessedTraceIds": []
}

```

Weitere Informationen finden Sie unter [Verwenden von AWS X-Ray API mit dem AWS CLI](#) im AWS X-Ray-Entwicklerhandbuch.

- API-Einheiten finden Sie [BatchTracesGet](#) in der AWS CLI Befehlsreferenz.

create-group

Das folgende Codebeispiel zeigt die Verwendung `create-group`.

AWS CLI

Um eine Gruppe zu erstellen

Im folgenden `create-group` Beispiel wird eine Gruppenressource mit dem Namen `AdminGroup` erstellt. Die Gruppe ruft einen Filterausdruck ab, der die Kriterien der Gruppe als Segment definiert, das sich auf einen bestimmten Dienst bezieht, der einen Fehler oder einen Fehler verursacht.

```

aws xray create-group \
  --group-name "AdminGroup" \
  --filter-expression "service(\"mydomain.com\") {fault OR error}"

```

Ausgabe:

```
{
  "GroupName": "AdminGroup",
  "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",
  "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [CreateGroup](#) in der AWS CLI Befehlsreferenz.

create-sampling-rule

Das folgende Codebeispiel zeigt die Verwendung `create-sampling-rule`.

AWS CLI

Um eine Stichprobenregel zu erstellen

Im folgenden `create-sampling-rule` Beispiel wird eine Regel zur Steuerung des Sampling-Verhaltens für instrumentierte Anwendungen erstellt. Die Regeln werden in einer JSON-Datei bereitgestellt. Die meisten Felder für die Stichprobenregel sind erforderlich, um die Regel zu erstellen.

```
aws xray create-sampling-rule \
  --cli-input-json file://9000-base-scorekeep.json
```

Inhalt von `9000-base-scorekeep.json`:

```
{
  "SamplingRule": {
    "RuleName": "base-scorekeep",
    "ResourceARN": "*",
    "Priority": 9000,
    "FixedRate": 0.1,
    "ReservoirSize": 5,
    "ServiceName": "Scorekeep",
    "ServiceType": "*",
    "Host": "*"
  }
}
```

```
    "HTTPMethod": "*",
    "URLPath": "*",
    "Version": 1
  }
}
```

Ausgabe:

```
{
  "SamplingRuleRecord": {
    "SamplingRule": {
      "RuleName": "base-scorekeep",
      "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/base-
scorekeep",
      "ResourceARN": "*",
      "Priority": 9000,
      "FixedRate": 0.1,
      "ReservoirSize": 5,
      "ServiceName": "Scorekeep",
      "ServiceType": "*",
      "Host": "*",
      "HTTPMethod": "*",
      "URLPath": "*",
      "Version": 1,
      "Attributes": {}
    },
    "CreatedAt": 1530574410.0,
    "ModifiedAt": 1530574410.0
  }
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [CreateSamplingRule](#) in der AWS CLI Befehlsreferenz.

delete-group

Das folgende Codebeispiel zeigt die Verwendung `delete-group`.

AWS CLI

Um eine Gruppe zu löschen

Im folgenden `delete-group` Beispiel wird die angegebene Gruppenressource gelöscht.

```
aws xray delete-group \  
  --group-name "AdminGroup" \  
  --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [DeleteGroup](#) in der AWS CLI Befehlsreferenz.

delete-sampling-rule

Das folgende Codebeispiel zeigt die Verwendung `delete-sampling-rule`.

AWS CLI

Um eine Stichprobenregel zu löschen

Im folgenden `delete-sampling-rule` Beispiel wird die angegebene Stichprobenregel gelöscht. Sie können die Gruppe angeben, indem Sie entweder den Gruppennamen oder die Gruppe verwenden. ARN

```
aws xray delete-sampling-rule \  
  --rule-name polling-scorekeep
```

Ausgabe:

```
{  
  "SamplingRuleRecord": {  
    "SamplingRule": {  
      "RuleName": "polling-scorekeep",  
      "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/polling-  
scorekeep",  
      "ResourceARN": "*",  
      "Priority": 5000,  
      "FixedRate": 0.003,  
      "ReservoirSize": 0,  
      "ServiceName": "Scorekeep",  
      "ServiceType": "*",  
    }  
  }  
}
```

```
        "Host": "*",
        "HTTPMethod": "GET",
        "URLPath": "/api/state/*",
        "Version": 1,
        "Attributes": {}
    },
    "CreatedAt": 1530574399.0,
    "ModifiedAt": 1530574399.0
}
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [DeleteSamplingRule](#) in der AWS CLI Befehlsreferenz.

get-encryption-config

Das folgende Codebeispiel zeigt die Verwendung `get-encryption-config`.

AWS CLI

Um die Verschlüsselungskonfiguration abzurufen

Im folgenden `get-encryption-config` Beispiel wird die aktuelle Verschlüsselungskonfiguration für Ihre AWS X-Ray-Daten abgerufen.

```
aws xray get-encryption-config
```

Ausgabe:

```
{
  "EncryptionConfig": {
    "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
    "Status": "ACTIVE",
    "Type": "NONE"
  }
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [GetEncryptionConfig](#) in der AWS CLI Befehlsreferenz.

get-group

Das folgende Codebeispiel zeigt die Verwendung `get-group`.

AWS CLI

Um eine Gruppe abzurufen

Im folgenden `get-group` Beispiel werden Details für die angegebene Gruppenressource angezeigt. Zu den Details gehören der Gruppenname, die Gruppe ARN und der Filterausdruck, der die Kriterien für diese Gruppe definiert. Gruppen können auch von abgerufen werden ARN.

```
aws xray get-group \
  --group-name "AdminGroup"
```

Ausgabe:

```
{
  "Group": [
    {
      "GroupName": "AdminGroup",
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
      "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [GetGroup](#) in der AWS CLI Befehlsreferenz.

get-groups

Das folgende Codebeispiel zeigt die Verwendung `get-groups`.

AWS CLI

Um alle Gruppen abzurufen

Im folgenden Beispiel werden Details für alle aktiven Gruppen angezeigt.

```
aws xray get-groups
```

Ausgabe:

```
{
  "Groups": [
    {
      "GroupName": "AdminGroup",
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
      "FilterExpression": "service(\"example.com\") {fault OR error}"
    },
    {
      "GroupName": "SDETGroup",
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
SDETGroup/987654321",
      "FilterExpression": "responsetime > 2"
    }
  ]
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [GetGroups](#) in der AWS CLI Befehlsreferenz.

get-sampling-rules

Das folgende Codebeispiel zeigt die Verwendung `get-sampling-rules`.

AWS CLI

Um alle Stichprobenregeln abzurufen

Im folgenden `get-sampling-rules` Beispiel werden Details zu allen verfügbaren Stichprobenregeln angezeigt. :

```
aws xray get-sampling-rules
```

Ausgabe:

```
{
  "SamplingRuleRecords": [
    {
      "SamplingRule": {
        "RuleName": "Default",
        "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/Default",
        "ResourceARN": "*",
        "Priority": 10000,
        "FixedRate": 0.01,
        "ReservoirSize": 0,
        "ServiceName": "*",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1,
        "Attributes": {}
      },
      "CreatedAt": 0.0,
      "ModifiedAt": 1530558121.0
    },
    {
      "SamplingRule": {
        "RuleName": "base-scorekeep",
        "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/base-scorekeep",
        "ResourceARN": "*",
        "Priority": 9000,
        "FixedRate": 0.1,
        "ReservoirSize": 2,
        "ServiceName": "Scorekeep",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1,
        "Attributes": {}
      },
      "CreatedAt": 1530573954.0,
      "ModifiedAt": 1530920505.0
    },
    {
      "SamplingRule": {
```

```

    "RuleName": "polling-scorekeep",
    "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/polling-
scorekeep",
    "ResourceARN": "*",
    "Priority": 5000,
    "FixedRate": 0.003,
    "ReservoirSize": 0,
    "ServiceName": "Scorekeep",
    "ServiceType": "*",
    "Host": "*",
    "HTTPMethod": "GET",
    "URLPath": "/api/state/*",
    "Version": 1,
    "Attributes": {}
  },
  "CreatedAt": 1530918163.0,
  "ModifiedAt": 1530918163.0
}
]
}

```

Weitere Informationen finden Sie unter [Verwenden von Sampling-Regeln mit X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [GetSamplingRules](#) in der AWS CLI Befehlsreferenz.

get-sampling-targets

Das folgende Codebeispiel zeigt die Verwendung `get-sampling-targets`.

AWS CLI

Um ein Stichprobenkontingent anzufordern

Im folgenden `get-sampling-targets` Beispiel wird ein Stichprobenkontingent für Regeln angefordert, die der Dienst für Stichprobenanfragen verwendet. Die Antwort von AWS X-Ray beinhaltet ein Kontingent, das verwendet werden kann, anstatt Kredite aus dem Reservoir aufzunehmen.

```

aws xray get-sampling-targets \
  --sampling-statistics-documents '[ { "RuleName": "base-scorekeep", "ClientID":
"ABCDEF1234567890ABCDEF10", "Timestamp": "2018-07-07T00:20:06", "RequestCount": 110,

```

```
"SampledCount": 20, "BorrowCount": 10 }, { "RuleName": "polling-scorekeep", 31,
"BorrowCount": 0 } ]'
```

Ausgabe:

```
{
  "SamplingTargetDocuments": [
    {
      "RuleName": "base-scorekeep",
      "FixedRate": 0.1,
      "ReservoirQuota": 2,
      "ReservoirQuotaTTL": 1530923107.0,
      "Interval": 10
    },
    {
      "RuleName": "polling-scorekeep",
      "FixedRate": 0.003,
      "ReservoirQuota": 0,
      "ReservoirQuotaTTL": 1530923107.0,
      "Interval": 10
    }
  ],
  "LastRuleModification": 1530920505.0,
  "UnprocessedStatistics": []
}
```

Weitere Informationen finden Sie unter [Verwenden von Sampling-Regeln mit X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [GetSamplingTargets](#) in der AWS CLI Befehlsreferenz.

get-service-graph

Das folgende Codebeispiel zeigt die Verwendung von `get-service-graph`.

AWS CLI

Um ein Service-Diagramm zu erhalten

Im folgenden Beispiel wird ein Dokument innerhalb eines bestimmten Zeitraums angezeigt, in dem Dienste beschrieben werden, die eingehende Anfragen verarbeiten, sowie die nachgelagerten Dienste, die sie daraufhin aufrufen. :

```
aws xray get-service-graph \  
--start-time 1568835392.0 \  
--end-time 1568835446.0
```

Ausgabe:

```
{  
  "Services": [  
    {  
      "ReferenceId": 0,  
      "Name": "Scorekeep",  
      "Names": [  
        "Scorekeep"  
      ],  
      "Root": true,  
      "Type": "AWS::ElasticBeanstalk::Environment",  
      "State": "active",  
      "StartTime": 1568835392.0,  
      "EndTime": 1568835446.0,  
      "Edges": [  
        {  
          "ReferenceId": 1,  
          "StartTime": 1568835392.0,  
          "EndTime": 1568835446.0,  
          "SummaryStatistics": {  
            "OkCount": 14,  
            "ErrorStatistics": {  
              "ThrottleCount": 0,  
              "OtherCount": 0,  
              "TotalCount": 0  
            },  
            "FaultStatistics": {  
              "OtherCount": 0,  
              "TotalCount": 0  
            },  
            "TotalCount": 14,  
            "TotalResponseTime": 0.13  
          },  
          "ResponseTimeHistogram": [  
            {  
              "Value": 0.008,  
              "Count": 1  
            }  
          ],  
        }  
      ],  
    }  
  ]  
}
```



```
        {
            "Value": 0.005,
            "Count": 7
        },
        {
            "Value": 0.009,
            "Count": 1
        },
        {
            "Value": 0.021,
            "Count": 1
        },
        {
            "Value": 0.038,
            "Count": 1
        },
        {
            "Value": 0.007,
            "Count": 1
        },
        {
            "Value": 0.006,
            "Count": 2
        }
    ],
    "Aliases": []
},
... TRUNCATED FOR BREVITY ...
]
}
],
"StartTime": 1568835392.0,
"EndTime": 1568835446.0,
"ContainsOldGroupVersions": false
}
```

Weitere Informationen finden Sie unter [Verwenden von AWS X-Ray API mit dem AWS CLI](#) im AWS X-Ray-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetServiceGraph](#) in der AWS CLI Befehlsreferenz.

get-trace-summaries

Das folgende Codebeispiel zeigt die Verwendung `get-trace-summaries`.

AWS CLI

Um eine Trace-Zusammenfassung zu erhalten

Im folgenden `get-trace-summaries` Beispiel werden Metadaten für Traces abgerufenIDs, die innerhalb eines bestimmten Zeitraums verfügbar sind.

```
aws xray get-trace-summaries \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

Ausgabe:

```
[  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/  
VSAE93HF/GSSD2NTB/DP0PCC09",  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/  
GCQ2B35P/FREELDFT/4LRE643M",  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/game/  
VSAE93HF/GSSD2NTB/starttime/1568835513",  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/  
move/4MQNA5NN/L99KK2RF/null"  
]
```

Weitere Informationen finden Sie unter [Verwenden von AWS X-Ray API mit dem AWS CLI](#) im AWS X-Ray-Entwicklerhandbuch.

- API-Einheiten finden Sie [GetTraceSummaries](#) in der AWS CLI Befehlsreferenz.

put-encryption-config

Das folgende Codebeispiel zeigt die Verwendung `put-encryption-config`.

AWS CLI

Um die Verschlüsselungskonfiguration zu aktualisieren

Das Folgende `put-encryption-config` `example` updates the encryption configuration for AWS X-Ray data to use the default AWS managed KMS key `aws/xray`.

```
aws xray put-encryption-config \
  --type KMS \
  --key-id alias/aws/xray
```

Ausgabe:

```
{
  "EncryptionConfig": {
    "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-
b0ea215cbba5",
    "Status": "UPDATING",
    "Type": "KMS"
  }
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [PutEncryptionConfig](#) in der AWS CLI Befehlsreferenz.

put-trace-segments

Das folgende Codebeispiel zeigt die Verwendung `put-trace-segments`.

AWS CLI

Um ein Segment hochzuladen

Im folgenden `put-trace-segments` Beispiel werden Segmentdokumente auf AWS X-Ray hochgeladen. Das Segmentdokument wird als Liste von JSON Segmentdokumenten verwendet.

```
aws xray put-trace-segments \
  --trace-segment-documents '{"id\":\"20312a0e2b8809f4\", \"name
\": \"DynamoDB\", \"trace_id\": \"1-5832862d-a43aafded3334a971fe312db\",
\"start_time\": 1.479706157195E9, \"end_time\": 1.479706157202E9, \"parent_id\":
\"79736b962fe3239e\", \"http\": {\"response\": {\"content_length\": 60, \"status
\": 200}}, \"inferred\": true, \"aws\": {\"consistent_read\": false, \"table_name
```

```
\":\\"scorekeep-session-xray\\",\\"operation\\":\\"GetItem\\",\\"request_id\\":
\\"SCAU230M6M8F038UASGC7785ARVV4KQNS05AEMVJF66Q9ASUAAJG\\",\\"resource_names\\":
[\\"scorekeep-session-xray\\"}],\\"origin\\":\\"AWS::DynamoDB::Table\\"}"
```

Ausgabe:

```
{
  "UnprocessedTraceSegments": []
}
```

Weitere Informationen finden Sie unter [Senden von Trace-Daten an AWS X-Ray](#) im AWS X-Ray-Entwicklerhandbuch.

- API-Einheiten finden Sie [PutTraceSegments](#) in der AWS CLI Befehlsreferenz.

update-group

Das folgende Codebeispiel zeigt die Verwendung `update-group`.

AWS CLI

Um eine Gruppe zu aktualisieren

Im folgenden `update-group` Beispiel werden die Kriterien aktualisiert, anhand derer Ablaufverfolgungen für die angegebene Gruppe akzeptiert AdminGroup werden. Sie können die gewünschte Gruppe angeben, indem Sie entweder den Gruppennamen oder die Gruppe verwenden ARN.

```
aws xray update-group \
  --group-name "AdminGroup" \
  --group-arn "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789" \
  --filter-expression "service(\"mydomain.com\") {fault}"
```

Ausgabe:

```
{
  "GroupName": "AdminGroup",
  "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
  "FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [UpdateGroup](#) in der AWS CLI Befehlsreferenz.

update-sampling-rule

Das folgende Codebeispiel zeigt die Verwendung `update-sampling-rule`.

AWS CLI

Um eine Stichprobenregel zu aktualisieren

Im folgenden `update-sampling-rule` Beispiel wird die Konfiguration einer Stichprobenregel geändert. Die Regeln werden aus einer JSON-Datei übernommen. Nur die Felder, die aktualisiert werden, sind erforderlich.

```
aws xray update-sampling-rule \  
--cli-input-json file:///1000-default.json
```

Inhalt von `1000-default.json`:

```
{  
  "SamplingRuleUpdate": {  
    "RuleName": "Default",  
    "FixedRate": 0.01,  
    "ReservoirSize": 0  
  }  
}
```

Ausgabe:

```
{  
  "SamplingRuleRecords": [  
    {  
      "SamplingRule": {  
        "RuleName": "Default",  
        "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/  
Default",  
        "ResourceARN": "*",  
        "Priority": 10000,  
        "FixedRate": 0.01,  

```

```
        "ReservoirSize": 0,  
        "ServiceName": "*",  
        "ServiceType": "*",  
        "Host": "*",  
        "HTTPMethod": "*",  
        "URLPath": "*",  
        "Version": 1,  
        "Attributes": {}  
    },  
    "CreatedAt": 0.0,  
    "ModifiedAt": 1529959993.0  
  }  
]  
}
```

Weitere Informationen finden Sie unter [Konfiguration von Sampling-, Gruppen- und Verschlüsselungseinstellungen mit AWS X-Ray API im AWS X-Ray Developer Guide](#).

- API-Einheiten finden Sie [UpdateSamplingRule](#) in der AWS CLI Befehlsreferenz.

AWS CLI mit Bash-Skriptcodebeispielen

Die Codebeispiele in diesem Thema zeigen Ihnen, wie Sie das with Bash-Skript AWS Command Line Interface mit verwenden. AWS

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wichtigsten Operationen innerhalb eines Dienstes ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Dienstes oder in Kombination mit anderen aufrufen AWS-Services.

Services

- [DynamoDB-Beispiele für die Verwendung AWS CLI mit einem Bash-Skript](#)
- [EC2Amazon-Beispiele für die Verwendung AWS CLI mit dem Bash-Skript](#)
- [HealthImaging Beispiele für die Verwendung AWS CLI mit dem Bash-Skript](#)

- [IAMBeispiele für die Verwendung AWS CLI mit dem Bash-Skript](#)
- [Amazon S3 S3-Beispiele für die Verwendung AWS CLI mit dem Bash-Skript](#)
- [AWS STS Beispiele für die Verwendung AWS CLI mit dem Bash-Skript](#)

DynamoDB-Beispiele für die Verwendung AWS CLI mit einem Bash-Skript

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie das with Bash-Skript AWS Command Line Interface mit DynamoDB verwenden.

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wesentlichen Operationen innerhalb eines Dienstes ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Grundlagen](#)
- [Aktionen](#)

Grundlagen

Erlernen Sie die Grundlagen

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen einer Tabelle, die Filmdaten enthalten kann.
- Einfügen, Abrufen und Aktualisieren eines einzelnen Films in der Tabelle.
- Schreiben Sie Filmdaten aus einer JSON Beispieldatei in die Tabelle.
- Abfragen nach Filmen, die in einem bestimmten Jahr veröffentlicht wurden.
- Scan nach Filmen, die in mehreren Jahren veröffentlicht wurden.
- Löschen eines Films aus der Tabelle und anschließendes Löschen der Tabelle.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

Das DynamoDB-Szenario „Erste Schritte“.

```
#####
# function dynamodb_getting_started_movies
#
# Scenario to create an Amazon DynamoDB table and perform a series of operations on
# the table.
#
# Returns:
#     0 - If successful.
#     1 - If an error occurred.
#####
function dynamodb_getting_started_movies() {

    source ./dynamodb_operations.sh

    key_schema_json_file="dynamodb_key_schema.json"
    attribute_definitions_json_file="dynamodb_attr_def.json"
    item_json_file="movie_item.json"
    key_json_file="movie_key.json"
    batch_json_file="batch.json"
    attribute_names_json_file="attribute_names.json"
    attributes_values_json_file="attribute_values.json"

    echo_repeat "*" 88
    echo
    echo "Welcome to the Amazon DynamoDB getting started demo."
    echo
    echo_repeat "*" 88
    echo

    local table_name
    echo -n "Enter a name for a new DynamoDB table: "
    get_input
    table_name=$get_input_result
}
```



```
local provisioned_throughput="ReadCapacityUnits=5,WriteCapacityUnits=5"

echo '['
{"AttributeName": "year", "KeyType": "HASH"},
{"AttributeName": "title", "KeyType": "RANGE"}
]' >"$key_schema_json_file"

echo '['
{"AttributeName": "year", "AttributeType": "N"},
{"AttributeName": "title", "AttributeType": "S"}
]' >"$attribute_definitions_json_file"

if dynamodb_create_table -n "$table_name" -a "$attribute_definitions_json_file" \
-k "$key_schema_json_file" -p "$provisioned_throughput" 1>/dev/null; then
  echo "Created a DynamoDB table named $table_name"
else
  errecho "The table failed to create. This demo will exit."
  clean_up
  return 1
fi

echo "Waiting for the table to become active...."

if dynamodb_wait_table_active -n "$table_name"; then
  echo "The table is now active."
else
  errecho "The table failed to become active. This demo will exit."
  cleanup "$table_name"
  return 1
fi

echo
echo_repeat "*" 88
echo

echo -n "Enter the title of a movie you want to add to the table: "
get_input
local added_title
added_title=$get_input_result

local added_year
get_int_input "What year was it released? "
added_year=$get_input_result
```

```
local rating
get_float_input "On a scale of 1 - 10, how do you rate it? " "1" "10"
rating=$get_input_result

local plot
echo -n "Summarize the plot for me: "
get_input
plot=$get_input_result

echo '{
  "year": {"N" : ""$added_year""},
  "title": {"S" : ""$added_title""},
  "info": {"M" : {"plot": {"S" : ""$plot""}, "rating": {"N" : ""$rating""} } }
}' >"$item_json_file"

if dynamodb_put_item -n "$table_name" -i "$item_json_file"; then
  echo "The movie '$added_title' was successfully added to the table
'$table_name'."
else
  errecho "Put item failed. This demo will exit."
  clean_up "$table_name"
  return 1
fi

echo
echo_repeat "*" 88
echo

echo "Let's update your movie '$added_title'."
get_float_input "You rated it $rating, what new rating would you give it? " "1"
"10"
rating=$get_input_result

echo -n "You summarized the plot as '$plot'."
echo "What would you say now? "
get_input
plot=$get_input_result

echo '{
  "year": {"N" : ""$added_year""},
  "title": {"S" : ""$added_title""}
}' >"$key_json_file"
```

```
echo '{
  "r": {"N" : ""$rating""},
  "p": {"S" : ""$plot""}
}' >"$item_json_file"

local update_expression="SET info.rating = :r, info.plot = :p"

if dynamodb_update_item -n "$table_name" -k "$key_json_file" -e
"$update_expression" -v "$item_json_file"; then
  echo "Updated '$added_title' with new attributes."
else
  errecho "Update item failed. This demo will exit."
  clean_up "$table_name"
  return 1
fi

echo
echo_repeat "*" 88
echo

echo "We will now use batch write to upload 150 movie entries into the table."

local batch_json
for batch_json in movie_files/movies_*.json; do
  echo "{ \"$table_name\" : $(<"$batch_json") }" >"$batch_json_file"
  if dynamodb_batch_write_item -i "$batch_json_file" 1>/dev/null; then
    echo "Entries in $batch_json added to table."
  else
    errecho "Batch write failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi
done

local title="The Lord of the Rings: The Fellowship of the Ring"
local year="2001"

if get_yes_no_input "Let's move on...do you want to get info about '$title'? (y/n)
"; then
  echo '{
"year": {"N" : ""$year""},
"title": {"S" : ""$title""}
}' >"$key_json_file"
  local info
```

```
info=$(dynamodb_get_item -n "$table_name" -k "$key_json_file")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "Get item failed. This demo will exit."
    clean_up "$table_name"
    return 1
fi

echo "Here is what I found:"
echo "$info"
fi

local ask_for_year=true
while [[ "$ask_for_year" == true ]]; do
    echo "Let's get a list of movies released in a given year."
    get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"
    year=$get_input_result
    echo '{
"#n": "year"
}' >"$attribute_names_json_file"

    echo '{
":v": {"N" :"""$year"""}
}' >"$attributes_values_json_file"

    response=$(dynamodb_query -n "$table_name" -k "#n=:v" -a
"$attribute_names_json_file" -v "$attributes_values_json_file")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "Query table failed. This demo will exit."
    clean_up "$table_name"
    return 1
fi

echo "Here is what I found:"
echo "$response"

if ! get_yes_no_input "Try another year? (y/n) "; then
    ask_for_year=false
fi
done
```

```
echo "Now let's scan for movies released in a range of years. Enter a year: "  
get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"  
local start=$get_input_result  
  
get_int_input "Enter another year: " "1972" "2018"  
local end=$get_input_result  
  
echo '{  
  "#n": "year"  
}' >"$attribute_names_json_file"  
  
echo '{  
  ":v1": {"N" : ""$start""},  
  ":v2": {"N" : ""$end""}  
}' >"$attributes_values_json_file"  
  
response=$(dynamodb_scan -n "$table_name" -f "#n BETWEEN :v1 AND :v2" -a  
"$attribute_names_json_file" -v "$attributes_values_json_file")  
  
# shellcheck disable=SC2181  
if [[ ${?} -ne 0 ]]; then  
  errecho "Scan table failed. This demo will exit."  
  clean_up "$table_name"  
  return 1  
fi  
  
echo "Here is what I found:"  
echo "$response"  
  
echo  
echo_repeat "*" 88  
echo  
  
echo "Let's remove your movie '$added_title' from the table."  
  
if get_yes_no_input "Do you want to remove '$added_title'? (y/n) "; then  
  echo '{  
"year": {"N" : ""$added_year""},  
"title": {"S" : ""$added_title""}  
}' >"$key_json_file"  
  
  if ! dynamodb_delete_item -n "$table_name" -k "$key_json_file"; then  
    errecho "Delete item failed. This demo will exit."  
    clean_up "$table_name"
```

```

        return 1
    fi
fi

if get_yes_no_input "Do you want to delete the table '$table_name'? (y/n) "; then
    if ! clean_up "$table_name"; then
        return 1
    fi
else
    if ! clean_up; then
        return 1
    fi
fi

return 0
}

```

Die in diesem Szenario verwendeten „DynamoDB“-Funktionen.

```

#####
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table to create.
#     -a attribute_definitions -- JSON file path of a list of attributes and their
types.
#     -k key_schema -- JSON file path of a list of attributes and their key types.
#     -p provisioned_throughput -- Provisioned throughput settings for the table.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_create_table() {
    local table_name attribute_definitions key_schema provisioned_throughput response
    local option OPTARG # Required to use getopt command in a function.

#####
# Function usage explanation
#####

```

```
function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table."
    echo " -n table_name -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
types."
    echo " -p provisioned_throughput -- Provisioned throughput settings for the
table."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:a:k:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        a) attribute_definitions="${OPTARG}" ;;
        k) key_schema="${OPTARG}" ;;
        p) provisioned_throughput="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$attribute_definitions" ]]; then
    errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
    usage
    return 1
fi
```

```
fi

if [[ -z "$key_schema" ]]; then
    errecho "ERROR: You must provide a key schema json file path the -k parameter."
    usage
    return 1
fi

if [[ -z "$provisioned_throughput" ]]; then
    errecho "ERROR: You must provide a provisioned throughput json file path the -p
parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:    $table_name"
iecho "    attribute_definitions:  $attribute_definitions"
iecho "    key_schema:    $key_schema"
iecho "    provisioned_throughput:  $provisioned_throughput"
iecho ""

response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"${attribute_definitions}" \
    --key-schema file://"${key_schema}" \
    --provisioned-throughput "$provisioned_throughput")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
fi

return 0
}

#####
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
```



```

# Parameters:
#   -n table_name  -- The name of the table.
#
# Response:
#   - TableStatus:
#   And:
#   0 - Table is active.
#   1 - If it fails.
#####
function dynamodb_describe_table {
    local table_name
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_describe_table"
        echo "Describe the status of a DynamoDB table."
        echo "  -n table_name  -- The name of the table."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$table_name" ]]; then
        errecho "ERROR: You must provide a table name with the -n parameter."
        usage
        return 1
    fi
}

```

```

fi

local table_status
table_status=$(
  aws dynamodb describe-table \
    --table-name "$table_name" \
    --output text \
    --query 'Table.TableStatus'
)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log "$error_code"
  errecho "ERROR: AWS reports describe-table operation failed.$table_status"
  return 1
fi

echo "$table_status"

return 0
}

#####
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#   -n table_name -- The name of the table.
#   -i item -- Path to json file containing the item values.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_put_item() {
  local table_name item response
  local option OPTARG # Required to use getopt command in a function.

  #####
  # Function usage explanation
  #####
  function usage() {

```

```
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -i item -- Path to json file containing the item values."
    echo ""
}

while getopts "n:i:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        i) item="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  item:       $item"
iecho ""
iecho ""

response=$(aws dynamodb put-item \
    --table-name "$table_name" \
```

```

    --item file://"${item}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.$response"
    return 1
fi

return 0

}

#####
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
# Parameters:
#   -n table_name -- The name of the table.
#   -k keys -- Path to json file containing the keys that identify the item to
#   update.
#   -e update expression -- An expression that defines one or more attributes
#   to be updated.
#   -v values -- Path to json file containing the update values.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_update_item() {
    local table_name keys update_expression values response
    local option OPTARG # Required to use getopt command in a function.

#####
# Function usage explanation
#####
function usage() {
    echo "function dynamodb_update_item"
    echo "Update an item in a DynamoDB table."
    echo " -n table_name -- The name of the table."

```

```
    echo " -k keys -- Path to json file containing the keys that identify the item
to update."
    echo " -e update expression -- An expression that defines one or more
attributes to be updated."
    echo " -v values -- Path to json file containing the update values."
    echo ""
}

while getopts "n:k:e:v:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) keys="${OPTARG}" ;;
        e) update_expression="${OPTARG}" ;;
        v) values="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -z "$update_expression" ]]; then
    errecho "ERROR: You must provide an update expression with the -e parameter."
    usage
    return 1
fi
```

```

if [[ -z "$values" ]]; then
    errecho "ERROR: You must provide a values json file path the -v parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  keys:       $keys"
iecho "  update_expression:  $update_expression"
iecho "  values:     $values"

response=$(aws dynamodb update-item \
    --table-name "$table_name" \
    --key file://" $keys" \
    --update-expression "$update_expression" \
    --expression-attribute-values file://" $values")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
fi

return 0
}

#####
# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#   -i item  -- Path to json file containing the items to write.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_batch_write_item() {
    local item response

```

```

local option OPTARG # Required to use getopt command in a function.

#####
# Function usage explanation
#####
function usage() {
    echo "function dynamodb_batch_write_item"
    echo "Write a batch of items into a DynamoDB table."
    echo " -i item -- Path to json file containing the items to write."
    echo ""
}
while getopt "i:h" option; do
    case "${option}" in
        i) item="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  item:       $item"
iecho ""

response=$(aws dynamodb batch-write-item \
    --request-items file://"${item}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then

```

```

    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
fi

return 0
}

#####
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#   -n table_name  -- The name of the table.
#   -k keys        -- Path to json file containing the keys that identify the item to
#   get.
#   [-q query]    -- Optional JMESPath query expression.
#
# Returns:
#   The item as text output.
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_get_item() {
    local table_name keys query response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_get_item"
        echo "Get an item from a DynamoDB table."
        echo " -n table_name  -- The name of the table."
        echo " -k keys        -- Path to json file containing the keys that identify the item
to get."
        echo " [-q query]    -- Optional JMESPath query expression."
        echo ""
    }
    query=""
    while getopt "n:k:q:h" option; do
        case "${option}" in

```



```
n) table_name="${OPTARG}" ;;
k) keys="${OPTARG}" ;;
q) query="${OPTARG}" ;;
h)
    usage
    return 0
    ;;
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -n "$query" ]]; then
    response=$(aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"${keys}" \
        --output text \
        --query "$query")
else
    response=$(
        aws dynamodb get-item \
            --table-name "$table_name" \
            --key file://"${keys}" \
            --output text
    )
fi

local error_code=${?}
```

```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
fi

if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
query inserts on some strings.
else
    echo "$response"
fi

return 0
}

#####
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k key_condition_expression -- The key condition expression.
#     -a attribute_names -- Path to JSON file containing the attribute names.
#     -v attribute_values -- Path to JSON file containing the attribute values.
#     [-p projection_expression] -- Optional projection expression.
#
# Returns:
#     The items as json output.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_query() {
    local table_name key_condition_expression attribute_names attribute_values
projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
function usage() {

```

```
    echo "function dynamodb_query"
    echo "Query a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -k key_condition_expression -- The key condition expression."
    echo " -a attribute_names -- Path to JSON file containing the attribute names."
    echo " -v attribute_values -- Path to JSON file containing the attribute
values."
    echo " [-p projection_expression] -- Optional projection expression."
    echo ""
}

while getopts "n:k:a:v:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) key_condition_expression="${OPTARG}" ;;
        a) attribute_names="${OPTARG}" ;;
        v) attribute_values="${OPTARG}" ;;
        p) projection_expression="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$key_condition_expression" ]]; then
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
fi
```

```

if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
fi

if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
fi

if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"${attribute_names}" \
        --expression-attribute-values file://"${attribute_values}")
else
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"${attribute_names}" \
        --expression-attribute-values file://"${attribute_values}" \
        --projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function dynamodb_scan
#
# This function scans a DynamoDB table.

```

```

#
# Parameters:
#   -n table_name  -- The name of the table.
#   -f filter_expression  -- The filter expression.
#   -a expression_attribute_names  -- Path to JSON file containing the expression
attribute names.
#   -v expression_attribute_values  -- Path to JSON file containing the
expression attribute values.
#   [-p projection_expression]  -- Optional projection expression.
#
# Returns:
#   The items as json output.
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_scan() {
    local table_name filter_expression expression_attribute_names
expression_attribute_values projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_scan"
        echo "Scan a DynamoDB table."
        echo " -n table_name  -- The name of the table."
        echo " -f filter_expression  -- The filter expression."
        echo " -a expression_attribute_names  -- Path to JSON file containing the
expression attribute names."
        echo " -v expression_attribute_values  -- Path to JSON file containing the
expression attribute values."
        echo " [-p projection_expression]  -- Optional projection expression."
        echo ""
    }

    while getopt "n:f:a:v:p:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            f) filter_expression="${OPTARG}" ;;
            a) expression_attribute_names="${OPTARG}" ;;
            v) expression_attribute_values="${OPTARG}" ;;
            p) projection_expression="${OPTARG}" ;;
        esac
    done
}

```

```
h)
  usage
  return 0
  ;;
\?)
  echo "Invalid parameter"
  usage
  return 1
  ;;
esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$filter_expression" ]]; then
  errecho "ERROR: You must provide a filter expression with the -f parameter."
  usage
  return 1
fi

if [[ -z "$expression_attribute_names" ]]; then
  errecho "ERROR: You must provide expression attribute names with the -a
parameter."
  usage
  return 1
fi

if [[ -z "$expression_attribute_values" ]]; then
  errecho "ERROR: You must provide expression attribute values with the -v
parameter."
  usage
  return 1
fi

if [[ -z "$projection_expression" ]]; then
  response=$(aws dynamodb scan \
    --table-name "$table_name" \
    --filter-expression "$filter_expression" \
    --expression-attribute-names file://"${expression_attribute_names} \
```

```

    --expression-attribute-values file://"$expression_attribute_values")
else
    response=$(aws dynamodb scan \
        --table-name "$table_name" \
        --filter-expression "$filter_expression" \
        --expression-attribute-names file://"$expression_attribute_names" \
        --expression-attribute-values file://"$expression_attribute_values" \
        --projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports scan operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k keys -- Path to json file containing the keys that identify the item to
#     delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_item() {
    local table_name keys response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####

```

```
function usage() {
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -k keys -- Path to json file containing the keys that identify the item
to delete."
    echo ""
}
while getopts "n:k:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) keys="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  keys:       $keys"
iecho ""

response=$(aws dynamodb delete-item \
  --table-name "$table_name" \
```



```

    --key file://"${keys}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
fi

return 0

}

#####
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#     -n table_name  -- The name of the table to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_table() {
    local table_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function dynamodb_delete_table"
        echo "Deletes an Amazon DynamoDB table."
        echo " -n table_name  -- The name of the table to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            h)
                usage

```

```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho ""

response=$(aws dynamodb delete-table \
    --table-name "$table_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-table operation failed.$response"
    return 1
fi

return 0
}

```

Die in diesem Szenario verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.

```

```
#####
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  fi
}
```

```

elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API-Einheiten finden Sie in den folgenden Themen in der AWS CLI Befehlsreferenz.
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Abfrage](#)
 - [Scan](#)
 - [UpdateItem](#)

Aktionen

BatchGetItem

Das folgende Codebeispiel zeigt, wie Sie es verwenden `BatchGetItem`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
```

```

# function dynamodb_batch_get_item
#
# This function gets a batch of items from a DynamoDB table.
#
# Parameters:
#     -i item -- Path to json file containing the keys of the items to get.
#
# Returns:
#     The items as json output.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_batch_get_item() {
    local item response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_batch_get_item"
        echo "Get a batch of items from a DynamoDB table."
        echo " -i item -- Path to json file containing the keys of the items to get."
        echo ""
    }

    while getopt "i:h" option; do
        case "${option}" in
            i) item="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$item" ]]; then

```

```

    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

response=$(aws dynamodb batch-get-item \
  --request-items file://"$item")
local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-get-item operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#

```

```

# Returns:
#         0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [BatchGetItem](#) in der AWS CLI Befehlsreferenz.

BatchWriteItem

Das folgende Codebeispiel zeigt die Verwendung `BatchWriteItem`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
```

```

# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#     -i item -- Path to json file containing the items to write.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_batch_write_item() {
    local item response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_batch_write_item"
        echo "Write a batch of items into a DynamoDB table."
        echo " -i item -- Path to json file containing the items to write."
        echo ""
    }
    while getopt "i:h" option; do
        case "${option}" in
            i) item="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$item" ]]; then
        errecho "ERROR: You must provide an item with the -i parameter."
        usage
        return 1
    fi
}

```



```

fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho "    item:    $item"
iecho ""

response=$(aws dynamodb batch-write-item \
  --request-items file://"$item")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports batch-write-item operation failed.$response"
  return 1
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

```

```

}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [BatchWriteItem](#) in der AWS CLI Befehlsreferenz.

CreateTable

Das folgende Codebeispiel zeigt die Verwendung `CreateTable`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table to create.
#     -a attribute_definitions -- JSON file path of a list of attributes and their
#     types.
#     -k key_schema -- JSON file path of a list of attributes and their key types.
#     -p provisioned_throughput -- Provisioned throughput settings for the table.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_create_table() {
    local table_name attribute_definitions key_schema provisioned_throughput response
    local option OPTARG # Required to use getopt command in a function.

#####
# Function usage explanation
#####
function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table."
    echo " -n table_name -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
types."
```

```
    echo " -p provisioned_throughput -- Provisioned throughput settings for the
table."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:a:k:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        a) attribute_definitions="${OPTARG}" ;;
        k) key_schema="${OPTARG}" ;;
        p) provisioned_throughput="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$attribute_definitions" ]]; then
    errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
    usage
    return 1
fi

if [[ -z "$key_schema" ]]; then
    errecho "ERROR: You must provide a key schema json file path the -k parameter."
    usage
    return 1
fi
```

```

if [[ -z "$provisioned_throughput" ]]; then
    errecho "ERROR: You must provide a provisioned throughput json file path the -p
parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  attribute_definitions:  $attribute_definitions"
iecho "  key_schema:  $key_schema"
iecho "  provisioned_throughput:  $provisioned_throughput"
iecho ""

response=$(aws dynamodb create-table \
  --table-name "$table_name" \
  --attribute-definitions file://"${attribute_definitions}" \
  --key-schema file://"${key_schema}" \
  --provisioned-throughput "$provisioned_throughput")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

```

```
fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    }
}
```

```

fi

return 0
}

```

- API-Einheiten finden Sie [CreateTable](#) in der AWS CLI Befehlsreferenz.

DeleteItem

Das folgende Codebeispiel zeigt die Verwendung `DeleteItem`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#   -n table_name  -- The name of the table.
#   -k keys        -- Path to json file containing the keys that identify the item to
#                   delete.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_delete_item() {
    local table_name keys response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    # #####
    function usage() {

```

```
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
    echo " -n table_name  -- The name of the table."
    echo " -k keys      -- Path to json file containing the keys that identify the item
to delete."
    echo ""
}
while getopts "n:k:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) keys="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  keys:       $keys"
iecho ""

response=$(aws dynamodb delete-item \
  --table-name "$table_name" \
  --key file://"keys")
```



```

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#

```

```

# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [DeleteItem](#) in der AWS CLI Befehlsreferenz.

DeleteTable

Das folgende Codebeispiel zeigt die Verwendung `DeleteTable`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#     -n table_name  -- The name of the table to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_table() {
    local table_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function dynamodb_delete_table"
        echo "Deletes an Amazon DynamoDB table."
        echo " -n table_name  -- The name of the table to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
```

```

        usage
        return 1
    ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho ""

response=$(aws dynamodb delete-table \
    --table-name "$table_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-table operation failed.$response"
    return 1
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

```

```
fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    }
}
```

```

fi

return 0
}

```

- API-Einheiten finden Sie [DeleteTable](#) in der AWS CLI Befehlsreferenz.

DescribeTable

Das folgende Codebeispiel zeigt die Verwendung `DescribeTable`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
# Parameters:
#   -n table_name  -- The name of the table.
#
# Response:
#   - TableStatus:
#   And:
#   0 - Table is active.
#   1 - If it fails.
#####
function dynamodb_describe_table {
    local table_name
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {

```

```
    echo "function dynamodb_describe_table"
    echo "Describe the status of a DynamoDB table."
    echo "  -n table_name  -- The name of the table."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) table_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

local table_status
table_status=$(
  aws dynamodb describe-table \
    --table-name "$table_name" \
    --output text \
    --query 'Table.TableStatus'
)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log "$error_code"
  errecho "ERROR: AWS reports describe-table operation failed.$table_status"
  return 1
fi
```

```

echo "$table_status"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then

```



```

    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API-Einheiten finden Sie [DescribeTable](#) in der AWS CLI Befehlsreferenz.

GetItem

Das folgende Codebeispiel zeigt die Verwendung `GetItem`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k keys -- Path to json file containing the keys that identify the item to
# get.
#     [-q query] -- Optional JMESPath query expression.
#
# Returns:
#     The item as text output.
# And:
#     0 - If successful.

```

```

#      1 - If it fails.
#####
function dynamodb_get_item() {
    local table_name keys query response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_get_item"
        echo "Get an item from a DynamoDB table."
        echo " -n table_name -- The name of the table."
        echo " -k keys -- Path to json file containing the keys that identify the item
to get."
        echo " [-q query] -- Optional JMESPath query expression."
        echo ""
    }
    query=""
    while getopt "n:k:q:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            k) keys="${OPTARG}" ;;
            q) query="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$table_name" ]]; then
        errecho "ERROR: You must provide a table name with the -n parameter."
        usage
        return 1
    fi

    if [[ -z "$keys" ]]; then

```

```

    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -n "$query" ]]; then
    response=$(aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"keys" \
        --output text \
        --query "$query")
else
    response=$(
        aws dynamodb get-item \
            --table-name "$table_name" \
            --key file://"keys" \
            --output text
    )
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
fi

if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
query inserts on some strings.
else
    echo "$response"
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho

```

```
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API-Einheiten finden Sie [GetItem](#) in der AWS CLI Befehlsreferenz.

ListTables

Das folgende Codebeispiel zeigt die Verwendung `ListTables`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function dynamodb_list_tables
#
# This function lists all the tables in a DynamoDB.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_list_tables() {
    response=$(aws dynamodb list-tables \
        --output text \
        --query "TableNames")

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
        aws_cli_error_log $error_code
        errecho "ERROR: AWS reports batch-write-item operation failed.$response"
        return 1
    fi

    echo "$response" | tr -s "[:space:]" "\n"

    return 0
}
```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    }
}
```

```

elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API-Einheiten finden Sie [ListTables](#) in der AWS CLI Befehlsreferenz.

PutItem

Das folgende Codebeispiel zeigt die Verwendung `PutItem`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -i item -- Path to json file containing the item values.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_put_item() {
    local table_name item response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####

```

```
function usage() {
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -i item -- Path to json file containing the item values."
    echo ""
}

while getopts "n:i:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        i) item="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  item:       $item"
iecho ""
iecho ""

response=$(aws dynamodb put-item \
```



```

    --table-name "$table_name" \
    --item file://" $item")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.$response"
    return 1
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.

```

```

#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [PutItem](#) in der AWS CLI Befehlsreferenz.

Query

Das folgende Codebeispiel zeigt die Verwendung Query.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k key_condition_expression -- The key condition expression.
#     -a attribute_names -- Path to JSON file containing the attribute names.
#     -v attribute_values -- Path to JSON file containing the attribute values.
#     [-p projection_expression] -- Optional projection expression.
#
# Returns:
#     The items as json output.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_query() {
    local table_name key_condition_expression attribute_names attribute_values
    projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    # #####
    function usage() {
        echo "function dynamodb_query"
        echo "Query a DynamoDB table."
        echo " -n table_name -- The name of the table."
        echo " -k key_condition_expression -- The key condition expression."
        echo " -a attribute_names -- Path to JSON file containing the attribute names."
        echo " -v attribute_values -- Path to JSON file containing the attribute
values."
```

```
    echo " [-p projection_expression] -- Optional projection expression."
    echo ""
}

while getopts "n:k:a:v:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) key_condition_expression="${OPTARG}" ;;
        a) attribute_names="${OPTARG}" ;;
        v) attribute_values="${OPTARG}" ;;
        p) projection_expression="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$key_condition_expression" ]]; then
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
fi

if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
fi

if [[ -z "$attribute_values" ]]; then
```

```

    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
fi

if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"${attribute_names}" \
        --expression-attribute-values file://"${attribute_values}")
else
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"${attribute_names}" \
        --expression-attribute-values file://"${attribute_values}" \
        --projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```

```

}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie unter [Abfrage](#) in der AWS CLI Befehlsreferenz.

Scan

Das folgende Codebeispiel zeigt die Verwendung Scan.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function dynamodb_scan
#
# This function scans a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -f filter_expression -- The filter expression.
#     -a expression_attribute_names -- Path to JSON file containing the expression
#     attribute names.
#     -v expression_attribute_values -- Path to JSON file containing the
#     expression attribute values.
#     [-p projection_expression] -- Optional projection expression.
#
# Returns:
#     The items as json output.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_scan() {
    local table_name filter_expression expression_attribute_names
    expression_attribute_values projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    # #####
    function usage() {
        echo "function dynamodb_scan"
        echo "Scan a DynamoDB table."
    }
}
```

```
    echo " -n table_name -- The name of the table."
    echo " -f filter_expression -- The filter expression."
    echo " -a expression_attribute_names -- Path to JSON file containing the
expression attribute names."
    echo " -v expression_attribute_values -- Path to JSON file containing the
expression attribute values."
    echo " [-p projection_expression] -- Optional projection expression."
    echo ""
}

while getopts "n:f:a:v:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        f) filter_expression="${OPTARG}" ;;
        a) expression_attribute_names="${OPTARG}" ;;
        v) expression_attribute_values="${OPTARG}" ;;
        p) projection_expression="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$filter_expression" ]]; then
    errecho "ERROR: You must provide a filter expression with the -f parameter."
    usage
    return 1
fi

if [[ -z "$expression_attribute_names" ]]; then
```



```
errecho "ERROR: You must provide expression attribute names with the -a
parameter."
usage
return 1
fi

if [[ -z "$expression_attribute_values" ]]; then
errecho "ERROR: You must provide expression attribute values with the -v
parameter."
usage
return 1
fi

if [[ -z "$projection_expression" ]]; then
response=$(aws dynamodb scan \
--table-name "$table_name" \
--filter-expression "$filter_expression" \
--expression-attribute-names file://"${expression_attribute_names}" \
--expression-attribute-values file://"${expression_attribute_values}")
else
response=$(aws dynamodb scan \
--table-name "$table_name" \
--filter-expression "$filter_expression" \
--expression-attribute-names file://"${expression_attribute_names}" \
--expression-attribute-values file://"${expression_attribute_values}" \
--projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
aws_cli_error_log $error_code
errecho "ERROR: AWS reports scan operation failed.$response"
return 1
fi

echo "$response"

return 0
}
```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi
}
```

```
    return 0
}
```

- API-Einheiten finden Sie unter Befehlsreferenz zum [Scannen](#) in der AWS CLI Befehlsreferenz.

UpdateItem

Das folgende Codebeispiel zeigt die Verwendung `UpdateItem`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
#
# Parameters:
#     -n table_name  -- The name of the table.
#     -k keys        -- Path to json file containing the keys that identify the item to
#                       update.
#     -e update expression  -- An expression that defines one or more attributes
#                       to be updated.
#     -v values      -- Path to json file containing the update values.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_update_item() {
    local table_name keys update_expression values response
    local option OPTARG # Required to use getopt command in a function.
```

```
#####  
# Function usage explanation  
#####  
function usage() {  
    echo "function dynamodb_update_item"  
    echo "Update an item in a DynamoDB table."  
    echo " -n table_name -- The name of the table."  
    echo " -k keys -- Path to json file containing the keys that identify the item  
to update."  
    echo " -e update expression -- An expression that defines one or more  
attributes to be updated."  
    echo " -v values -- Path to json file containing the update values."  
    echo ""  
}  
  
while getopts "n:k:e:v:h" option; do  
    case "${option}" in  
        n) table_name="${OPTARG}" ;;  
        k) keys="${OPTARG}" ;;  
        e) update_expression="${OPTARG}" ;;  
        v) values="${OPTARG}" ;;  
        h)  
            usage  
            return 0  
            ;;  
        \?)  
            echo "Invalid parameter"  
            usage  
            return 1  
            ;;  
    esac  
done  
export OPTIND=1  
  
if [[ -z "$table_name" ]]; then  
    errecho "ERROR: You must provide a table name with the -n parameter."  
    usage  
    return 1  
fi  
  
if [[ -z "$keys" ]]; then  
    errecho "ERROR: You must provide a keys json file path the -k parameter."  
    usage  
    return 1  
fi
```

```

fi
if [[ -z "$update_expression" ]]; then
    errecho "ERROR: You must provide an update expression with the -e parameter."
    usage
    return 1
fi

if [[ -z "$values" ]]; then
    errecho "ERROR: You must provide a values json file path the -v parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  keys:       $keys"
iecho "  update_expression:  $update_expression"
iecho "  values:     $values"

response=$(aws dynamodb update-item \
  --table-name "$table_name" \
  --key file://" $keys" \
  --update-expression "$update_expression" \
  --expression-attribute-values file://" $values")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function iecho
#

```

```
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    fi
}
```

```
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- API-Einheiten finden Sie [UpdateItem](#) in der AWS CLI Befehlsreferenz.

EC2 Amazon-Beispiele für die Verwendung AWS CLI mit dem Bash-Skript

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie das AWS Command Line Interface with Bash-Skript mit Amazon EC2 verwenden.

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wichtigsten Operationen innerhalb eines Services ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Grundlagen](#)
- [Aktionen](#)

Grundlagen

Erlernen Sie die Grundlagen

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie ein Schlüsselpaar und eine Sicherheitsgruppe.
- Wählen Sie ein Amazon Machine Image (AMI) und einen kompatiblen Instance-Typ aus und erstellen Sie dann eine Instance.
- Halten Sie die Instance an und starten Sie sie neu.
- Verknüpfen einer Elastic-IP-Adresse mit der Instance.
- Connect zu Ihrer Instance mit her SSH und bereinigen Sie anschließend die Ressourcen.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

Führen Sie ein interaktives Szenario an einer Eingabeaufforderung aus.

```
#####
# function get_started_with_ec2_instances
#
# Runs an interactive scenario that shows how to get started using EC2 instances.
#
# "EC2 access" permissions are needed to run this code.
#
# Returns:
# 0 - If successful.
# 1 - If an error occurred.
#####
function get_started_with_ec2_instances() {
    # Requires version 4 for mapfile.
    local required_version=4.0

    # Get the current Bash version
    # Check if BASH_VERSION is set
    local current_version
    if [[ -n "$BASH_VERSION" ]]; then
        # Convert BASH_VERSION to a number for comparison
        current_version=$BASH_VERSION
    else
        # Get the current Bash version using the bash command
```



```
    current_version=$(bash --version | head -n 1 | awk '{ print $4 }')
fi

# Convert version strings to numbers for comparison
local required_version_num current_version_num
required_version_num=$(echo "$required_version" | awk -F. '{ print ($1 * 10000) +
($2 * 100) + $3 }')
current_version_num=$(echo "$current_version" | awk -F. '{ print ($1 * 10000) +
($2 * 100) + $3 }')

# Compare versions
if ((current_version_num < required_version_num)); then
    echo "Error: This script requires Bash version $required_version or higher."
    echo "Your current Bash version is number is $current_version."
    exit 1
fi

{
    if [ "$EC2_OPERATIONS_SOURCED" != "True" ]; then

        source ./ec2_operations.sh
    fi
}

echo_repeat "*" 88
echo "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started with
instances demo."
echo_repeat "*" 88
echo

echo "Let's create an RSA key pair that you can be use to securely connect to "
echo "your EC2 instance."

echo -n "Enter a unique name for your key: "
get_input
local key_name
key_name=$get_input_result

local temp_dir
temp_dir=$(mktemp -d)
local key_file_name="$temp_dir/${key_name}.pem"

if ec2_create_keypair -n "${key_name}" -f "${key_file_name}"; then
    echo "Created a key pair $key_name and saved the private key to $key_file_name"
```

```
    echo
else
    errecho "The key pair failed to create. This demo will exit."
    return 1
fi

chmod 400 "${key_file_name}"

if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then
    local keys_and_fingerprints
    keys_and_fingerprints="$(ec2_describe_key_pairs)" && {
        local image_name_and_id
        while IFS=$'\n' read -r image_name_and_id; do
            local entries
            IFS=$'\t' read -ra entries <<<"$image_name_and_id"
            echo "Found rsa key ${entries[0]} with fingerprint:"
            echo "    ${entries[1]}"
        done <<<"$keys_and_fingerprints"
    }
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create a security group to manage access to your instance."
echo -n "Enter a unique name for your security group: "
get_input
local security_group_name
security_group_name=$get_input_result
local security_group_id
security_group_id=$(ec2_create_security_group -n "$security_group_name" \
    -d "Security group for EC2 instance") || {
    errecho "The security failed to create. This demo will exit."
    clean_up "$key_name" "$key_file_name"
    return 1
}

echo "Security group created with ID $security_group_id"
echo

local public_ip
public_ip=$(curl -s http://checkip.amazonaws.com)
```

```

echo "Let's add a rule to allow SSH only from your current IP address."
echo "Your public IP address is $public_ip"
echo -n "press return to add this rule to your security group."
get_input

if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i "$public_ip"
-p tcp -f 22 -t 22; then
  errecho "The security group rules failed to update. This demo will exit."
  clean_up "$key_name" "$key_file_name" "$security_group_id"
  return 1
fi

echo "Security group rules updated"

local security_group_description
security_group_description="$(ec2_describe_security_groups -g
"${security_group_id}")" || {
  errecho "Failed to describe security groups. This demo will exit."
  clean_up "$key_name" "$key_file_name" "$security_group_id"
  return 1
}

mapfile -t parameters <<<"$security_group_description"
IFS=$'\t' read -ra entries <<<"${parameters[0]}"
echo "Security group: ${entries[0]}"
echo "  ID: ${entries[1]}"
echo "  VPC: ${entries[2]}"
echo "Inbound permissions:"
IFS=$'\t' read -ra entries <<<"${parameters[1]}"
echo "  IpProtocol: ${entries[0]}"
echo "  FromPort: ${entries[1]}"
echo "  ToPort: ${entries[2]}"
echo "  CidrIp: ${parameters[2]}"

local parameters
parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
  errecho "Failed to get parameters. This demo will exit."
  clean_up "$key_name" "$key_file_name" "$security_group_id"
  return 1
}

local image_ids=""

```

```
mapfile -t parameters <<<"$parameters"
for image_name_and_id in "${parameters[@]}"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
        image_ids+="${values[1]} "
    fi
done

local images
images="$(ec2_describe_images -i "$image_ids")" || {
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

new_line_and_tab_to_list "$images"
local images=("${list_result[@]}")

# Get the size of the array
local images_count=${#images[@]}

if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some options:"
for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
done

integer_input "Please enter the number of the AMI you want to use: " 1
"$((images_count / 3))"
local choice=$get_input_result
choice=$((choice - 1) * 3)

echo "Great choice."
echo
```

```

local architecture=${images[$((choice + 1))]}
local image_id=${images[$((choice + 2))]}
echo "Here are some instance types that support the ${architecture} architecture
of the image:"
response="$(ec2_describe_instance_types -a "${architecture}" -t
"*micro,*small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local instance_types
mapfile -t instance_types <<<"$response"

# Get the size of the array
local instance_types_count=${#instance_types[@]}

echo "Here are some options:"
for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
done

integer_input "Which one do you want to use? " 1 "${#instance_types[@]}"
"
choice=$get_input_result
local instance_type=${instance_types[$((choice - 1))]}
echo "Another great choice."
echo

echo "Creating your instance and waiting for it to start..."
local instance_id
instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k "$key_name"
-s "$security_group_id") || {
    errecho "Failed to run instance. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

ec2_wait_for_instance_running -i "$instance_id"
echo "Your instance is ready:"
echo

local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

```

```
echo
print_instance_details "${instance_details}"

local public_ip
public_ip=$(echo "${instance_details}" | awk '{print $6}')
echo
echo "You can use SSH to connect to your instance"
echo "If the connection attempt times out, you might have to manually update the
SSH ingress rule"
echo "for your IP address in the AWS Management Console."
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
ec2_wait_for_instance_running -i "$instance_id"

echo "Your instance is running again."
local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

public_ip=$(echo "${instance_details}" | awk '{print $6}')

echo "Every time your instance is restarted, its public IP address changes"
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
```

```
echo_repeat "*" 88

echo "You can allocate an Elastic IP address and associate it with your instance"
echo "to keep a consistent IP address even when your instance restarts."

local result
result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
}

local elastic_ip allocation_id
elastic_ip=$(echo "$result" | awk '{print $1}')
allocation_id=$(echo "$result" | awk '{print $2}')

echo "Allocated static Elastic IP address: $elastic_ip"

local association_id
association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id") || {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
"$allocation_id"
    return 1
}

echo "Associated your Elastic IP with your instance."
echo "You can now use SSH to connect to your instance by using the Elastic IP."
connect_to_instance "$key_file_name" "$elastic_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
```

```

ec2_wait_for_instance_running -i "$instance_id"

echo "Your instance is running again."
local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

echo "Because you have associated an Elastic IP with your instance, you can"
echo "connect by using a consistent IP address after the instance restarts."
connect_to_instance "$key_file_name" "$elastic_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

if yes_no_input "Do you want to delete the resources created in this demo: (y/n)
"; then
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \
        "$allocation_id" "$association_id"
else
    echo "The following resources were not deleted."
    echo "Key pair: $key_name"
    echo "Key file: $key_file_name"
    echo "Security group: $security_group_id"
    echo "Instance: $instance_id"
    echo "Elastic IP address: $elastic_ip"
fi
}

#####
# function clean_up
#
# This function cleans up the created resources.
# $1 - The name of the ec2 key pair to delete.
# $2 - The name of the key file to delete.
# $3 - The ID of the security group to delete.
# $4 - The ID of the instance to terminate.
# $5 - The ID of the elastic IP address to release.
# $6 - The ID of the elastic IP address to disassociate.
#
# Returns:

```



```
#      0 - If successful.
#      1 - If an error occurred.
#####
function clean_up() {
    local result=0
    local key_pair_name=$1
    local key_file_name=$2
    local security_group_id=$3
    local instance_id=$4
    local allocation_id=$5
    local association_id=$6

    if [ -n "$association_id" ]; then
        # bashsupport disable=BP2002
        if (ec2_disassociate_address -a "$association_id"); then
            echo "Disassociated elastic IP address with ID $association_id"
        else
            errecho "The elastic IP address disassociation failed."
            result=1
        fi
    fi

    if [ -n "$allocation_id" ]; then
        # bashsupport disable=BP2002
        if (ec2_release_address -a "$allocation_id"); then
            echo "Released elastic IP address with ID $allocation_id"
        else
            errecho "The elastic IP address release failed."
            result=1
        fi
    fi

    if [ -n "$instance_id" ]; then
        # bashsupport disable=BP2002
        if (ec2_terminate_instances -i "$instance_id"); then
            echo "Started terminating instance with ID $instance_id"

            ec2_wait_for_instance_terminated -i "$instance_id"
        else
            errecho "The instance terminate failed."
            result=1
        fi
    fi
fi
```

```

if [ -n "$security_group_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_security_group -i "$security_group_id"); then
        echo "Deleted security group with ID $security_group_id"
    else
        errecho "The security group delete failed."
        result=1
    fi
fi

if [ -n "$key_pair_name" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_keypair -n "$key_pair_name"); then
        echo "Deleted key pair named $key_pair_name"
    else
        errecho "The key pair delete failed."
        result=1
    fi
fi

if [ -n "$key_file_name" ]; then
    rm -f "$key_file_name"
fi

return $result
}

#####
# function ssm_get_parameters_by_path
#
# This function retrieves one or more parameters from the AWS Systems Manager
# Parameter Store
# by specifying a parameter path.
#
# Parameters:
#     -p parameter_path - The path of the parameter(s) to retrieve.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ssm_get_parameters_by_path() {
    local parameter_path response
    local option OPTARG # Required to use getopt command in a function.

```

```
# bashsupport disable=BP5008
function usage() {
    echo "function ssm_get_parameters_by_path"
    echo "Retrieves one or more parameters from the AWS Systems Manager Parameter
Store by specifying a parameter path."
    echo "  -p parameter_path - The path of the parameter(s) to retrieve."
    echo ""
}

# Retrieve the calling parameters.
while getopts "p:h" option; do
    case "${option}" in
        p) parameter_path="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$parameter_path" ]]; then
    errecho "ERROR: You must provide a parameter path with the -p parameter."
    usage
    return 1
fi

response=$(aws ssm get-parameters-by-path \
    --path "$parameter_path" \
    --query "Parameters[*].[Name, Value]" \
    --output text) || {
    aws_cli_error_log $?
    errecho "ERROR: AWS reports get-parameters-by-path operation failed.$response"
    return 1
}

echo "$response"
```

```

    return 0
}

#####
# function print_instance_details
#
# This function prints the details of an Amazon Elastic Compute Cloud (Amazon EC2)
instance.
#
# Parameters:
#     instance_details - The instance details in the format "InstanceId ImageId
InstanceType KeyName VpcId PublicIpAddress State.Name".
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function print_instance_details() {
    local instance_details="$1"

    if [[ -z "${instance_details}" ]]; then
        echo "Error: Missing required instance details argument."
        return 1
    fi

    local instance_id image_id instance_type key_name vpc_id public_ip state
    instance_id=$(echo "${instance_details}" | awk '{print $1}')
    image_id=$(echo "${instance_details}" | awk '{print $2}')
    instance_type=$(echo "${instance_details}" | awk '{print $3}')
    key_name=$(echo "${instance_details}" | awk '{print $4}')
    vpc_id=$(echo "${instance_details}" | awk '{print $5}')
    public_ip=$(echo "${instance_details}" | awk '{print $6}')
    state=$(echo "${instance_details}" | awk '{print $7}')

    echo "    ID: ${instance_id}"
    echo "    Image ID: ${image_id}"
    echo "    Instance type: ${instance_type}"
    echo "    Key name: ${key_name}"
    echo "    VPC ID: ${vpc_id}"
    echo "    Public IP: ${public_ip}"
    echo "    State: ${state}"

    return 0
}

```

```
#####
# function connect_to_instance
#
# This function displays the public IP address of an Amazon Elastic Compute Cloud
  (Amazon EC2) instance and prompts the user to connect to the instance via SSH.
#
# Parameters:
#   $1 - The name of the key file used to connect to the instance.
#   $2 - The public IP address of the instance.
#
# Returns:
#   None
#####
function connect_to_instance() {
  local key_file_name="$1"
  local public_ip="$2"

  # Validate the input parameters
  if [[ -z "$key_file_name" ]]; then
    echo "ERROR: You must provide a key file name as the first argument." >&2
    return 1
  fi

  if [[ -z "$public_ip" ]]; then
    echo "ERROR: You must provide a public IP address as the second argument." >&2
    return 1
  fi

  # Display the public IP address and connection command
  echo "To connect, run the following command:"
  echo "  ssh -i ${key_file_name} ec2-user@${public_ip}"

  # Prompt the user to connect to the instance
  if yes_no_input "Do you want to connect now? (y/n) "; then
    echo "After you have connected, you can return to this example by typing 'exit'"
    ssh -i "${key_file_name}" ec2-user@"${public_ip}"
  fi
}

#####
# function get_input
#
# This function gets user input from the command line.
```

```
#
# Outputs:
#   User input to stdout.
#
# Returns:
#   0
#####
function get_input() {

    if [ -z "${mock_input+x}" ]; then
        read -r get_input_result
    else

        if [ "$mock_input_array_index" -lt ${#mock_input_array[@]} ]; then
            get_input_result="${mock_input_array[$mock_input_array_index]}"
            # bashsupport disable=BP2001
            # shellcheck disable=SC2206
            ((mock_input_array_index++))
            echo -n "$get_input_result"
        else
            echo "MOCK_INPUT_ARRAY has no more elements" 1>&2
            return 1
        fi
    fi

    return 0
}

#####
# function yes_no_input
#
# This function requests a yes/no answer from the user, following to a prompt.
#
# Parameters:
#   $1 - The prompt.
#
# Returns:
#   0 - If yes.
#   1 - If no.
#####
function yes_no_input() {
    if [ -z "$1" ]; then
        echo "Internal error yes_no_input"
        return 1
    fi
}
```

```
fi

local index=0
local response="N"
while [[ $index -lt 10 ]]; do
    index=$((index + 1))
    echo -n "$1"
    if ! get_input; then
        return 1
    fi
    response=$(echo "$get_input_result" | tr '[:upper:]' '[:lower:]')
    if [ "$response" = "y" ] || [ "$response" = "n" ]; then
        break
    else
        echo -e "\nPlease enter or 'y' or 'n'."
    fi
done

echo

if [ "$response" = "y" ]; then
    return 0
else
    return 1
fi
}

#####
# function integer_input
#
# This function prompts the user to enter an integer within a specified range
# and validates the input.
#
# Parameters:
#     $1 - The prompt message to display to the user.
#     $2 - The minimum value of the accepted range.
#     $3 - The maximum value of the accepted range.
#
# Returns:
#     The valid integer input from the user.
#     If the input is invalid or out of range, the function will continue
#     prompting the user until a valid input is provided.
#####
function integer_input() {
```

```
local prompt="$1"
local min_value="$2"
local max_value="$3"
local input=""

while true; do
  # Display the prompt message and wait for user input
  echo -n "$prompt"

  if ! get_input; then
    return 1
  fi

  input="$get_input_result"

  # Check if the input is a valid integer
  if [[ "$input" =~ ^-?[0-9]+$ ]]; then
    # Check if the input is within the specified range
    if ((input >= min_value && input <= max_value)); then
      return 0
    else
      echo "Error: Input, $input, must be between $min_value and $max_value."
    fi
  else
    echo "Error: Invalid input- $input. Please enter an integer."
  fi
done
}
#####
# function new_line_and_tab_to_list
#
# This function takes a string input containing newlines and tabs, and
# converts it into a list (array) of elements.
#
# Parameters:
#     $1 - The input string containing newlines and tabs.
#
# Returns:
#     The resulting list (array) is stored in the global variable
#     'list_result'.
#####
function new_line_and_tab_to_list() {
  local input=$1
  export list_result
```



```

list_result=()
mapfile -t lines <<<"$input"
local line
for line in "${lines[@]}"; do
    IFS=$'\t' read -ra parameters <<<"$line"
    list_result+=("${parameters[@]}")
done
}

#####
# function echo_repeat
#
# This function prints a string 'n' times to stdout.
#
# Parameters:
#     $1 - The string.
#     $2 - Number of times to print the string.
#
# Outputs:
#     String 'n' times to stdout.
#
# Returns:
#     0
#####
function echo_repeat() {
    local end=$2
    for ((i = 0; i < end; i++)); do
        echo -n "$1"
    done
    echo
}

```

Die in diesem Szenario verwendeten „DynamoDB“-Funktionen.

```

#####
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
# 2048-bit RSA key pair
# and writes it to a file.
#

```

```

# Parameters:
#     -n key_pair_name - A key pair name.
#     -f file_path - File to store the key pair.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_create_keypair() {
    local key_pair_name file_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_create_keypair"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
RSA key pair"
        echo " and writes it to a file."
        echo "  -n key_pair_name - A key pair name."
        echo "  -f file_path - File to store the key pair."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:f:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
            f) file_path="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$key_pair_name" ]]; then
        errecho "ERROR: You must provide a key name with the -n parameter."
        usage
    fi
}

```

```

    return 1
fi

if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
fi

response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
}

if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
fi

return 0
}

#####
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
# pairs.
#
# Parameters:
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_key_pairs() {
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_key_pairs"
    }
}

```

```

    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs."
    echo " -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopts "h" option; do
    case "${option}" in
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local response

response=$(aws ec2 describe-key-pairs \
  --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
  --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
}

echo "$response"

return 0
}

#####
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -n security_group_name - The name of the security group.

```

```

#       -d security_group_description - The description of the security group.
#
# Returns:
#       The ID of the created security group, or an error message if the operation
#       fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
#####
function ec2_create_security_group() {
    local security_group_name security_group_description response

    # Function to display usage information
    function usage() {
        echo "function ec2_create_security_group"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -n security_group_name - The name of the security group."
        echo "  -d security_group_description - The description of the security group."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "n:d:h" option; do
        case "${option}" in
            n) security_group_name="${OPTARG}" ;;
            d) security_group_description="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$security_group_name" ]]; then
        errecho "ERROR: You must provide a security group name with the -n parameter."
        return 1
    fi
}

```

```

fi

if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
parameter."
    return 1
fi

# Create the security group
response=$(aws ec2 create-security-group \
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}

#####
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# security groups.
#
# Parameters:
#     -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_security_groups() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_security_groups"
    }

```

```

    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
groups."
    echo "  -g security_group_id - The ID of the security group to describe
(optional)."

```

```
    return 0
}

#####
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
# (Amazon EC2) security group.
#
# Parameters:
#     -g security_group_id - The ID of the security group.
#     -i ip_address - The IP address or CIDR block to authorize.
#     -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#     -f from_port - The start of the port range to authorize.
#     -t to_port - The end of the port range to authorize.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_authorize_security_group_ingress() {
    local security_group_id ip_address protocol from_port to_port response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_authorize_security_group_ingress"
        echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
        EC2) security group."
        echo "  -g security_group_id - The ID of the security group."
        echo "  -i ip_address - The IP address or CIDR block to authorize."
        echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
        echo "  -f from_port - The start of the port range to authorize."
        echo "  -t to_port - The end of the port range to authorize."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "g:i:p:f:t:h" option; do
        case "${option}" in
            g) security_group_id="${OPTARG}" ;;
            i) ip_address="${OPTARG}" ;;
            p) protocol="${OPTARG}" ;;
            f) from_port="${OPTARG}" ;;
```



```
t) to_port="${OPTARG}" ;;
h)
    usage
    return 0
    ;;
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -g parameter."
    usage
    return 1
fi

if [[ -z "$ip_address" ]]; then
    errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
    usage
    return 1
fi

if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
fi

if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
fi

if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
fi
```

```

response=$(aws ec2 authorize-security-group-ingress \
  --group-id "$security_group_id" \
  --cidr "${ip_address}/32" \
  --protocol "$protocol" \
  --port "$from_port-$to_port" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
  return 1
}

return 0
}

#####
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# images.
#
# Parameters:
#   -i image_ids - A space-separated list of image IDs (optional).
#   -h - Display help.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_describe_images() {
  local image_ids response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.

```

```

while getopts "i:h" option; do
  case "${option}" in
    i) image_ids="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$image_ids" ]]; then
  # shellcheck disable=SC2206
  aws_cli_args+=("--image-ids" $image_ids)
fi

response=$(aws ec2 describe-images \
  "${aws_cli_args[@]}" \
  --query 'Images[*].[Description,Architecture,ImageId]' \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports describe-images operation failed.$response"
  return 1
}

echo "$response"

return 0
}

#####
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE Specify the processor architecture (e.g., x86_64)

```

```

# -t, --type INSTANCE_TYPE      Comma-separated list of instance types (e.g.,
                                t2.micro)
# -h, --help                    Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#####
function ec2_describe_instance_types() {
    local architecture=""
    local instance_types=""

    # bashsupport disable=BP5008
    function usage() {
        echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
        echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
(e.g., x86_64)"
        echo "  -t, --type INSTANCE_TYPE            Comma-separated list of instance types
(e.g., t2.micro)"
        echo "  -h, --help                            Show this help message"
    }

    while [[ $# -gt 0 ]]; do
        case "$1" in
            -a | --architecture)
                architecture="$2"
                shift 2
                ;;
            -t | --type)
                instance_types="$2"
                shift 2
                ;;
            -h | --help)
                usage
                return 0
                ;;
            *)
                echo "Unknown argument: $1"
                return 1
                ;;
        esac
    done

    if [[ -z "$architecture" ]]; then

```

```

    errecho "Error: Architecture not specified."
    usage
    return 1
fi

if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '['
{
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=', ' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n """"${items[$i]}"""" >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ', ' >>"$tmp_json_file"
    fi
done
echo -n ']],'
{
    "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=', ' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n """"${items[$i]}"""" >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ', ' >>"$tmp_json_file"
    fi
done

echo -n ']]]' >>"$tmp_json_file"

local response

```

```

response=$(aws ec2 describe-instance-types --filters file://"${tmp_json_file}" \
  --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "${tmp_json_file}"

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  echo "ERROR: AWS reports describe-instance-types operation failed."
  return 1
fi

echo "$response"
return 0
}

#####
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#   -t instance_type - The instance type to use (e.g., t2.micro).
#   -k key_pair_name - The name of the key pair to use.
#   -s security_group_id - The ID of the security group to use.
#   -c count - The number of instances to launch (default: 1).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
  }
}

```

```
    echo " -t instance_type - The instance type to use (e.g., t2.micro)."  
    echo " -k key_pair_name - The name of the key pair to use."  
    echo " -s security_group_id - The ID of the security group to use."  
    echo " -c count - The number of instances to launch (default: 1)."  
    echo " -h - Display help."  
    echo ""  
}  
  
# Retrieve the calling parameters.  
while getopts "i:t:k:s:c:h" option; do  
    case "${option}" in  
        i) image_id="${OPTARG}" ;;  
        t) instance_type="${OPTARG}" ;;  
        k) key_pair_name="${OPTARG}" ;;  
        s) security_group_id="${OPTARG}" ;;  
        c) count="${OPTARG}" ;;  
        h)  
            usage  
            return 0  
            ;;  
        \?)  
            echo "Invalid parameter"  
            usage  
            return 1  
            ;;  
    esac  
done  
export OPTIND=1  
  
if [[ -z "$image_id" ]]; then  
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i  
parameter."  
    usage  
    return 1  
fi  
  
if [[ -z "$instance_type" ]]; then  
    errecho "ERROR: You must provide an instance type with the -t parameter."  
    usage  
    return 1  
fi  
  
if [[ -z "$key_pair_name" ]]; then  
    errecho "ERROR: You must provide a key pair name with the -k parameter."
```

```

    usage
    return 1
fi

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
fi

if [[ -z "$count" ]]; then
    count=1
fi

response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}

#####
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID of the instance to describe (optional).
#     -q query - The query to filter the response (optional).
#     -h - Display help.
#
# Returns:

```



```

#      0 - If successful.
#      1 - If it fails.
#####
function ec2_describe_instances() {
    local instance_id query response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_instances"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID of the instance to describe (optional)."
        echo "  -q query - The query to filter the response (optional)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:q:h" option; do
        case "${option}" in
            i) instance_id="${OPTARG}" ;;
            q) query="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local aws_cli_args=()

    if [[ -n "$instance_id" ]]; then
        # shellcheck disable=SC2206
        aws_cli_args+=("--instance-ids" $instance_id)
    fi

    local query_arg=""

```

```

if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
fi

# shellcheck disable=SC2086
response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}

#####
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_stop_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_stop_instances"
        echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    }
}

```

```

    echo " -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
```

```

    echo " -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopts "i:h" option; do
    case "${option}" in
        i) instance_ids="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 stop-instances \
--instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
}

return 0
}

#####
# function ec2_start_instances
#
```

```

# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances.
#
# Parameters:
#   -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_start_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_start_instances"
        echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$instance_ids" ]]; then

```

```

    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 start-instances \
  --instance-ids "${instance_ids}") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports start-instances operation failed with $response."
  return 1
}

return 0
}

#####
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#   -d domain - The domain for the Elastic IP address (either 'vpc' or
'standard').
#
# Returns:
#   The allocated Elastic IP address, or an error message if the operation
fails.
# And:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_allocate_address() {
  local domain response

  # Function to display usage information
  function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
(Amazon EC2) instances in a specific AWS Region."
    echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
'standard')."
  }

```

```
    echo ""
}

# Parse the command-line arguments
while getopts "d:h" option; do
    case "${option}" in
        d) domain="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
'standard')."
    return 1
fi

if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
fi

# Allocate the Elastic IP address
response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
}

echo "$response"
```

```
    return 0
}

#####
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a allocation_id - The allocation ID of the Elastic IP address to associate.
#     -i instance_id - The ID of the EC2 instance to associate the Elastic IP
# address with.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_associate_address() {
    local allocation_id instance_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_associate_address"
        echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
associate."
        echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
address with."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:i:h" option; do
        case "${option}" in
            a) allocation_id="${OPTARG}" ;;
            i) instance_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)

```

```

        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}

#####
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a association_id - The association ID that represents the association of
#     the Elastic IP address with an instance.

```



```
#
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_disassociate_address() {
    local association_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_disassociate_address"
        echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo "  -a association_id - The association ID that represents the association
of the Elastic IP address with an instance."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) association_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$association_id" ]]; then
        errecho "ERROR: You must provide an association ID with the -a parameter."
        return 1
    fi

    response=$(aws ec2 disassociate-address \
        --association-id "$association_id") || {
```

```

    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
}

return 0
}

#####
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Parameters:
#     -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_release_address() {
    local allocation_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_release_address"
        echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
        (Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
        release."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) allocation_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
        esac
    done
}

```

```

    \?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
  errecho "ERROR: You must provide an allocation ID with the -a parameter."
  return 1
fi

response=$(aws ec2 release-address \
  --allocation-id "$allocation_id") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports release-address operation failed."
  errecho "$response"
  return 1
}

return 0
}

#####
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#   -i instance_ids - A space-separated list of instance IDs.
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_terminate_instances() {
  local instance_ids response
  local option OPTARG # Required to use getopt command in a function.

```

```
# bashsupport disable=BP5008
function usage() {
    echo "function ec2_terminate_instances"
    echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
    echo "  -i instance_ids - A space-separated list of instance IDs."
    echo "  -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopts "i:h" option; do
    case "${option}" in
        i) instance_ids="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Check if instance ID is provided
if [[ -z "${instance_ids}" ]]; then
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
fi

# shellcheck disable=SC2086
response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
    "--query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    "--output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
}
```

```

    return 0
}

#####
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -i security_group_id - The ID of the security group to delete.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_security_group() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_security_group"
        echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -i security_group_id - The ID of the security group to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) security_group_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1
}

```

```

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
}

return 0
}

#####
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_keypair() {
    local key_pair_name response

    local option OPTARG # Required to use getopt command in a function.
    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_keypair"
        echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
        echo "  -n key_pair_name - A key pair name."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;

```

```

    h)
    usage
    return 0
    ;;
    \?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
  esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
  errecho "ERROR: You must provide a key pair name with the -n parameter."
  usage
  return 1
fi

response=$(aws ec2 delete-key-pair \
  --key-name "$key_pair_name") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
  return 1
}

return 0
}

```

Die in diesem Szenario verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()

```

```

#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie in den folgenden Themen in der AWS CLI Befehlsreferenz.

- [AllocateAddress](#)
- [AssociateAddress](#)
- [AuthorizeSecurityGroupIngress](#)
- [CreateKeyPair](#)
- [CreateSecurityGroup](#)
- [DeleteKeyPair](#)

- [DeleteSecurityGroup](#)
- [DescribeImages](#)
- [DescribeInstanceTypes](#)
- [DescribeInstances](#)
- [DescribeKeyPairs](#)
- [DescribeSecurityGroups](#)
- [DisassociateAddress](#)
- [ReleaseAddress](#)
- [RunInstances](#)
- [StartInstances](#)
- [StopInstances](#)
- [TerminateInstances](#)
- [UnmonitorInstances](#)

Aktionen

AllocateAddress

Das folgende Codebeispiel zeigt, wie Sie es verwenden `AllocateAddress`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####  
# function ec2_allocate_address  
#  
# This function allocates an Elastic IP address for use with Amazon Elastic Compute  
# Cloud (Amazon EC2) instances in a specific AWS Region.  
#  
# Parameters:
```

```
# -d domain - The domain for the Elastic IP address (either 'vpc' or
'standard').
#
# Returns:
# The allocated Elastic IP address, or an error message if the operation
fails.
# And:
# 0 - If successful.
# 1 - If it fails.
#
#####
function ec2_allocate_address() {
    local domain response

    # Function to display usage information
    function usage() {
        echo "function ec2_allocate_address"
        echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
(Amazon EC2) instances in a specific AWS Region."
        echo " -d domain - The domain for the Elastic IP address (either 'vpc' or
'standard')."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "d:h" option; do
        case "${option}" in
            d) domain="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$domain" ]]; then
```

```

    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
'standard')."
    return 1
fi

if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
fi

# Allocate the Elastic IP address
response=$(aws ec2 allocate-address \
  --domain "$domain" \
  --query "[PublicIp,AllocationId]" \
  --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:

```

```

#          $1 - The error code returned by the AWS CLI.
#
# Returns:
#          0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [AllocateAddress](#) in der AWS CLI Befehlsreferenz.

AssociateAddress

Das folgende Codebeispiel zeigt, wie man es benutzt `AssociateAddress`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
  Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a allocation_id - The allocation ID of the Elastic IP address to associate.
#   -i instance_id - The ID of the EC2 instance to associate the Elastic IP
  address with.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
  (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
  associate."
    echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
  address with."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:i:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      i) instance_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
    esac
  done
}
```

```

        return 1
        ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```

```

}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [AssociateAddress](#) in der AWS CLI Befehlsreferenz.

AuthorizeSecurityGroupIngress

Das folgende Codebeispiel zeigt, wie man es benutzt `AuthorizeSecurityGroupIngress`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
# (Amazon EC2) security group.
#
# Parameters:
#   -g security_group_id - The ID of the security group.
#   -i ip_address - The IP address or CIDR block to authorize.
#   -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#   -f from_port - The start of the port range to authorize.
#   -t to_port - The end of the port range to authorize.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_authorize_security_group_ingress() {
    local security_group_id ip_address protocol from_port to_port response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_authorize_security_group_ingress"
        echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
        EC2) security group."
        echo "  -g security_group_id - The ID of the security group."
        echo "  -i ip_address - The IP address or CIDR block to authorize."
        echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
        echo "  -f from_port - The start of the port range to authorize."
        echo "  -t to_port - The end of the port range to authorize."
        echo ""
    }
}
```



```
# Retrieve the calling parameters.
while getopts "g:i:p:f:t:h" option; do
  case "${option}" in
    g) security_group_id="${OPTARG}" ;;
    i) ip_address="${OPTARG}" ;;
    p) protocol="${OPTARG}" ;;
    f) from_port="${OPTARG}" ;;
    t) to_port="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
  errecho "ERROR: You must provide a security group ID with the -g parameter."
  usage
  return 1
fi

if [[ -z "$ip_address" ]]; then
  errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
  usage
  return 1
fi

if [[ -z "$protocol" ]]; then
  errecho "ERROR: You must provide a protocol with the -p parameter."
  usage
  return 1
fi

if [[ -z "$from_port" ]]; then
  errecho "ERROR: You must provide a start port with the -f parameter."
  usage
  return 1
fi
```

```

fi

if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
fi

response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
    return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#

```

```

# Returns:
#         0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [AuthorizeSecurityGroupIngress](#) in der AWS CLI Befehlsreferenz.

CreateKeyPair

Das folgende Codebeispiel zeigt, wie man es benutzt `CreateKeyPair`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
```

```

# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
# 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#     -f file_path - File to store the key pair.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_create_keypair() {
    local key_pair_name file_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_create_keypair"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
RSA key pair"
        echo " and writes it to a file."
        echo "  -n key_pair_name - A key pair name."
        echo "  -f file_path - File to store the key pair."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:f:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
            f) file_path="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}

```

```

done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
fi

response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
}

if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
fi

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```

```
#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [CreateKeyPair](#) in der AWS CLI Befehlsreferenz.

CreateSecurityGroup

Das folgende Codebeispiel zeigt, wie man es benutzt `CreateSecurityGroup`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -n security_group_name - The name of the security group.
#     -d security_group_description - The description of the security group.
#
# Returns:
#     The ID of the created security group, or an error message if the operation
#     fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_create_security_group() {
    local security_group_name security_group_description response

    # Function to display usage information
    function usage() {
        echo "function ec2_create_security_group"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -n security_group_name - The name of the security group."
        echo "  -d security_group_description - The description of the security group."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "n:d:h" option; do
        case "${option}" in
            n) security_group_name="${OPTARG}" ;;
            d) security_group_description="${OPTARG}" ;;
        esac
    done
}
```

```
h)
  usage
  return 0
  ;;
\?)
  echo "Invalid parameter"
  usage
  return 1
  ;;
esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$security_group_name" ]]; then
  errecho "ERROR: You must provide a security group name with the -n parameter."
  return 1
fi

if [[ -z "$security_group_description" ]]; then
  errecho "ERROR: You must provide a security group description with the -d
parameter."
  return 1
fi

# Create the security group
response=$(aws ec2 create-security-group \
  --group-name "$security_group_name" \
  --description "$security_group_description" \
  --query "GroupId" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports create-security-group operation failed."
  errecho "$response"
  return 1
}

echo "$response"
return 0
}
```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.


```
#####  
# function errecho  
#  
# This function outputs everything sent to it to STDERR (standard error output).  
#####  
function errecho() {  
    printf "%s\n" "$*" 1>&2  
}  
  
#####  
# function aws_cli_error_log()  
#  
# This function is used to log the error messages from the AWS CLI.  
#  
# The function expects the following argument:  
#     $1 - The error code returned by the AWS CLI.  
#  
# Returns:  
#     0: - Success.  
#  
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- API-Einheiten finden Sie [CreateSecurityGroup](#) in der AWS CLI Befehlsreferenz.

DeleteKeyPair

Das folgende Codebeispiel zeigt, wie man es benutzt `DeleteKeyPair`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_keypair() {
    local key_pair_name response

    local option OPTARG # Required to use getopt command in a function.
    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_keypair"
        echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
        echo "  -n key_pair_name - A key pair name."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
        esac
    done
}
```

```

        h)
        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()

```

```

#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}


```

- API-Einheiten finden Sie [DeleteKeyPair](#) in der AWS CLI Befehlsreferenz.

DeleteSecurityGroup

Das folgende Codebeispiel zeigt, wie man es benutzt `DeleteSecurityGroup`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -i security_group_id - The ID of the security group to delete.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_security_group() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_security_group"
        echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -i security_group_id - The ID of the security group to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) security_group_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
```

```

        usage
        return 1
    ;;
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#

```

```

# Returns:
#         0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [DeleteSecurityGroup](#) in der AWS CLI Befehlsreferenz.

DescribeImages

Das folgende Codebeispiel zeigt, wie man es benutzt `DescribeImages`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
```

```

# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
images.
#
# Parameters:
#   -i image_ids - A space-separated list of image IDs (optional).
#   -h - Display help.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_describe_images() {
    local image_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_images"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
        echo "  -i image_ids - A space-separated list of image IDs (optional)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) image_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local aws_cli_args=()

```



```

if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
fi

response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
}

echo "$response"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#

```

```
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
  fi

  return 0
}
```

- API-Einheiten finden Sie [DescribeImages](#) in der AWS CLI Befehlsreferenz.

DescribeInstanceTypes

Das folgende Codebeispiel zeigt, wie man es benutzt `DescribeInstanceTypes`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
```

```

# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE      Comma-separated list of instance types (e.g.,
t2.micro)
# -h, --help                    Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#####
function ec2_describe_instance_types() {
    local architecture=""
    local instance_types=""

    # bashsupport disable=BP5008
    function usage() {
        echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
        echo "  -a, --architecture ARCHITECTURE Specify the processor architecture
(e.g., x86_64)"
        echo "  -t, --type INSTANCE_TYPE      Comma-separated list of instance types
(e.g., t2.micro)"
        echo "  -h, --help                    Show this help message"
    }

    while [[ $# -gt 0 ]]; do
        case "$1" in
            -a | --architecture)
                architecture="$2"
                shift 2
                ;;
            -t | --type)
                instance_types="$2"
                shift 2
                ;;
            -h | --help)
                usage
                return 0
                ;;
            *)
                echo "Unknown argument: $1"
                return 1
                ;;
        esac
    done
}

```

```
done

if [[ -z "$architecture" ]]; then
    errecho "Error: Architecture not specified."
    usage
    return 1
fi

if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '['
{
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '""${items[$i]}""' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ',' >>"$tmp_json_file"
    fi
done
echo -n ']],'
{
    "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=',' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '""${items[$i]}""' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ',' >>"$tmp_json_file"
    fi
done
```

```

echo -n ']]]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
  --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  echo "ERROR: AWS reports describe-instance-types operation failed."
  return 1
fi

echo "$response"
return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#

```

```
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
  fi

  return 0
}
```

- API-Einheiten finden Sie [DescribeInstanceTypes](#) in der AWS CLI Befehlsreferenz.

DescribeInstances

Das folgende Codebeispiel zeigt, wie man es benutzt `DescribeInstances`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_describe_instances
#
```

```
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances.
#
# Parameters:
#     -i instance_id - The ID of the instance to describe (optional).
#     -q query - The query to filter the response (optional).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_instances() {
    local instance_id query response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_instances"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID of the instance to describe (optional).\"
        echo "  -q query - The query to filter the response (optional).\"
        echo "  -h - Display help.\"
        echo \"\"
    }

    # Retrieve the calling parameters.
    while getopt \"i:q:h\" option; do
        case \"${option}\" in
            i) instance_id=\"${OPTARG}\" ;;
            q) query=\"${OPTARG}\" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo \"Invalid parameter\"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1
```

```

local aws_cli_args=()

if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
fi

local query_arg=""
if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
fi

# shellcheck disable=SC2086
response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```



```
#####  
# function aws_cli_error_log()  
#  
# This function is used to log the error messages from the AWS CLI.  
#  
# The function expects the following argument:  
#     $1 - The error code returned by the AWS CLI.  
#  
# Returns:  
#     0: - Success.  
#  
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- API-Einheiten finden Sie [DescribeInstances](#) in der AWS CLI Befehlsreferenz.

DescribeKeyPairs

Das folgende Codebeispiel zeigt, wie man es benutzt `DescribeKeyPairs`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
# pairs.
#
# Parameters:
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_key_pairs() {
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_key_pairs"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "h" option; do
        case "${option}" in
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
        esac
    done
}
```

```

        usage
        return 1
    ;;
    esac
done
export OPTIND=1

local response

response=$(aws ec2 describe-key-pairs \
  --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
  --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
}

echo "$response"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:

```

```

#          0: - Success.
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
  fi

  return 0
}

```

- API-Einheiten finden Sie [DescribeKeyPairs](#) in der AWS CLI Befehlsreferenz.

DescribeSecurityGroups

Das folgende Codebeispiel zeigt, wie man es benutzt `DescribeSecurityGroups`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function ec2_describe_security_groups

```

```

#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# security groups.
#
# Parameters:
#     -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_security_groups() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_security_groups"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
groups."
        echo "  -g security_group_id - The ID of the security group to describe
(optional)."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "g:h" option; do
        case "${option}" in
            g) security_group_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

```

```

if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.

```

```
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
  fi

  return 0
}
```

- API-Einheiten finden Sie [DescribeSecurityGroups](#) in der AWS CLI Befehlsreferenz.

DisassociateAddress

Das folgende Codebeispiel zeigt, wie man es benutzt `DisassociateAddress`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_disassociate_address
#
```

```

# This function disassociates an Elastic IP address from an Amazon Elastic Compute
Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a association_id - The association ID that represents the association of
the Elastic IP address with an instance.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_disassociate_address() {
    local association_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_disassociate_address"
        echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo " -a association_id - The association ID that represents the association
of the Elastic IP address with an instance."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) association_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$association_id" ]]; then

```



```

    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
fi

response=$(aws ec2 disassociate-address \
  --association-id "$association_id") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports disassociate-address operation failed."
  errecho "$response"
  return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then

```

```

    errecho " One or more S3 transfers failed."
elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API-Einheiten finden Sie [DisassociateAddress](#) in der AWS CLI Befehlsreferenz.

ReleaseAddress

Das folgende Codebeispiel zeigt, wie man es benutzt `ReleaseAddress`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Parameters:
#     -a allocation_id - The allocation ID of the Elastic IP address to release.
#

```

```

# Returns:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_release_address() {
    local allocation_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_release_address"
        echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
release."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) allocation_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$allocation_id" ]]; then
        errecho "ERROR: You must provide an allocation ID with the -a parameter."
        return 1
    fi

    response=$(aws ec2 release-address \
        --allocation-id "$allocation_id") || {
        aws_cli_error_log ${?}
    }
}

```

```

    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    }
}

```

```

elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API-Einheiten finden Sie [ReleaseAddress](#) in der AWS CLI Befehlsreferenz.

RunInstances

Das folgende Codebeispiel zeigt, wie man es benutzt `RunInstances`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#   -t instance_type - The instance type to use (e.g., t2.micro).
#   -k key_pair_name - The name of the key pair to use.
#   -s security_group_id - The ID of the security group to use.
#   -c count - The number of instances to launch (default: 1).
#   -h - Display help.
#
# Returns:
#   0 - If successful.

```

```

# 1 - If it fails.
#####
function ec2_run_instances() {
    local image_id instance_type key_pair_name security_group_id count response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo " -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo " -t instance_type - The instance type to use (e.g., t2.micro)."
    echo " -k key_pair_name - The name of the key pair to use."
    echo " -s security_group_id - The ID of the security group to use."
    echo " -c count - The number of instances to launch (default: 1)."
    echo " -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopt "i:t:k:s:c:h" option; do
    case "${option}" in
        i) image_id="${OPTARG}" ;;
        t) instance_type="${OPTARG}" ;;
        k) key_pair_name="${OPTARG}" ;;
        s) security_group_id="${OPTARG}" ;;
        c) count="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage

```

```
    return 1
fi

if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
fi

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
fi

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
fi

if [[ -z "$count" ]]; then
    count=1
fi

response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}
```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
}
```


- API-Einheiten finden Sie [RunInstances](#) in der AWS CLI Befehlsreferenz.

StartInstances

Das folgende Codebeispiel zeigt, wie man es benutzt `StartInstances`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_start_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_start_instances"
        echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-separated)."
        echo "  -h - Display help."
        echo ""
    }
}
```

```

}

# Retrieve the calling parameters.
while getopts "i:h" option; do
  case "${option}" in
    i) instance_ids="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
  errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
  usage
  return 1
fi

response=$(aws ec2 start-instances \
  --instance-ids "${instance_ids}") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports start-instances operation failed with $response."
  return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).

```

```
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [StartInstances](#) in der AWS CLI Befehlsreferenz.

StopInstances

Das folgende Codebeispiel zeigt, wie man es benutzt `StopInstances`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_stop_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_stop_instances"
        echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
```

```

    i) instance_ids="${OPTARG}" ;;
    h)
        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 stop-instances \
--instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```

```
#####  
# function aws_cli_error_log()  
#  
# This function is used to log the error messages from the AWS CLI.  
#  
# The function expects the following argument:  
#     $1 - The error code returned by the AWS CLI.  
#  
# Returns:  
#     0: - Success.  
#  
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- API-Einheiten finden Sie [StopInstances](#) in der AWS CLI Befehlsreferenz.

TerminateInstances

Das folgende Codebeispiel zeigt, wie man es benutzt `TerminateInstances`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#     -i instance_ids - A space-separated list of instance IDs.
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_terminate_instances() {
    local instance_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_terminate_instances"
        echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_ids - A space-separated list of instance IDs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage

```

```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

# Check if instance ID is provided
if [[ -z "${instance_ids}" ]]; then
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
fi

# shellcheck disable=SC2086
response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
    --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
}

return 0
}

```

Die in diesem Beispiel verwendeten Dienstprogrammfunktionen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```



```
#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- API-Einheiten finden Sie [TerminateInstances](#) in der AWS CLI Befehlsreferenz.

HealthImaging Beispiele für die Verwendung AWS CLI mit dem Bash-Skript

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie das with Bash-Skript AWS Command Line Interface mit verwenden.

HealthImaging

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen in den zugehörigen Szenarien im Kontext sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

CreateDatastore

Das folgende Codebeispiel zeigt die Verwendung `CreateDatastore`.

AWS CLI mit Bash-Skript

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_create_datastore
#
# This function creates an AWS HealthImaging data store for importing DICOM P10
# files.
#
# Parameters:
#     -n data_store_name - The name of the data store.
#
# Returns:
#     The datastore ID.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
```

```
function imaging_create_datastore() {
    local datastore_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_create_datastore"
        echo "Creates an AWS HealthImaging data store for importing DICOM P10 files."
        echo "  -n data_store_name - The name of the data store."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) datastore_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$datastore_name" ]]; then
        errecho "ERROR: You must provide a data store name with the -n parameter."
        usage
        return 1
    fi

    response=$(aws medical-imaging create-datastore \
        --datastore-name "$datastore_name" \
        --output text \
        --query 'datastoreId')

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
        aws_cli_error_log $error_code
    fi
}
```

```

    errecho "ERROR: AWS reports medical-imaging create-datastore operation failed.
$response"
    return 1
fi

echo "$response"

return 0
}

```

- API-Einheiten finden Sie [CreateDatastore](#) in der AWS CLI Befehlsreferenz.

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

DeleteDatastore

Das folgende Codebeispiel zeigt, wie man es benutzt DeleteDatastore.

AWS CLI mit Bash-Skript

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_delete_datastore
#
# This function deletes an AWS HealthImaging data store.
#
# Parameters:
#     -i datastore_id - The ID of the data store.
#
# Returns:

```

```

#      0 - If successful.
#      1 - If it fails.
#####
function imaging_delete_datastore() {
    local datastore_id response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function imaging_delete_datastore"
    echo "Deletes an AWS HealthImaging data store."
    echo "  -i datastore_id - The ID of the data store."
    echo ""
}

# Retrieve the calling parameters.
while getopt "i:h" option; do
    case "${option}" in
        i) datastore_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$datastore_id" ]]; then
    errecho "ERROR: You must provide a data store ID with the -i parameter."
    usage
    return 1
fi

response=$(aws medical-imaging delete-datastore \
    --datastore-id "$datastore_id")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then

```

```

aws_cli_error_log $error_code
errecho "ERROR: AWS reports medical-imaging delete-datastore operation failed.
$response"
return 1
fi

return 0
}

```

- API-Einheiten finden Sie [DeleteDatastore](#) in der AWS CLI Befehlsreferenz.

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

GetDatastore

Das folgende Codebeispiel zeigt, wie man es benutzt `GetDatastore`.

AWS CLI mit Bash-Skript

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_get_datastore
#
# Get a data store's properties.
#
# Parameters:
#     -i data_store_id - The ID of the data store.
#
# Returns:

```

```

#     [datastore_name, datastore_id, datastore_status, datastore_arn,  created_at,
updated_at]
#     And:
#         0 - If successful.
#         1 - If it fails.
#####
function imaging_get_datastore() {
    local datastore_id option OPTARG # Required to use getopt command in a function.
    local error_code
    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_get_datastore"
        echo "Gets a data store's properties."
        echo "  -i datastore_id - The ID of the data store."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) datastore_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$datastore_id" ]]; then
        errecho "ERROR: You must provide a data store ID with the -i parameter."
        usage
        return 1
    fi

    local response

    response=$(
        aws medical-imaging get-datastore \

```

```

--datastore-id "$datastore_id" \
--output text \
--query "[ datastoreProperties.datastoreName,
datastoreProperties.datastoreId, datastoreProperties.datastoreStatus,
datastoreProperties.datastoreArn,  datastoreProperties.createdAt,
datastoreProperties.updatedAt]"
)
error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-datastores operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

```

- API-Einheiten finden Sie [GetDatastore](#) in der AWS CLI Befehlsreferenz.

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

ListDatastores

Das folgende Codebeispiel zeigt, wie man es benutzt `ListDatastores`.

AWS CLI mit Bash-Skript

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```



```
#####
# function imaging_list_datastores
#
# List the HealthImaging data stores in the account.
#
# Returns:
#     [[datastore_name, datastore_id, datastore_status]]
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function imaging_list_datastores() {
    local option OPTARG # Required to use getopt command in a function.
    local error_code
    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_list_datastores"
        echo "Lists the AWS HealthImaging data stores in the account."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "h" option; do
        case "${option}" in
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local response
    response=$(aws medical-imaging list-datastores \
        --output text \
        --query "datastoreSummaries[*][datastoreName, datastoreId, datastoreStatus]")
    error_code=${?}
}
```

```
if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-datastores operation failed.$response"
    return 1
fi

echo "$response"

return 0
}
```

- API-Einheiten finden Sie [ListDatastores](#) in der AWS CLI Befehlsreferenz.

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

IAM-Beispiele für die Verwendung AWS CLI mit dem Bash-Skript

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie das with Bash-Skript AWS Command Line Interface mit verwenden. IAM

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wesentlichen Operationen innerhalb eines Dienstes ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Grundlagen](#)
- [Aktionen](#)

Grundlagen

Erlernen Sie die Grundlagen

Das folgende Codebeispiel veranschaulicht, wie Sie einen Benutzer erstellen und eine Rolle annehmen lassen.

Warning

Verwenden Sie zur Vermeidung von Sicherheitsrisiken keine IAM Benutzer zur Authentifizierung, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie [AWS IAM Identity Center](#).

- Erstellen Sie einen Benutzer ohne Berechtigungen.
- Erstellen einer Rolle, die die Berechtigung zum Auflisten von Amazon-S3-Buckets für das Konto erteilt.
- Hinzufügen einer Richtlinie, damit der Benutzer die Rolle übernehmen kann.
- Übernehmen Sie die Rolle und listen Sie S3-Buckets mit temporären Anmeldeinformationen auf, und bereinigen Sie dann die Ressourcen.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####  
# function iam_create_user_assume_role  
#  
# Scenario to create an IAM user, create an IAM role, and apply the role to the  
# user.  
#  
# "IAM access" permissions are needed to run this code.  
# "STS assume role" permissions are needed to run this code. (Note: It might be  
# necessary to
```

```
#         create a custom policy).
#
# Returns:
#     0 - If successful.
#     1 - If an error occurred.
#####
function iam_create_user_assume_role() {
    {
        if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

            source ./iam_operations.sh
        fi
    }

    echo_repeat "*" 88
    echo "Welcome to the IAM create user and assume role demo."
    echo
    echo "This demo will create an IAM user, create an IAM role, and apply the role to
the user."
    echo_repeat "*" 88
    echo

    echo -n "Enter a name for a new IAM user: "
    get_input
    user_name=$get_input_result

    local user_arn
    user_arn=$(iam_create_user -u "$user_name")

    # shellcheck disable=SC2181
    if [[ ${?} == 0 ]]; then
        echo "Created demo IAM user named $user_name"
    else
        errecho "$user_arn"
        errecho "The user failed to create. This demo will exit."
        return 1
    fi

    local access_key_response
    access_key_response=$(iam_create_user_access_key -u "$user_name")
    # shellcheck disable=SC2181
    if [[ ${?} != 0 ]]; then
        errecho "The access key failed to create. This demo will exit."
        clean_up "$user_name"
    fi
}
```

```
    return 1
fi

IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
local key_name=${access_key_values[0]}
local key_secret=${access_key_values[1]}

echo "Created access key named $key_name"

echo "Wait 10 seconds for the user to be ready."
sleep 10
echo_repeat "*" 88
echo

local iam_role_name
iam_role_name=$(generate_random_name "test-role")
echo "Creating a role named $iam_role_name with user $user_name as the principal."

local assume_role_policy_document="{
  \"Version\": \"2012-10-17\",
  \"Statement\": [{
    \"Effect\": \"Allow\",
    \"Principal\": {\"AWS\": \"$user_arn\"},
    \"Action\": \"sts:AssumeRole\"
  }]
}"

local role_arn
role_arn=$(iam_create_role -n "$iam_role_name" -p "$assume_role_policy_document")

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
  echo "Created IAM role named $iam_role_name"
else
  errecho "The role failed to create. This demo will exit."
  clean_up "$user_name" "$key_name"
  return 1
fi

local policy_name
policy_name=$(generate_random_name "test-policy")
local policy_document="{
  \"Version\": \"2012-10-17\",
  \"Statement\": [{
```

```

        \Effect\": \Allow\",
        \Action\": \s3:ListAllMyBuckets\",
        \Resource\": \arn:aws:s3:::*\"}}]"

local policy_arn
policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
# shellcheck disable=SC2181
if [[ ${?} == 0 ]]; then
    echo "Created IAM policy named $policy_name"
else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name"
    return 1
fi

if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
    echo "Attached policy $policy_arn to role $iam_role_name"
else
    errecho "The policy failed to attach."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
    return 1
fi

local assume_role_policy_document="{
    \Version\": \2012-10-17\",
    \Statement\": [{
        \Effect\": \Allow\",
        \Action\": \sts:AssumeRole\",
        \Resource\": \${role_arn}\"}]}"

local assume_role_policy_name
assume_role_policy_name=$(generate_random_name "test-assume-role-")

# shellcheck disable=SC2181
local assume_role_policy_arn
assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    echo "Created IAM policy named $assume_role_policy_name for sts assume role"
else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
    return 1

```

```
fi

echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
sleep 10
echo_repeat "*" 88
echo

echo "Try to list buckets without the new user assuming the role."
echo_repeat "*" 88
echo

# Set the environment variables for the created user.
# bashsupport disable=BP2001
export AWS_ACCESS_KEY_ID=$key_name
# bashsupport disable=BP2001
export AWS_SECRET_ACCESS_KEY=$key_secret

local buckets
buckets=$(s3_list_buckets)

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. This should not have
happened."
else
    errecho "Because the role with permissions has not been assumed, listing buckets
failed."
fi

echo
echo_repeat "*" 88
echo "Now assume the role $iam_role_name and list the buckets."
echo_repeat "*" 88
echo

local credentials

credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    echo "Assumed role $iam_role_name"
```

```
else
    errecho "Failed to assume role."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
fi

IFS=$'\t ' read -r -a credentials <<<"$credentials"

export AWS_ACCESS_KEY_ID=${credentials[0]}
export AWS_SECRET_ACCESS_KEY=${credentials[1]}
# bashsupport disable=BP2001
export AWS_SESSION_TOKEN=${credentials[2]}

buckets=$(s3_list_buckets)

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets succeeded
because of "
    echo "the assumed role."
else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
fi

local result=0
export AWS_ACCESS_KEY_ID=""
export AWS_SECRET_ACCESS_KEY=""

echo
echo_repeat "*" 88
echo "The created resources will now be deleted."
echo_repeat "*" 88
echo
```



```

clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    result=1
fi

return $result
}

```

Die in diesem Szenario verwendeten IAM Funktionen.

```

#####
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
# (IAM) user already exists.
#
# Parameters:
#     $1 - The name of the IAM user to check.
#
# Returns:
#     0 - If the user already exists.
#     1 - If the user doesn't exist.
#####
function iam_user_exists() {
    local user_name
    user_name=$1

    # Check whether the IAM user already exists.
    # We suppress all output - we're interested only in the return code.

    local errors
    errors=$(aws iam get-user \
        --user-name "$user_name" 2>&1 >/dev/null)

    local error_code=${?}

    if [[ $error_code -eq 0 ]]; then
        return 0 # 0 in Bash script means true.
    fi
}

```

```

else
  if [[ $errors != *"error"*(NoSuchEntity)* ]]; then
    aws_cli_error_log $error_code
    errecho "Error calling iam get-user $errors"
  fi

  return 1 # 1 in Bash script means false.
fi
}

#####
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#   -u user_name  -- The name of the user to create.
#
# Returns:
#   The ARN of the user.
#   And:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_create_user() {
  local user_name response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user"
    echo "Creates an WS Identity and Access Management (IAM) user. You must supply a
username:"
    echo "  -u user_name    The name of the user. It must be unique within the
account."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopt "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)

```

```
        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    User name:  $user_name"
iecho ""

# If the user already exists, we don't want to try to create it.
if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
fi

response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
fi

echo "$response"

return 0
}
```

```
#####
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#     -u user_name -- The name of the IAM user.
#     [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#     [access_key_id access_key_secret]
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_user_access_key() {
    local user_name file_name response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name] Optional file name for the access key output."
    echo ""
}

# Retrieve the calling parameters.
while getopt "u:f:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        f) file_name="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac

```

```

done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
fi

if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
fi

local key_id key_secret
# shellcheck disable=SC2086
key_id=$(echo $response | cut -f 2 -d ' ')
# shellcheck disable=SC2086
key_secret=$(echo $response | cut -f 4 -d ' ')

echo "$key_id $key_secret"

return 0
}

#####
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_json -- The assume role policy document.

```

```
#
# Returns:
#     The ARN of the role.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_role() {
    local role_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user_access_key"
        echo "Creates an AWS Identity and Access Management (IAM) role."
        echo "  -n role_name    The name of the IAM role."
        echo "  -p policy_json -- The assume role policy document."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_document="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$role_name" ]]; then
        errecho "ERROR: You must provide a role name with the -n parameter."
        usage
        return 1
    fi
}
```

```

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#     -n policy_name -- The name of the IAM policy.
#     -p policy_json -- The policy document.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_policy() {
    local policy_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {

```

```
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name  The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) policy_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
  errecho "ERROR: You must provide a policy name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-policy \
  --policy-name "$policy_name" \
  --policy-document "$policy_document" \
  --output text \
  --query Policy.Arn)

local error_code=${?}
```



```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
fi

echo "$response"
}

#####
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_attach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_attach_role_policy"
        echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
        echo "  -n role_name    The name of the IAM role."
        echo "  -p policy_ARN -- The IAM policy document ARN."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_arn="${OPTARG}" ;;
            h)
                usage

```

```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a role.

```

```

#
# Parameters:
#   -n role_name -- The name of the IAM role.
#   -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_detach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_detach_role_policy"
        echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
        echo "  -n role_name  The name of the IAM role."
        echo "  -p policy_ARN -- The IAM policy document ARN."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_arn="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$role_name" ]]; then
        errecho "ERROR: You must provide a role name with the -n parameter."
        usage
    fi
}

```

```

    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#     -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_policy() {
    local policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {

```

```
    echo "function iam_delete_policy"
    echo "Deletes an WS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy arn with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "  Policy arn: $policy_arn"
iecho ""

response=$(aws iam delete-policy \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
  return 1
fi
```

```
iecho "delete-policy response:$response"
iecho

return 0
}

#####
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_role() {
    local role_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_role"
        echo "Deletes an WS Identity and Access Management (IAM) role"
        echo " -n role_name -- The name of the IAM role."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}
```

```
done
export OPTIND=1

echo "role_name:$role_name"
if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    Role name: $role_name"
iecho ""

response=$(aws iam delete-role \
    --role-name "$role_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
fi

iecho "delete-role response:$response"
iecho

return 0
}

#####
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#     -u user_name -- The name of the user.
#     -k access_key -- The access key to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
```

```
function iam_delete_access_key() {
    local user_name access_key response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_access_key"
        echo "Deletes an WS Identity and Access Management (IAM) access key for the
specified IAM user"
        echo "  -u user_name    The name of the user."
        echo "  -k access_key    The access key to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:k:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            k) access_key="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$user_name" ]]; then
        errecho "ERROR: You must provide a username with the -u parameter."
        usage
        return 1
    fi

    if [[ -z "$access_key" ]]; then
        errecho "ERROR: You must provide an access key with the -k parameter."
        usage
        return 1
    fi
}
```



```

iecho "Parameters:\n"
iecho "  Username:  $user_name"
iecho "  Access key: $access_key"
iecho ""

response=$(aws iam delete-access-key \
  --user-name "$user_name" \
  --access-key-id "$access_key")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
  return 1
fi

iecho "delete-access-key response:$response"
iecho

return 0
}

#####
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#   -u user_name  -- The name of the user to create.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an WS Identity and Access Management (IAM) user. You must supply a
username:"

```

```
    echo "  -u user_name    The name of the user."
    echo ""
}

# Retrieve the calling parameters.
while getopts "u:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  User name:  $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
fi

response=$(aws iam delete-user \
  --user-name "$user_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
```

```
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
fi

iecho "delete-user response:$response"
iecho

return 0
}
```

- API-Einheiten finden Sie in den folgenden Themen in der AWS CLI Befehlsreferenz.
 - [AttachRolePolicy](#)
 - [CreateAccessKey](#)
 - [CreatePolicy](#)
 - [CreateRole](#)
 - [CreateUser](#)
 - [DeleteAccessKey](#)
 - [DeletePolicy](#)
 - [DeleteRole](#)
 - [DeleteUser](#)
 - [DeleteUserPolicy](#)
 - [DetachRolePolicy](#)
 - [PutUserPolicy](#)

Aktionen

AttachRolePolicy

Das folgende Codebeispiel zeigt, wie Sie es verwenden `AttachRolePolicy`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_attach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
    echo " -n role_name The name of the IAM role."
    echo " -p policy_ARN -- The IAM policy document ARN."
    echo ""
}
```

```
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam attach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
  return 1
fi
```

```

    echo "$response"

    return 0
}

```

- API-Einheiten finden Sie [AttachRolePolicy](#) in der AWS CLI Befehlsreferenz.

CreateAccessKey

Das folgende Codebeispiel zeigt, wie man es benutzt `CreateAccessKey`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#     -u user_name -- The name of the IAM user.
#     [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#     [access_key_id access_key_secret]
#     And:
#     0 - If successful.

```

```

#      1 - If it fails.
#####
function iam_create_user_access_key() {
    local user_name file_name response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]  Optional file name for the access key output."
    echo ""
}

# Retrieve the calling parameters.
while getopt "u:f:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        f) file_name="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

local error_code=${?}

```

```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
fi

if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
fi

local key_id key_secret
# shellcheck disable=SC2086
key_id=$(echo $response | cut -f 2 -d ' ')
# shellcheck disable=SC2086
key_secret=$(echo $response | cut -f 4 -d ' ')

echo "$key_id $key_secret"

return 0
}

```

- API-Einheiten finden Sie [CreateAccessKey](#) in der AWS CLI Befehlsreferenz.

CreatePolicy

Das folgende Codebeispiel zeigt, wie man es benutzt `CreatePolicy`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####

```



```

function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#     -n policy_name -- The name of the IAM policy.
#     -p policy_json -- The policy document.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_policy() {
    local policy_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_policy"
        echo "Creates an AWS Identity and Access Management (IAM) policy."
        echo "  -n policy_name  The name of the IAM policy."
        echo "  -p policy_json -- The policy document."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) policy_name="${OPTARG}" ;;
            p) policy_document="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}

```

```
    esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
fi

echo "$response"
}
```

- API-Einheiten finden Sie [CreatePolicy](#) in der AWS CLI Befehlsreferenz.

CreateRole

Das folgende Codebeispiel zeigt, wie man es benutzt `CreateRole`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_json -- The assume role policy document.
#
# Returns:
#     The ARN of the role.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_role() {
    local role_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name  The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
}
```

```
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_document="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
```

```

    return 1
fi

echo "$response"

return 0
}

```

- API-Einheiten finden Sie [CreateRole](#) in der AWS CLI Befehlsreferenz.

CreateUser

Das folgende Codebeispiel zeigt, wie man es benutzt `CreateUser`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```

```

}

#####
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#     -u user_name  -- The name of the user to create.
#
# Returns:
#     The ARN of the user.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_user() {
    local user_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user"
        echo "Creates an WS Identity and Access Management (IAM) user. You must supply a
username:"
        echo "  -u user_name    The name of the user. It must be unique within the
account."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}

```

```
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    User name:  $user_name"
iecho ""

# If the user already exists, we don't want to try to create it.
if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
fi

response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
fi

echo "$response"

return 0
}
```

- API-Einheiten finden Sie [CreateUser](#) in der AWS CLI Befehlsreferenz.

DeleteAccessKey

Das folgende Codebeispiel zeigt, wie man es benutzt `DeleteAccessKey`.

AWS CLI mit Bash-Skript

 Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#     -u user_name -- The name of the user.
#     -k access_key -- The access key to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_access_key() {
    local user_name access_key response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_access_key"
        echo "Deletes an WS Identity and Access Management (IAM) access key for the
specified IAM user"
        echo "  -u user_name    The name of the user."
        echo "  -k access_key   The access key to delete."
        echo ""
    }
}
```



```
}

# Retrieve the calling parameters.
while getopts "u:k:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    k) access_key="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

if [[ -z "$access_key" ]]; then
  errecho "ERROR: You must provide an access key with the -k parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "  Username:  $user_name"
iecho "  Access key: $access_key"
iecho ""

response=$(aws iam delete-access-key \
  --user-name "$user_name" \
  --access-key-id "$access_key")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```

aws_cli_error_log $error_code
errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
return 1
fi

iecho "delete-access-key response:$response"
iecho

return 0
}

```

- API-Einheiten finden Sie [DeleteAccessKey](#) in der AWS CLI Befehlsreferenz.

DeletePolicy

Das folgende Codebeispiel zeigt, wie man es benutzt DeletePolicy.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).

```

```
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#     -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_policy() {
    local policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_policy"
        echo "Deletes an WS Identity and Access Management (IAM) policy"
        echo "  -n policy_arn -- The name of the IAM policy arn."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) policy_arn="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done

```

```
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    Policy arn: $policy_arn"
iecho ""

response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
fi

iecho "delete-policy response:$response"
iecho

return 0
}
```

- API-Einheiten finden Sie [DeletePolicy](#) in der AWS CLI Befehlsreferenz.

DeleteRole

Das folgende Codebeispiel zeigt, wie man es benutzt `DeleteRole`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_role() {
    local role_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_role"
        echo "Deletes an WS Identity and Access Management (IAM) role"
        echo "  -n role_name -- The name of the IAM role."
        echo ""
    }
}
```

```
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

echo "role_name:$role_name"
if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "  Role name: $role_name"
iecho ""

response=$(aws iam delete-role \
  --role-name "$role_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-role operation failed.\n$response"
  return 1
fi

iecho "delete-role response:$response"
iecho
```

```

    return 0
}

```

- API-Einheiten finden Sie [DeleteRole](#) in der AWS CLI Befehlsreferenz.

DeleteUser

Das folgende Codebeispiel zeigt, wie man es benutzt `DeleteUser`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_user
#

```

```

# This function deletes the specified IAM user.
#
# Parameters:
#     -u user_name  -- The name of the user to create.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_user() {
    local user_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_user"
        echo "Deletes an WS Identity and Access Management (IAM) user. You must supply a
username:"
        echo "  -u user_name    The name of the user."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$user_name" ]]; then
        errecho "ERROR: You must provide a username with the -u parameter."
        usage
        return 1
    fi
}

```



```
iecho "Parameters:\n"
iecho "    User name:  $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
fi

response=$(aws iam delete-user \
    --user-name "$user_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
fi

iecho "delete-user response:$response"
iecho

return 0
}
```

- API-Einheiten finden Sie [DeleteUser](#) in der AWS CLI Befehlsreferenz.

DetachRolePolicy

Das folgende Codebeispiel zeigt, wie man es benutzt `DetachRolePolicy`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_detach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_detach_role_policy"
        echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
        echo "  -n role_name    The name of the IAM role."
        echo "  -p policy_ARN -- The IAM policy document ARN."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_arn="${OPTARG}" ;;
            h)
                usage

```

```
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}
```

- API-Einheiten finden Sie [DetachRolePolicy](#) in der AWS CLI Befehlsreferenz.

GetUser

Das folgende Codebeispiel zeigt, wie man es benutzt `GetUser`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
# (IAM) user already exists.
#
# Parameters:
#     $1 - The name of the IAM user to check.
#
# Returns:
#     0 - If the user already exists.
#     1 - If the user doesn't exist.
#####
function iam_user_exists() {
    local user_name
    user_name=$1

    # Check whether the IAM user already exists.
    # We suppress all output - we're interested only in the return code.

    local errors
    errors=$(aws iam get-user \
```

```

--user-name "$user_name" 2>&1 >/dev/null)

local error_code=${?}

if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
else
    if [[ $errors != *"error"*(NoSuchEntity)* ]]; then
        aws_cli_error_log $error_code
        errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
fi
}

```

- API-Einheiten finden Sie [GetUser](#) in der AWS CLI Befehlsreferenz.

ListAccessKeys

Das folgende Codebeispiel zeigt, wie man es benutzt `ListAccessKeys`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_list_access_keys

```

```
#
# This function lists the access keys for the specified user.
#
# Parameters:
#   -u user_name -- The name of the IAM user.
#
# Returns:
#   access_key_ids
#   And:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_list_access_keys() {

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_list_access_keys"
        echo "Lists the AWS Identity and Access Management (IAM) access key IDs for the
specified user."
        echo "  -u user_name  The name of the IAM user."
        echo ""
    }

    local user_name response
    local option OPTARG # Required to use getopt command in a function.
    # Retrieve the calling parameters.
    while getopt "u:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$user_name" ]]; then
        errecho "ERROR: You must provide a username with the -u parameter."
    fi
}
```

```

usage
return 1
fi

response=$(aws iam list-access-keys \
  --user-name "$user_name" \
  --output text \
  --query 'AccessKeyMetadata[].AccessKeyId')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports list-access-keys operation failed.$response"
  return 1
fi

echo "$response"

return 0
}

```

- API-Einheiten finden Sie [ListAccessKeys](#) in der AWS CLI Befehlsreferenz.

ListUsers

Das folgende Codebeispiel zeigt, wie man es benutzt `ListUsers`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####

```

```

function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_list_users
#
# List the IAM users in the account.
#
# Returns:
#     The list of users names
#     And:
#     0 - If the user already exists.
#     1 - If the user doesn't exist.
#####
function iam_list_users() {
    local option OPTARG # Required to use getopt command in a function.
    local error_code
    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_list_users"
        echo "Lists the AWS Identity and Access Management (IAM) user in the account."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "h" option; do
        case "${option}" in
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local response

    response=$(aws iam list-users \

```



```
--output text \  
--query "Users[].UserName")  
error_code=${?}  
  
if [[ $error_code -ne 0 ]]; then  
    aws_cli_error_log $error_code  
    errecho "ERROR: AWS reports list-users operation failed.$response"  
    return 1  
fi  
  
echo "$response"  
  
return 0  
}
```

- API-Einheiten finden Sie [ListUsers](#) in der AWS CLI Befehlsreferenz.

Amazon S3 S3-Beispiele für die Verwendung AWS CLI mit dem Bash-Skript

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie das AWS Command Line Interface with Bash-Skript mit Amazon S3 verwenden.

Basics sind Codebeispiele, die Ihnen zeigen, wie Sie die wesentlichen Operationen innerhalb eines Service ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Grundlagen](#)
- [Aktionen](#)

Grundlagen

Erlernen Sie die Grundlagen

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie einen Bucket und laden Sie eine Datei in ihn hoch.
- Laden Sie ein Objekt aus einem Bucket herunter.
- Kopieren Sie ein Objekt in einen Unterordner eines Buckets.
- Listen Sie die Objekte in einem Bucket auf.
- Löschen Sie die Bucket-Objekte und den Bucket.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#     0 - If successful.
#     1 - If an error occurred.
#####
function s3_getting_started() {
    {
        if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
            cd bucket-lifecycle-operations || exit

            source ./bucket_operations.sh
            cd ..
        fi
    }

    echo_repeat "*" 88
}
```

```
echo "Welcome to the Amazon S3 getting started demo."
echo_repeat "*" 88
  echo "A unique bucket will be created by appending a Universally Unique
Identifier to a bucket name prefix."
  echo -n "Enter a prefix for the S3 bucket that will be used in this demo: "
  get_input
  bucket_name_prefix=$get_input_result
local bucket_name
bucket_name=$(generate_random_name "$bucket_name_prefix")

local region_code
region_code=$(aws configure get region)

if create_bucket -b "$bucket_name" -r "$region_code"; then
  echo "Created demo bucket named $bucket_name"
else
  errecho "The bucket failed to create. This demo will exit."
  return 1
fi

local file_name
while [ -z "$file_name" ]; do
  echo -n "Enter a file you want to upload to your bucket: "
  get_input
  file_name=$get_input_result

  if [ ! -f "$file_name" ]; then
    echo "Could not find file $file_name. Are you sure it exists?"
    file_name=""
  fi
done

local key
key="$(basename "$file_name")"

local result=0
if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
  echo "Uploaded file $file_name into bucket $bucket_name with key $key."
else
  result=1
fi

local destination_file
destination_file="$file_name.download"
```

```
    if yes_no_input "Would you like to download $key to the file $destination_file?
(y/n) "; then
        if download_object_from_bucket "$bucket_name" "$destination_file" "$key"; then
            echo "Downloaded $key in the bucket $bucket_name to the file
$destination_file."
        else
            result=1
        fi
    fi

    if yes_no_input "Would you like to copy $key a new object key in your bucket? (y/
n) "; then
        local to_key
        to_key="demo/$key"
        if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
            echo "Copied $key in the bucket $bucket_name to the $to_key."
        else
            result=1
        fi
    fi

    local bucket_items
    bucket_items=$(list_items_in_bucket "$bucket_name")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        result=1
    fi

    echo "Your bucket contains the following items."
    echo -e "Name\t\tSize"
    echo "$bucket_items"

    if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
(y/n) "; then
        bucket_items=$(echo "$bucket_items" | cut -f 1)

        if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
            echo "The following items were deleted from the bucket $bucket_name"
            echo "$bucket_items"
        else
            result=1
        fi
    fi
```

```

    if delete_bucket "$bucket_name"; then
        echo "Deleted the bucket $bucket_name"
    else
        result=1
    fi
fi

return $result
}

```

Die in diesem Szenario verwendeten Amazon S3 S3-Funktionen.

```

#####
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name  -- The name of the bucket to create.
#     -r region_code  -- The code for an AWS Region in which to
#                       create the bucket.
#
# Returns:
#     The URL of the bucket that was created.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo "  -b bucket_name    The name of the bucket. It must be globally unique."
        echo "  [-r region_code]  The code for an AWS Region in which the bucket is
created."
        echo ""
    }
}

```

```
# Retrieve the calling parameters.
while getopts "b:r:h" option; do
  case "${option}" in
    b) bucket_name="${OPTARG}" ;;
    r) region_code="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done

if [[ -z "$bucket_name" ]]; then
  errecho "ERROR: You must provide a bucket name with the -b parameter."
  usage
  return 1
fi

local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
  bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "   Bucket name:   $bucket_name"
iecho "   Region code:   $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
  errecho "ERROR: A bucket with that name already exists. Try again."
  return 1
fi

# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
```

```

    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_file_to_bucket() {
    local response bucket_name source_file destination_file_name
    bucket_name=$1
    source_file=$2
    destination_file_name=$3

    response=$(aws s3api put-object \
        --bucket "$bucket_name" \
        --body "$source_file" \
        --key "$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
fi
}

#####
# function download_object_from_bucket

```

```

#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
    local response

    response=$(aws s3api get-object \
        --bucket "$bucket_name" \
        --key "$object_name" \
        "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}

#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####

```



```
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \
        --copy-source "$bucket_name/$source_key" \
        --key "$destination_key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
        return 1
    fi
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function list_items_in_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api list-objects \
        --bucket "$bucket_name" \
        --output text \
        --query 'Contents[].{Key: Key, Size: Size}')

    # shellcheck disable=SC2181
```

```

if [[ ${?} -eq 0 ]]; then
    echo "$response"
else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
fi
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1
    local keys=$2
    local response

    # Create the JSON for the items to delete.
    local delete_items
    delete_items="{\"Objects\":["
    for key in $keys; do
        delete_items="$delete_items{\"Key\": \"$key\"},"
    done
    delete_items=${delete_items%?} # Remove the final comma.
    delete_items="$delete_items]"

    response=$(aws s3api delete-objects \
        --bucket "$bucket_name" \
        --delete "$delete_items")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n$response"
        return 1
    fi
}

```

```

}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api delete-bucket \
        --bucket "$bucket_name")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
        return 1
    fi
}

```

- API-Einheiten finden Sie in den folgenden Themen in der AWS CLI Befehlsreferenz.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Aktionen

CopyObject

Das folgende Codebeispiel zeigt die Verwendung `CopyObject`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response
```

```

response=$(aws s3api copy-object \
  --bucket "$bucket_name" \
  --copy-source "$bucket_name/$source_key" \
  --key "$destination_key")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
  errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
  return 1
fi
}

```

- API-Einheiten finden Sie [CopyObject](#) in der AWS CLI Befehlsreferenz.

CreateBucket

Das folgende Codebeispiel zeigt die Verwendung `CreateBucket`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#####
# function errecho
#

```

```

# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name -- The name of the bucket to create.
#     -r region_code -- The code for an AWS Region in which to
#                       create the bucket.
#
# Returns:
#     The URL of the bucket that was created.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo "  -b bucket_name    The name of the bucket. It must be globally unique."
        echo "  [-r region_code]  The code for an AWS Region in which the bucket is
created."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "b:r:h" option; do
        case "${option}" in
            b) bucket_name="${OPTARG}" ;;
            r) region_code="${OPTARG}" ;;
            h)
                usage

```

```
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done

if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
fi

local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "    Bucket name:    $bucket_name"
iecho "    Region code:    $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi

# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
```

```
}

```

- API-Einheiten finden Sie [CreateBucket](#) in der AWS CLI Befehlsreferenz.

DeleteBucket

Das folgende Codebeispiel zeigt die Verwendung `DeleteBucket`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response

```



```

response=$(aws s3api delete-bucket \
  --bucket "$bucket_name")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
  errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
  return 1
fi
}

```

- API-Einheiten finden Sie [DeleteBucket](#) in der AWS CLI Befehlsreferenz.

DeleteObject

Das folgende Codebeispiel zeigt die Verwendung `DeleteObject`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#   $1 - The name of the bucket.

```

```
#      $2 - The key (file name) in the bucket to delete.

# Returns:
#      0 - If successful.
#      1 - If it fails.
#####
function delete_item_in_bucket() {
    local bucket_name=$1
    local key=$2
    local response

    response=$(aws s3api delete-object \
        --bucket "$bucket_name" \
        --key "$key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n$response"
        return 1
    fi
}
}
```

- API-Einheiten finden Sie [DeleteObject](#) in der AWS CLI Befehlsreferenz.

DeleteObjects

Das folgende Codebeispiel zeigt die Verwendung `DeleteObjects`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
```

```

function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1
    local keys=$2
    local response

    # Create the JSON for the items to delete.
    local delete_items
    delete_items="{\"Objects\":["
    for key in $keys; do
        delete_items="$delete_items{\"Key\": \"$key\"},"
    done
    delete_items=${delete_items%?} # Remove the final comma.
    delete_items="$delete_items]"

    response=$(aws s3api delete-objects \
        --bucket "$bucket_name" \
        --delete "$delete_items")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n$response"
        return 1
    fi
}

```

- API-Einheiten finden Sie [DeleteObjects](#) in der AWS CLI Befehlsreferenz.

GetObject

Das folgende Codebeispiel zeigt die Verwendung `GetObject`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
    local response

    response=$(aws s3api get-object \
        --bucket "$bucket_name" \
```

```

--key "$object_name" \
"$destination_file_name")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
fi
}

```

- API-Einheiten finden Sie [GetObject](#) in der AWS CLI Befehlsreferenz.

HeadBucket

Das folgende Codebeispiel zeigt die Verwendung `HeadBucket`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

#####
# function bucket_exists
#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#     $1 - The name of the bucket to check.
#
# Returns:
#     0 - If the bucket already exists.
#     1 - If the bucket doesn't exist.
#####
function bucket_exists() {
    local bucket_name
    bucket_name=$1

    # Check whether the bucket already exists.

```

```
# We suppress all output - we're interested only in the return code.

if aws s3api head-bucket \
  --bucket "$bucket_name" \
  >/dev/null 2>&1; then
  return 0 # 0 in Bash script means true.
else
  return 1 # 1 in Bash script means false.
fi
}
```

- API-Einheiten finden Sie [HeadBucket](#) in der AWS CLI Befehlsreferenz.

ListObjectsV2

Das folgende Codebeispiel zeigt die Verwendung `ListObjectsV2`.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
```

```

#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function list_items_in_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api list-objects \
        --bucket "$bucket_name" \
        --output text \
        --query 'Contents[].{Key: Key, Size: Size}')

    # shellcheck disable=SC2181
    if [[ ${?} -eq 0 ]]; then
        echo "$response"
    else
        errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
        return 1
    fi
}

```

- API-Einheiten finden Sie unter [ListObjectsV2](#) in der AWS CLI Befehlsreferenz.

PutObject

Das folgende Codebeispiel zeigt die Verwendung PutObject.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_file_to_bucket() {
    local response bucket_name source_file destination_file_name
    bucket_name=$1
    source_file=$2
    destination_file_name=$3

    response=$(aws s3api put-object \
        --bucket "$bucket_name" \
        --body "$source_file" \
        --key "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}

```

- API-Einheiten finden Sie [PutObject](#) in der AWS CLI Befehlsreferenz.

AWS STS Beispiele für die Verwendung AWS CLI mit dem Bash-Skript

Die folgenden Codebeispiele zeigen Ihnen, wie Sie Aktionen ausführen und allgemeine Szenarien implementieren, indem Sie das with Bash-Skript AWS Command Line Interface mit verwenden. AWS STS

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Aktionen zeigen Ihnen zwar, wie Sie einzelne Servicefunktionen aufrufen, aber Sie können Aktionen im Kontext der zugehörigen Szenarien sehen.

Jedes Beispiel enthält einen Link zum vollständigen Quellcode, in dem Sie Anweisungen zum Einrichten und Ausführen des Codes im Kontext finden.

Themen

- [Aktionen](#)

Aktionen

AssumeRole

Das folgende Codebeispiel zeigt die VerwendungAssumeRole.

AWS CLI mit Bash-Skript

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
#####  
# function iecho  
#  
# This function enables the script to display the specified text only if  
# the global variable $VERBOSE is set to true.  
#####  
function iecho() {  
    if [[ $VERBOSE == true ]]; then  
        echo "$@"  
    fi  
}
```

```

}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
# credentials.
#
# Parameters:
#     -n role_session_name -- The name of the session.
#     -r role_arn -- The ARN of the role to assume.
#
# Returns:
#     [access_key_id, secret_access_key, session_token]
#     And:
#     0 - If successful.
#     1 - If an error occurred.
#####
function sts_assume_role() {
    local role_session_name role_arn response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function sts_assume_role"
    echo "Assumes a role in the AWS account and returns the temporary credentials:"
    echo "  -n role_session_name -- The name of the session."
    echo "  -r role_arn -- The ARN of the role to assume."
    echo ""
}

while getopt n:r:h option; do
    case "${option}" in
        n) role_session_name=${OPTARG} ;;
        r) role_arn=${OPTARG} ;;
    esac
done

```

```
h)
  usage
  return 0
  ;;
\?)
  ech o"Invalid parameter"
  usage
  return 1
  ;;
esac
done

response=$(aws sts assume-role \
  --role-session-name "$role_session_name" \
  --role-arn "$role_arn" \
  --output text \
  --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-role operation failed.\n$response"
  return 1
fi

echo "$response"

return 0
}
```

- API-Einheiten finden Sie [AssumeRole](#) in der AWS CLI Befehlsreferenz.

Sicherheit in der AWS CLI

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud.

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS Command Line Interface, finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#) .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung anwenden können, wenn Sie das AWS Command Line Interface (AWS CLI) verwenden. In den folgenden Themen erfahren Sie, wie Sie den konfigurieren AWS CLI , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie das verwenden können AWS CLI , um Ihre AWS Ressourcen zu überwachen und zu schützen.

Themen

- [Datenschutz in der AWS CLI](#)
- [Identitäts- und Zugriffsverwaltung](#)
- [Konformitätsvalidierung für dieses AWS Produkt oder diese Dienstleistung](#)
- [Resilienz für dieses AWS Produkt oder diese Dienstleistung](#)
- [Infrastruktursicherheit für dieses AWS Produkt oder diese Dienstleistung](#)
- [Durchsetzung einer Mindestversion von TLS für AWS CLI](#)

Datenschutz in der AWS CLI

Das [Modell der AWS gemeinsamen Verantwortung](#) und geteilter Verantwortung gilt für den Datenschutz in AWS Command Line Interface. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie im [Abschnitt Datenschutz FAQ](#). Informationen zum Datenschutz in Europa finden Sie im [AWS Shared Responsibility Model und](#) im GDPR Blogbeitrag auf dem AWS Security Blog.

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto eine Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Einrichtung API und Protokollierung von Benutzeraktivitäten mit AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie FIPS 140-3 validierte kryptografische Module für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine benötigen API, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten finden Sie unter [Federal Information Processing Standard](#) () 140-3. FIPS

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole arbeiten AWS CLI oder sie anderweitig AWS-Services verwenden, API, AWS CLI oder. AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn

Sie einem externen Server eine URL zur Verfügung stellen, empfehlen wir dringend, dass Sie keine Anmeldeinformationen in den angebenURL, um Ihre Anfrage an diesen Server zu überprüfen.

Datenverschlüsselung

Ein wesentliches Merkmal eines sicheren Service ist, dass Informationen verschlüsselt werden, wenn sie nicht aktiv verwendet werden.

Verschlüsselung im Ruhezustand

Der AWS CLI speichert selbst keine anderen Kundendaten als die Anmeldeinformationen, die er benötigt, um im Namen des Benutzers mit den AWS Diensten zu interagieren.

Wenn Sie den verwenden, AWS CLI um einen AWS Dienst aufzurufen, der Kundendaten zur Speicherung auf Ihren lokalen Computer überträgt, finden Sie im Kapitel Sicherheit und Compliance im Benutzerhandbuch dieses Dienstes Informationen darüber, wie diese Daten gespeichert, geschützt und verschlüsselt werden.

Verschlüsselung während der Übertragung

Standardmäßig werden alle Daten, die vom Client-Computer übertragen werden, auf dem die Endpunkte AWS CLI und die AWS Dienstendpunkte ausgeführt werden, verschlüsselt, indem alles über eine HTTPS TLS /-Verbindung gesendet wird.

Sie müssen nichts tun, um die Verwendung vonHTTPS/TLSzu aktivieren. Sie ist immer aktiviert, es sei denn, Sie deaktivieren sie explizit für einen einzelnen Befehl mithilfe der `--no-verify-ssl`-Befehlszeilenoption.

Identitäts- und Zugriffsverwaltung

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um AWS Ressourcen zu verwenden. IAMist eine AWS-Service , die Sie ohne zusätzliche Kosten verwenden können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)

- [Wie AWS-Services arbeiten Sie mit IAM](#)
- [Fehlerbehebung bei AWS Identität und Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie arbeiten AWS.

Dienstbenutzer — Wenn Sie dies AWS-Services für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Wenn Sie für Ihre Arbeit mehr AWS Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Falls Sie auf eine Funktion in der von Ihnen verwendeten Version nicht zugreifen können AWS, finden [Fehlerbehebung bei AWS Identität und Zugriff](#) Sie weitere Informationen in der Bedienungsanleitung des AWS-Service Geräts, das Sie verwenden.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die AWS Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf AWS. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehenIAM. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit verwenden kann AWS, finden Sie in der Bedienungsanleitung der von AWS-Service Ihnen verwendeten.

IAMAdministrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff darauf zu verwalten AWS. Beispiele für AWS identitätsbasierte Richtlinien, die Sie verwenden könnenIAM, finden Sie im Benutzerhandbuch der von AWS-Service Ihnen verwendeten.

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAMIdentity Center-)

Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie AWS mithilfe eines Verbunds darauf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAMBenutzerhandbuch unter [AWS Signature Version 4 für API Anfragen](#).

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center Benutzerhandbuch und [AWS Multi-Faktor-Authentifizierung IAM im](#) IAM Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich](#) sind. IAM

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAMBenutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind](#).

Eine [IAMGruppe](#) ist eine Identität, die eine Sammlung von IAM Benutzern spezifiziert. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie im Benutzerhandbuch unter [Anwendungsfälle für IAM IAM Benutzer](#).

IAMRollen

Eine [IAMRolle](#) ist eine Identität innerhalb von Ihrem AWS-Konto, für die bestimmte Berechtigungen gelten. Sie ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Um vorübergehend eine IAM Rolle in der zu übernehmen AWS Management Console, können Sie [von einem Benutzer zu einer IAM Rolle \(Konsole\) wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwenden. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie unter [Methoden zur Übernahme einer Rolle](#) im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle für einen externen Identitätsanbieter](#). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM Benutzerberechtigungen** — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- **Kontoübergreifender Zugriff** — Sie können eine IAM Rolle verwenden, um einer Person (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie [IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM
- **Serviceübergreifender Zugriff** — Einige AWS-Services verwenden Funktionen in anderen. AWS-Services Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.

- Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der an aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- Servicerolle — Eine Servicerolle ist eine [IAM-Rolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschen IAM. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine](#).
- Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den

Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter [Überblick über JSON Richtlinien](#).

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAM Richtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus dem AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie im Benutzerhandbuch unter [Definieren benutzerdefinierter IAM Berechtigungen mit vom Kunden verwalteten Richtlinien](#). IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können. AWS-Konto Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie [wählen können, finden Sie im IAMBenutzerhandbuch unter Wählen Sie zwischen verwalteten Richtlinien und Inline-Richtlinien](#).

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien nicht IAM in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3 und AWS WAF Amazon VPC sind Beispiele für Dienste, die Unterstützung bieten ACLs. Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAM Benutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAM Benutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).

- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Geräte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter [Richtlinien zur Servicesteuerung](#) im AWS Organizations Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Sitzungsrichtlinien](#).

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

Wie AWS-Services arbeiten Sie mit IAM

Einen allgemeinen Überblick darüber, wie die meisten IAM Funktionen AWS-Services funktionieren, finden Sie IAM im IAMBenutzerhandbuch unter [AWS Dienste, die mit funktionieren](#).

Informationen zur Verwendung AWS-Service einer IAM bestimmten Funktion finden Sie im Abschnitt Sicherheit im Benutzerhandbuch des jeweiligen Dienstes.

Fehlerbehebung bei AWS Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS und auftreten könnenIAM.

Themen

- [Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS Ressourcen ermöglichen](#)

Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM Benutzer versucht, die Konsole zu verwenden, um Details zu einer fiktiven `my-example-widget` Ressource anzuzeigen, aber nicht über die fiktiven `aws:GetWidget` Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `aws:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.


```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen AWS unterstützt werden, finden Sie unter [Wie AWS-Services arbeiten Sie mit IAM](#)
- Informationen darüber, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie [im IAM Benutzerhandbuch unter Gewähren des Zugriffs auf einen anderen IAMBenutzer AWS-Konto , dessen Eigentümer Sie sind](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAMBenutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie Zugriff über einen Identitätsverbund [gewähren, finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#). IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM

Konformitätsvalidierung für dieses AWS Produkt oder diese Dienstleistung

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt](#), finden Sie unter [Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen Anwendungen erstellen HIPAA können, die AWS für sie in Frage kommen.

Note

Nicht alle sind berechtigt AWS-Services . HIPAA Weitere Informationen finden Sie in der [Referenz für HIPAA qualifizierte Dienste](#).

- [AWS Ressourcen zur AWS](#) von Vorschriften — Diese Sammlung von Arbeitsmapen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien für Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zusammengefasst.

- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.
- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Dieses AWS Produkt oder dieser Service folgt dem [Modell der gemeinsamen Verantwortung](#) in Bezug auf die spezifischen Amazon Web Services (AWS) -Services, die es unterstützt. Informationen zur AWS Servicesicherheit finden Sie auf der [Seite mit der Dokumentation zur AWS Servicesicherheit](#) und den [AWS Services, für die das AWS Compliance-Programm zur Einhaltung der](#) Vorschriften zuständig ist.

Resilienz für dieses AWS Produkt oder diese Dienstleistung

Die AWS globale Infrastruktur basiert auf AWS-Regionen Availability Zones.

AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind.

Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Dieses AWS Produkt oder dieser Service folgt dem [Modell der gemeinsamen Verantwortung](#) in Bezug auf die spezifischen Amazon Web Services (AWS) -Services, die es unterstützt. Informationen zur AWS Servicesicherheit finden Sie auf der [Seite mit der Dokumentation zur AWS Servicesicherheit](#) und den [AWS Services, für die das AWS Compliance-Programm zur Einhaltung der](#) Vorschriften zuständig ist.

Infrastruktursicherheit für dieses AWS Produkt oder diese Dienstleistung

Dieses AWS Produkt oder dieser Dienst verwendet Managed Services und ist daher durch die AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API Aufrufe, um über das Netzwerk auf dieses AWS Produkt oder diesen Service zuzugreifen. Kunden müssen Folgendes unterstützen:

- Sicherheit auf Transportschicht (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit perfekter Vorwärtsgeheimhaltung (PFS) wie (Ephemeral Diffie-Hellman) oder DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem Prinzipal zugeordnet ist, signiert werden. IAM Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Dieses AWS Produkt oder dieser Service folgt dem [Modell der gemeinsamen Verantwortung](#) in Bezug auf die spezifischen Amazon Web Services (AWS) -Services, die es unterstützt. Informationen zur AWS Servicesicherheit finden Sie auf der [Seite mit der Dokumentation zur AWS Servicesicherheit](#) und den [AWS Services, für die das AWS Compliance-Programm zur Einhaltung der](#) Vorschriften zuständig ist.

Durchsetzung einer Mindestversion von TLS für AWS CLI

Bei der Verwendung von AWS Command Line Interface (AWS CLI) spielt das Transport Layer Security (TLS) -Protokoll eine entscheidende Rolle bei der Sicherung der Kommunikation zwischen AWS CLI und AWS-Services. Um die Sicherheit bei der Kommunikation mit AWS Diensten zu erhöhen, sollten Sie TLS 1.2 oder höher verwenden.

Das AWS CLI und AWS-Service kann Daten sicher austauschen, wobei das TLS Protokoll Verschlüsselung, Authentifizierung und Datenintegrität gewährleistet. Durch die Nutzung des TLS Protokolls AWS CLI wird sichergestellt, dass Ihre Interaktionen mit dem Unternehmen vor unbefugtem Zugriff und Datenschutzverletzungen geschützt AWS-Services sind, wodurch die allgemeine Sicherheit Ihres AWS Ökosystems verbessert wird.

Das [Modell der AWS gemeinsamen Verantwortung](#) gilt für den Datenschutz in. AWS Command Line Interface Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS-Services. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und AWS-Services die Verwaltungsaufgaben der von Ihnen verwendeten Geräte verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [the section called "Datenschutz"](#).

Um sicherzustellen, dass AWS CLI Version 1 keine TLS Version vor TLS 1.2 verwendet, müssen Sie möglicherweise Open SSL neu kompilieren, um dieses Minimum durchzusetzen, und dann Python neu kompilieren, um das neu erstellte Open zu verwenden. SSL

Themen

- [Ermitteln Ihrer derzeit unterstützten Protokolle](#)
- [Compile Open SSL und Python](#)

Ermitteln Ihrer derzeit unterstützten Protokolle

Erstellen Sie zunächst mit Open SSL ein selbstsigniertes Zertifikat, das für den Testserver und Python SDK verwendet werden soll.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Dann starten Sie einen Testserver mit OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

Erstellen Sie in einem neuen Terminalfenster eine virtuelle Umgebung und installieren Sie die SDK für Python.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Erstellen Sie ein neues Python-Skript mit dem Namen `check.py`, SDK das die zugrunde liegende HTTP Bibliothek verwendet.

```
$ import urllib3
URL = 'https://localhost:4433/'

http = urllib3.PoolManager(
    ca_certs='cert.pem',
    cert_reqs='CERT_REQUIRED',
)
r = http.request('GET', URL)
print(r.data.decode('utf-8'))
```

Führen Sie Ihr neues Skript aus.

```
$ python check.py
```

Damit werden Details über die hergestellte Verbindung angezeigt. Suchen Sie in der Ausgabe nach „Protokoll:“. Wenn die Ausgabe "TLSv1.2" oder höher ist, ist die SDK Standardeinstellung TLS v1.2 oder höher. Wenn es sich um eine frühere Version handelt, müssen Sie Open neu kompilieren SSL und Python neu kompilieren.

Selbst wenn Ihre Python-Installation standardmäßig TLS v1.2 oder höher ist, ist es dennoch möglich, dass Python auf eine Version vor v1.2 neu verhandelt, wenn der Server TLS v1.2 oder höher nicht unterstützt. TLS Um zu überprüfen, ob Python nicht automatisch auf frühere Versionen neu verhandelt, starten Sie den Testserver erneut mit Folgendem.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Wenn Sie eine frühere Version von Open verwenden, ist das Flag möglicherweise nicht verfügbar. `-no_tls_3` Wenn dies der Fall ist, entfernen Sie die Markierung, da die Version von Open, die Sie verwenden, TLS v1.3 nicht unterstützt. Führen Sie dann das Python-Skript aus.

```
$ python check.py
```

Wenn Ihre Python-Installation für Versionen vor TLS 1.2 nicht korrekt neu verhandelt, sollten Sie eine Fehlermeldung erhalten. SSL

```
$ urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='localhost', port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL: UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

Wenn Sie eine Verbindung herstellen können, müssen Sie Open SSL und Python neu kompilieren, um die Aushandlung von Protokollen vor Version 1.2 zu deaktivieren.

Compile Open SSL und Python

Um sicherzustellen, dass das SDK OR AWS CLI nicht für etwas vor TLS 1.2 aushandelt, müssen Sie Open SSL und Python neu kompilieren. Kopieren Sie dazu den folgenden Inhalt, um ein Skript zu erstellen und auszuführen.

```
#!/usr/bin/env bash
set -e

OPENSSL_VERSION="1.1.1d"
OPENSSL_PREFIX="/opt/openssl-with-min-tls1_2"
PYTHON_VERSION="3.8.1"
PYTHON_PREFIX="/opt/python-with-min-tls1_2"

curl -O "https://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz"
tar -xzf "openssl-$OPENSSL_VERSION.tar.gz"
cd openssl-$OPENSSL_VERSION
./config --prefix=$OPENSSL_PREFIX no-ssl3 no-tls1 no-tls1_1 no-shared
make > /dev/null
sudo make install_sw > /dev/null

cd /tmp
curl -O "https://www.python.org/ftp/python/$PYTHON_VERSION/Python-$PYTHON_VERSION.tgz"
```

```
tar -xzf "Python-$PYTHON_VERSION.tgz"  
cd Python-$PYTHON_VERSION  
./configure --prefix=$PYTHON_PREFIX --with-openssl=$OPENSSL_PREFIX --disable-shared > /  
dev/null  
make > /dev/null  
sudo make install > /dev/null
```

Dadurch wird eine Version von Python kompiliert, die ein statisch gelinktes Open hatSSL, das nicht automatisch etwas vor 1.2 aushandelt. TLS Dadurch wird auch Open SSL im `/opt/openssl-with-min-tls1_2` Verzeichnis und Python im `/opt/python-with-min-tls1_2` Verzeichnis installiert. Nachdem Sie dieses Skript ausgeführt haben, bestätigen Sie die Installation der neuen Version von Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Dadurch sollte Folgendes ausgedruckt werden.

```
$ Python 3.8.1
```

Um sicherzustellen, dass diese neue Version von Python keine Version vor TLS 1.2 aushandelt, führen Sie die Schritte zur [Ermitteln Ihrer derzeit unterstützten Protokolle](#) Verwendung der neu installierten Python-Version (d. h. `/opt/python-with-min-tls1_2/bin/python3`) erneut aus.

Behebung von Fehlern für den AWS CLI

In diesem Abschnitt werden häufige Fehler sowie Maßnahmen beschrieben, mit denen sich die Probleme beheben lassen. Wir empfehlen, zunächst die Maßnahmen der [allgemeinen Problembekämpfung](#) zu befolgen.

Inhalt

- [Allgemeine Fehlerbehebung, die Sie zuerst versuchen sollten](#)
 - [Überprüfen Sie die Formatierung Ihrer AWS CLI Befehle](#)
 - [Überprüfen Sie, ob AWS-Region Ihr AWS CLI Befehl verwendet](#)
 - [Sicherstellen, dass Sie eine aktuelle Version der AWS CLI ausführen](#)
 - [Verwenden der Option --debug](#)
 - [Aktivieren und überprüfen Sie die AWS CLI Befehlsverlaufsprotokolle](#)
 - [Vergewissern Sie sich, dass Ihr konfiguriert ist AWS CLI](#)
- [Fehler aufgrund eines nicht gefundenen Befehls](#)
- [Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück](#)
- [Der Befehl "aws --version" gibt nach der Deinstallation von eine Version zurück AWS CLI](#)
- [Hat einen Befehl mit einem unvollständigen Parameternamen AWS CLI verarbeitet](#)
- [Fehler aufgrund einer Zugriffsverweigerung](#)
- [Ungültige Anmeldeinformationen und Schlüsselfehler](#)
- [Fehler aufgrund einer nicht übereinstimmenden Signatur](#)
- [Fehler aufgrund nicht gefundener Windows-Konsole](#)
- [SSLZertifikatsfehler](#)
- [Ungültige JSON Fehler](#)
- [Weitere Ressourcen](#)

Allgemeine Fehlerbehebung, die Sie zuerst versuchen sollten

Wenn Sie einen Fehler erhalten oder auf ein Problem mit dem stoßen AWS CLI, empfehlen wir Ihnen die folgenden allgemeinen Tipps, um Ihnen bei der Behebung zu helfen.

[Zurück zum Seitenanfang](#)

Überprüfen Sie die Formatierung Ihrer AWS CLI Befehle

Wenn Sie eine Fehlermeldung erhalten, dass ein Befehl nicht vorhanden ist oder dass ein Parameter (`Parameter validation failed`) nicht erkannt wird, der laut der Dokumentation verfügbar ist, ist der Befehl möglicherweise falsch formatiert. Wir empfehlen, Folgendes zu überprüfen:

- Überprüfen Sie Ihren Befehl auf Rechtschreib- und Formatierungsfehler.
- Vergewissern Sie sich, dass alle [Anführungszeichen und Escapes für Ihr Terminal](#) in Ihrem Befehl korrekt sind.
- Generieren Sie ein [AWS CLI -Skeleton](#), um Ihre Befehlsstruktur zu bestätigen.
- Weitere Informationen finden Sie in der zusätzlichen [Problembehandlung für JSON Werte](#). JSON Wenn Sie Probleme mit der JSON Formatierung Ihrer Terminalverarbeitung haben, empfehlen wir Ihnen, die Anführungsregeln des Terminals zu umgehen, indem Sie [Blobs verwenden, um JSON Daten direkt an das zu übergeben](#). AWS CLI

Weitere Informationen darüber, wie ein bestimmter Befehl strukturiert sein sollte, finden Sie im [AWS CLI Referenzhandbuch AWS CLI](#)

[Zurück zum Seitenanfang](#)

Überprüfen Sie, ob AWS-Region Ihr AWS CLI Befehl verwendet

Note

Sie müssen eine angeben, AWS-Region wenn Sie die verwenden AWS CLI, entweder explizit oder indem Sie eine Standardregion festlegen. Eine Liste aller Regionen AWS-Regionen , die Sie angeben können, finden Sie unter [AWS Regionen und Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz. Bei den verwendeten AWS-Region Bezeichnungen AWS CLI handelt es sich um dieselben Namen wie in den Endpunkten AWS Management Console URLs und Diensten.

Fehler oder unerwartete Ergebnisse können auftreten, wenn eine für den von Ihnen angegebenen nicht AWS-Service verfügbar ist AWS-Region oder sich Ihre Ressourcen in einem anderen befinden. AWS-Region In der Reihenfolge ihrer Rangfolge AWS-Region wird das wie folgt festgelegt:

- Die Befehlszeilenoption `--region`

- Die [AWS_DEFAULT_REGION](#) Umgebungsvariable.
- Die [region](#) Profileinstellung.

Vergewissern Sie sich, dass Sie das Richtige AWS-Region für Ihre Ressourcen verwenden.

[Zurück zum Seitenanfang](#)

Sicherstellen, dass Sie eine aktuelle Version der AWS CLI ausführen

Wenn Sie eine Fehlermeldung erhalten, die darauf hinweist, dass ein Befehl nicht existiert oder dass ein Parameter nicht erkannt wird, der laut [AWS CLI Referenzhandbuch AWS CLI](#) verfügbar ist, überprüfen Sie zunächst, ob Ihr Befehl korrekt formatiert ist. Wenn die Formatierung korrekt ist, empfehlen wir, ein Upgrade auf die neueste Version der AWS CLI vorzunehmen. Aktualisierte Versionen von AWS CLI werden fast jeden Werktag veröffentlicht. In diesen neuen Versionen von werden neue AWS Dienste, Funktionen und Parameter eingeführt AWS CLI. Die einzige Möglichkeit, Zugriff auf diese neuen Services, Funktionen oder Parameter zu erhalten, besteht darin, ein Upgrade auf eine Version durchzuführen, die nach der erstmaligen Einführung dieses Elements veröffentlicht wurde.

Wie Sie Ihre Version von aktualisieren, AWS CLI hängt davon ab, wie Sie sie ursprünglich installiert haben, wie unter beschrieben [Installieren Sie das AWS CLI](#).

Wenn Sie eines der gebündelten Installationsprogramme verwendet haben, sollten Sie die vorhandene Installation entfernen und die neueste Version für Ihr Betriebssystem herunterladen und installieren.

[Zurück zum Seitenanfang](#)

Verwenden der Option **--debug**

Wenn der einen Fehler AWS CLI meldet, den Sie nicht sofort verstehen, oder zu Ergebnissen führt, die Sie nicht erwarten, können Sie weitere Informationen zu dem Fehler abrufen, indem Sie den Befehl mit der `--debug` Option erneut ausführen. Mit dieser Option gibt die AWS CLI Informationen zu den einzelnen Schritten aus, die zur Verarbeitung des Befehls erforderlich sind. Anhand der Informationen in der Ausgabe können Sie ermitteln, an welcher Stelle der Fehler auftritt und wo er seinen Ursprung hat.

Sie können die Ausgabe an eine Textdatei senden, um sie später zu überprüfen oder an den AWS Support zu senden, wenn Sie dazu aufgefordert werden.

Wenn Sie die Option `--debug` einfügen, umfassen die Informationen u. a.:

- Suche nach Anmeldeinformationen
- Analysieren der bereitgestellten Parameter
- Konstruieren der an Server gesendeten Anfrage AWS
- Der Inhalt der Anfrage wurde gesendet an AWS
- Inhalt der unformatierten Antwort
- Die formatierte Ausgabe

Es folgt ein Beispiel für einen Befehl, der mit und ohne die Option `--debug` ausgeführt wird.

```
$ aws iam list-groups --profile MyTestProfile
{
  "Groups": [
    {
      "Path": "/",
      "GroupName": "MyTestGroup",
      "GroupId": "AGPA0123456789EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
      "CreateDate": "2019-08-12T19:34:04Z"
    }
  ]
}
```

```
$ aws iam list-groups --profile MyTestProfile --debug
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.16.215 Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - Arguments entered to
CLI: ['iam', 'list-groups', '--debug']
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function add_scalar_parsers at 0x7fdf173161e0>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function register_uri_param_handler at 0x7fdf17dec400>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function inject_assume_role_provider_cache at
0x7fdf17da9378>
2019-08-12 12:36:18,307 - MainThread - botocore.credentials - DEBUG - Skipping
environment variable credential check because profile name was explicitly set.
2019-08-12 12:36:18,307 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function attach_history_handler at 0x7fdf173ed9d8>
```

```
2019-08-12 12:36:18,308 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
service-2.json
2019-08-12 12:36:18,317 - MainThread - botocore.hooks - DEBUG - Event building-command-
table.iam: calling handler <function add_waiters at 0x7fdf1731a840>
2019-08-12 12:36:18,320 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
waiters-2.json
2019-08-12 12:36:18,321 - MainThread - awscli.clidriver - DEBUG - OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
<awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
<awscli.arguments.CLIArgument object at 0x7fdf171b09b0>)])
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_streaming_output_arg at
0x7fdf17316510>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_cli_input_json at
0x7fdf17da9d90>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function unify_paging_params at
0x7fdf17328048>
2019-08-12 12:36:18,326 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
paginator-1.json
2019-08-12 12:36:18,326 - MainThread - awscli.customizations.paginate - DEBUG -
Modifying paging parameters for operation: ListGroups
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_generate_skeleton at
0x7fdf1737eae8>
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event
before-building-argument-table-parser.iam.list-groups: calling handler
<bound method OverrideRequiredArgsArgument.override_required_args of
<awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event
before-building-argument-table-parser.iam.list-groups: calling handler
<bound method GenerateCliSkeletonArgument.override_required_args of
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event operation-
args-parsed.iam.list-groups: calling handler functools.partial(<function
check_should_enable_pagination at 0x7fdf17328158>, ['marker', 'max-items'], {'max-
items': <awscli.arguments.CLIArgument object at 0x7fdf171b09b0>}, OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
<awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
```

```
<awscli.customizations.paginate.PageArgument object at 0x7fdf171c58d0>), ('cli-  
input-json', <awscli.customizations.cliinputjson.CliInputJSONArgument object at  
0x7fdf171b0a58>), ('starting-token', <awscli.customizations.paginate.PageArgument  
object at 0x7fdf171b0a20>), ('page-size', <awscli.customizations.paginate.PageArgument  
object at 0x7fdf171c5828>), ('generate-cli-skeleton',  
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at  
0x7fdf171c5978>]]))  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-  
arg.iam.list-groups.path-prefix: calling handler <awscli.paramfile.URIArgumentHandler  
object at 0x7fdf1725c978>  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-  
arg.iam.list-groups.marker: calling handler <awscli.paramfile.URIArgumentHandler object  
at 0x7fdf1725c978>  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-  
arg.iam.list-groups.max-items: calling handler <awscli.paramfile.URIArgumentHandler  
object at 0x7fdf1725c978>  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -  
Event load-cli-arg.iam.list-groups.cli-input-json: calling handler  
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -  
Event load-cli-arg.iam.list-groups.starting-token: calling handler  
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-  
arg.iam.list-groups.page-size: calling handler <awscli.paramfile.URIArgumentHandler  
object at 0x7fdf1725c978>  
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event  
load-cli-arg.iam.list-groups.generate-cli-skeleton: calling handler  
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>  
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG  
- Event calling-command.iam.list-groups: calling handler  
<bound method CliInputJSONArgument.add_to_call_parameters of  
<awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>  
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG -  
Event calling-command.iam.list-groups: calling handler <bound  
method GenerateCliSkeletonArgument.generate_json_skeleton of  
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at  
0x7fdf171c5978>>  
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for  
credentials via: assume-role  
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for  
credentials via: assume-role-with-web-identity  
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for  
credentials via: shared-credentials-file
```

```
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - INFO - Found credentials
in shared credentials file: ~/.aws/credentials
2019-08-12 12:36:18,330 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/endpoints.json
2019-08-12 12:36:18,334 - MainThread - botocore.hooks - DEBUG - Event choose-service-
name: calling handler <function handle_service_name_alias at 0x7fdf1898eb70>
2019-08-12 12:36:18,337 - MainThread - botocore.hooks - DEBUG - Event creating-client-
class.iam: calling handler <function add_generate_presigned_url at 0x7fdf18a028c8>
2019-08-12 12:36:18,337 - MainThread - botocore.regions - DEBUG - Using partition
endpoint for iam, us-west-2: aws-global
2019-08-12 12:36:18,337 - MainThread - botocore.args - DEBUG - The s3 config key is not
a dictionary type, ignoring its value of: None
2019-08-12 12:36:18,340 - MainThread - botocore.endpoint - DEBUG - Setting iam timeout
as (60, 60)
2019-08-12 12:36:18,341 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/_retry.json
2019-08-12 12:36:18,341 - MainThread - botocore.client - DEBUG - Registering retry
handlers for service: iam
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
parameter-build.iam.ListGroups: calling handler <function generate_idempotent_uuid at
0x7fdf189b10d0>
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
call.iam.ListGroups: calling handler <function inject_api_version_header_if_needed at
0x7fdf189b2a60>
2019-08-12 12:36:18,343 - MainThread - botocore.endpoint - DEBUG - Making
request for OperationModel(name=ListGroups) with params: {'url_path': '/',
'query_string': '', 'method': 'POST', 'headers': {'Content-Type': 'application/x-
www-form-urlencoded; charset=utf-8', 'User-Agent': 'aws-cli/1.16.215 Python/3.7.3
Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205'}, 'body': {'Action':
'ListGroups', 'Version': '2010-05-08'}, 'url': 'https://iam.amazonaws.com/',
'context': {'client_region': 'aws-global', 'client_config': <botoconfig.Config
object at 0x7fdf16e9a4a8>, 'has_streaming_input': False, 'auth_type': None}}
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event request-
created.iam.ListGroups: calling handler <bound method RequestSigner.handler of
<botoconfig.signers.RequestSigner object at 0x7fdf16e9a470>>
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event choose-
signer.iam.ListGroups: calling handler <function set_operation_specific_signer at
0x7fdf18996f28>
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - Calculating signature
using v4 auth.
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - CanonicalRequest:
POST
/
```

```

content-type:application/x-www-form-urlencoded; charset=utf-8
host:iam.amazonaws.com
x-amz-date:20190812T193618Z

content-type;host;x-amz-date
5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEEca69080e1e1f
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - StringToSign:
AWS4-HMAC-SHA256
20190812T193618Z
20190812/us-east-1/iam/aws4_request
ab7e367eEXAMPLE2769f178ea509978cf8bfa054874b3EXAMPLE8d043fab6cc9
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - Signature:
d85a0EXAMPLEeb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
2019-08-12 12:36:18,344 - MainThread - botocore.endpoint - DEBUG - Sending
http request: <AWSPreparedRequest stream_output=False, method=POST,
url=https://iam.amazonaws.com/, headers={'Content-Type': b'application/
x-www-form-urlencoded; charset=utf-8', 'User-Agent': b'aws-cli/1.16.215
Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205',
'X-Amz-Date': b'20190812T193618Z', 'Authorization': b'AWS4-HMAC-SHA256
Credential=AKIA01234567890EXAMPLE-east-1/iam/aws4_request, SignedHeaders=content-
type;host;x-amz-date, Signature=d85a07692aceb401EXAMPLEa1b18ad1ddf58a1a3ce7EXAMPLE',
'Content-Length': '36'}>
2019-08-12 12:36:18,344 - MainThread - urllib3.util.retry - DEBUG - Converted retries
value: False -> Retry(total=False, connect=None, read=None, redirect=0, status=None)
2019-08-12 12:36:18,344 - MainThread - urllib3.connectionpool - DEBUG - Starting new
HTTPS connection (1): iam.amazonaws.com:443
2019-08-12 12:36:18,664 - MainThread - urllib3.connectionpool - DEBUG - https://
iam.amazonaws.com:443 "POST / HTTP/1.1" 200 570
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response headers:
{'x-amzn-RequestId': '74c11606-bd38-11e9-9c82-559da0adb349', 'Content-Type': 'text/
xml', 'Content-Length': '570', 'Date': 'Mon, 12 Aug 2019 19:36:18 GMT'}
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response body:
b'<ListGroupResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">\n
<ListGroupResult>\n  <IsTruncated>>false</IsTruncated>\n  <Groups>\n
  <member>\n    <Path>/</Path>\n    <GroupName>MyTestGroup</GroupName>
\n    <Arn>arn:aws:iam::123456789012:group/MyTestGroup</Arn>\n
  <GroupId>AGPA1234567890EXAMPLE</GroupId>\n    <CreateDate>2019-08-12T19:34:04Z</
CreateDate>\n  </member>\n  </Groups>\n </ListGroupResult>\n
  <ResponseMetadata>\n    <RequestId>74c11606-bd38-11e9-9c82-559da0adb349</RequestId>\n
  </ResponseMetadata>\n</ListGroupResponse>\n'
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event needs-
retry.iam.ListGroups: calling handler <botocore.retryhandler.RetryHandler object at
0x7fdf16e9a780>
2019-08-12 12:36:18,665 - MainThread - botocore.retryhandler - DEBUG - No retry needed.

```

```
2019-08-12 12:36:18,665 - MainThread - boto3.hooks - DEBUG - Event after-
call.iam.ListGroups: calling handler <function json_decode_policies at 0x7fdf189b1d90>
{
  "Groups": [
    {
      "Path": "/",
      "GroupName": "MyTestGroup",
      "GroupId": "AGPA123456789012EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
      "CreateDate": "2019-08-12T19:34:04Z"
    }
  ]
}
```

[Zurück zum Seitenanfang](#)

Aktivieren und überprüfen Sie die AWS CLI Befehlsverlaufsprotokolle

Sie können die AWS CLI Befehlsverlaufsprotokolle mithilfe der [cli_history](#) Dateieinstellung aktivieren. Nach dem Aktivieren dieser Einstellung AWS CLI zeichnet das den Verlauf der aws Befehle auf.

Sie können Ihren Verlauf mit dem `aws history list`-Befehl auflisten und die resultierenden `command_ids` im `aws history show`-Befehl für Details verwenden. Weitere Informationen finden Sie unter [aws history](#) im AWS CLI -Referenzhandbuch.

Wenn Sie die Option `--debug` einfügen, umfassen die Informationen u. a.:

- APIAnrufe an Botocore
- Statuscodes
- HTTPAntworten
- Überschriften
- Rückgabecodes

Anhand dieser Informationen können Sie überprüfen, ob sich Parameterdaten und API Aufrufe erwartungsgemäß verhalten, und Sie können dann ableiten, an welchem Schritt im Prozess Ihr Befehl fehlschlägt.

[Zurück zum Seitenanfang](#)

Vergewissern Sie sich, dass Ihr konfiguriert ist AWS CLI

Verschiedene Fehler können auftreten, wenn Ihre `credentials` Dateien `config` und oder Ihr IAM Benutzer oder Ihre Rolle nicht richtig konfiguriert sind. Weitere Informationen zur Behebung von Fehlern mit `config` und `credentials` -Dateien oder Ihrem IAM Benutzer oder Ihren Rollen finden Sie unter [the section called “Fehler aufgrund einer Zugriffsverweigerung”](#) und [the section called “Ungültige Anmeldeinformationen und Schlüsselfehler”](#).

[Zurück zum Seitenanfang](#)

Fehler aufgrund eines nicht gefundenen Befehls

Dieser Fehler bedeutet, dass das Betriebssystem den AWS CLI Befehl nicht finden kann. Die Installation ist möglicherweise unvollständig oder muss aktualisiert werden.

Mögliche Ursache: Sie versuchen, eine neue AWS CLI Funktion als Ihre installierte Version zu verwenden, oder Sie haben eine falsche Formatierung

Beispiel-Fehlertext:

```
$ aws s3 copy
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                | website
cp                | mv
.....
```

Verschiedene Fehler können auftreten, wenn Ihr Befehl inkorrekt formatiert ist oder wenn Sie eine Version verwenden, die vor der Veröffentlichung der Funktion erstellt wurde. Weitere Informationen zur Behebung dieser beiden Arten von Problemen finden Sie unter [the section called “Überprüfen Sie die Formatierung Ihrer AWS CLI Befehle”](#) und [the section called “Sicherstellen, dass Sie eine aktuelle Version der AWS CLI ausführen”](#).

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Das Terminal muss nach der Installation neu gestartet werden.

Beispiel-Fehlertext:

```
$ aws --version  
command not found: aws
```

Wenn der `aws` Befehl nach der ersten Installation oder Aktualisierung von nicht gefunden werden kann AWS CLI, müssen Sie möglicherweise Ihr Terminal neu starten, damit es alle PATH Updates erkennt.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Der AWS CLI wurde nicht vollständig installiert

Beispiel-Fehlertext:

```
$ aws --version  
command not found: aws
```

Wenn der `aws` Befehl nach der ersten Installation oder Aktualisierung von nicht gefunden werden kann AWS CLI, wurde er möglicherweise nicht vollständig installiert. Versuchen Sie, sie erneut zu installieren, indem Sie die unter [Installieren Sie das AWS CLI](#) angegebenen Schritte für Ihre Plattform ausführen.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Der AWS CLI hat keine Berechtigungen (Linux)

Wenn der `aws` Befehl nach der ersten Installation oder Aktualisierung von AWS CLI unter Linux nicht gefunden werden kann, hat er möglicherweise keine execute Berechtigungen für den Ordner, in dem er installiert wurde. Führen Sie den folgenden Befehl zusammen mit Ihrer AWS CLI Installation aus, um [chmod](#) Berechtigungen für zu gewähren AWS CLI: PATH

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Die Umgebungsvariable **PATH** des Betriebssystems wurde während der Installation nicht aktualisiert.

Beispiel-Fehlertext:

```
$ aws --version
command not found: aws
```

Möglicherweise müssen Sie die ausführbare aws-Datei zur Umgebungsvariablen PATH des Betriebssystems hinzufügen. Verwenden Sie AWS CLI die folgenden Anweisungen für Ihr BetriebssystemPATH, um das zu Ihrem hinzuzufügen.

Linux and macOS

1. Suchen Sie das Profilskript der Shell in Ihrem Benutzerverzeichnis. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`
 - Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc` oder `.login`
2. Fügen Sie dem Profilskript einen Exportbefehl hinzu. Der folgende Befehl fügt Ihre lokale Ablage zur aktuellen PATH-Variablen hinzu.

```
export PATH=/usr/local/bin:$PATH
```

3. Laden Sie das hochgeladene Profil erneut in Ihre aktuelle Sitzung.

```
$ source ~/.bash_profile
```

Windows

1. Verwenden Sie in einer Windows-Eingabeaufforderung den `where`-Befehl mit dem Parameter `/R path`, um den Speicherort der aws-Datei zu finden. Die Ergebnisse geben alle Ordner zurück, die aws enthalten.

```
C:\> where /R c:\ aws
c:\Program Files\Amazon\AWSCLIV2\aws.exe
...
```

Standardmäßig befindet sich AWS CLI Version 2 in:

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. Betätigen Sie die Windows-Taste und geben Sie **environment variables** ein.
3. Wählen Sie aus der Liste der Vorschläge Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten) aus.
4. Wählen PATHS und dann Bearbeiten.
5. Fügen Sie den gefundenen Pfad in das Feld Variable value (Variablenwert) ein, z. B. **C:\Program Files\Amazon\AWSCLIV2\aws.exe**.
6. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
7. Schließen Sie alle laufenden Eingabeaufforderungen und öffnen Sie das Eingabeaufforderungsfenster erneut.

[Zurück zum Seitenanfang](#)

Der Befehl „aws --version“ gibt eine andere als die installierte Version zurück

Ihr Terminal gibt möglicherweise ein anderes PATH AWS CLI als erwartetes Ergebnis zurück.

Mögliche Ursache: Das Terminal muss nach der Installation neu gestartet werden.

Wenn der aws-Befehl die falsche Version anzeigt, müssen Sie möglicherweise Ihr Terminal neu starten, damit PATH-Aktualisierungen erkannt werden. Alle offenen Terminals müssen geschlossen werden, nicht nur Ihr aktives Terminal.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Das Terminal muss nach der Installation neu gestartet werden.

Wenn der aws-Befehl die falsche Version anzeigt, müssen Sie möglicherweise Ihr Terminal neu starten, damit PATH-Aktualisierungen erkannt werden.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Sie haben mehrere Versionen von AWS CLI

Wenn Sie das aktualisiert AWS CLI und eine andere Installationsmethode als Ihre bereits vorhandene Installation verwendet haben, können mehrere Versionen installiert werden. Beispiel:

Wenn Sie unter Linux oder macOS `pip` für die aktuelle Installation verwendet, aber versucht haben, die Aktualisierung mithilfe der `.pkg`-Installationsdatei auszuführen, könnte dies zu Konflikten führen – insbesondere, wenn `PATH` auf die alte Version verweist.

Um dies zu beheben, [deinstallieren Sie alle Versionen der AWS CLI](#) und nehmen Sie eine Neuinstallation vor.

Befolgen Sie nach der Deinstallation aller Versionen die Anweisungen für Ihr Betriebssystem, um die gewünschte Version der [AWS CLI Version 1](#) oder [AWS CLI Version 2](#) zu installieren.

Note

Wenn dies passiert, nachdem Sie AWS CLI Version 2 mit einer bereits vorhandenen Installation von AWS CLI Version 1 installiert haben, folgen Sie [bei der Migration von Version 1 den Migrationsanweisungen unter Installationsanweisungen](#). AWS CLI

[Zurück zum Seitenanfang](#)

Der Befehl "`aws --version`" gibt nach der Deinstallation von eine Version zurück AWS CLI

Dies tritt häufig auf, wenn noch irgendwo auf Ihrem System eine AWS CLI installiert ist.

Mögliche Ursache: Das Terminal muss nach der Deinstallation neu gestartet werden.

Wenn der Befehl `aws --version` weiterhin funktioniert, müssen Sie möglicherweise Ihr Terminal neu starten, damit Terminal-Aktualisierungen erkannt werden.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Sie haben mehrere Versionen von AWS CLI auf Ihrem System oder haben nicht dieselbe Deinstallationsmethode verwendet, mit der Sie das ursprünglich installiert haben AWS CLI

Sie werden AWS CLI möglicherweise nicht korrekt deinstalliert, wenn Sie sie AWS CLI mit einer anderen Methode deinstalliert haben, als Sie sie zur Installation verwendet haben, oder wenn Sie mehrere Versionen installiert haben. Wenn Sie beispielsweise `pip` für die aktuelle Installation verwendet haben, müssen Sie `pip` auch für die Deinstallation verwenden. Um dieses Problem zu beheben, deinstallieren Sie es AWS CLI mit derselben Methode, mit der Sie es installiert haben.

1. Befolgen Sie die Anweisungen für Ihr Betriebssystem und die ursprüngliche Installationsmethode, um die [AWS CLI Version 1](#) und die [AWS CLI Version 2](#) zu deinstallieren.
2. Schließen Sie alle offenen Terminals.
3. Öffnen Sie Ihr bevorzugtes Terminal, geben Sie den folgenden Befehl ein und vergewissern Sie sich, dass keine Version zurückgegeben wird.

```
$ aws --version  
command not found: aws
```

Wenn in der Ausgabe immer noch eine Version aufgeführt ist, AWS CLI wurde diese höchstwahrscheinlich mit einer anderen Methode installiert, oder es gibt mehrere Versionen. Wenn Sie nicht wissen, welche Methode Sie installiert haben AWS CLI, folgen Sie den Anweisungen für jede Deinstallationsmethode für die [AWS CLI Version 1](#) und [AWS CLI Version 2](#), die Ihrem Betriebssystem entsprechen, bis keine Versionsausgabe mehr eingeht.

Note

Wenn Sie einen Paket-Manager für die Installation der AWS CLI (pip, apt, brew usw.) verwendet haben, müssen Sie für die Deinstallation denselben Paket-Manager verwenden. Befolgen Sie unbedingt die Anweisungen des Paket-Managers zur Deinstallation aller Versionen eines Pakets.

[Zurück zum Seitenanfang](#)

Hat einen Befehl mit einem unvollständigen Parameternamen AWS CLI verarbeitet

Mögliche Ursache: Sie haben eine erkannte Abkürzung des Parameters AWS CLI verwendet.

Da der AWS CLI mit Python erstellt wurde, verwendet der die `argparse` Python-Bibliothek, einschließlich des [allow_abbrev](#) Arguments. Abkürzungen von Parametern werden von der erkannt AWS CLI und verarbeitet.

Das folgende [create-change-set](#) Befehlsbeispiel ändert den CloudFormation Stacknamen. Der Parameter `--change-set-n` wird als Abkürzung für `--change-set-name` erkannt und der Befehl wird von AWS CLI verarbeitet.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

Wenn sich Ihre Abkürzung auf mehrere Befehle beziehen könnte, wird der Parameter nicht als Abkürzung erkannt.

Das folgende [create-change-set](#) Befehlsbeispiel ändert den CloudFormation Stacknamen. Der Parameter `--change-set-` wird nicht als Abkürzung erkannt, da es mehrere Parameter gibt, für die er eine Abkürzung sein könnte, z. B. `--change-set-name` und `--change-set-type`. Daher verarbeitet der Befehl von AWS CLI nicht.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

Warning

Verwenden Sie nicht gezielt Parameterabkürzungen. Sie sind unzuverlässig und nicht abwärtskompatibel. Wenn einem Befehl neue Parameter hinzugefügt werden, die Ihre Abkürzungen durcheinanderbringen, werden Ihre Befehle nicht mehr funktionieren. Wenn es sich bei dem Parameter um ein Argument mit einem Wert handelt, kann es zudem zu unerwartetem Verhalten bei Ihren Befehlen kommen. Wenn mehrere Instanzen eines Arguments mit einem Wert übergeben werden, wird nur die letzte Instanz ausgeführt. Im folgenden Beispiel ist der Parameter `--filters` ein Argument mit einem Wert. Die Parameter `--filters` und `--filter` sind angegeben. Der Parameter `--filter` ist eine Abkürzung von `--filters`. Dies führt dazu, dass zwei Instanzen von `--filters` angewendet werden und nur das letzte `--filter`-Argument gilt.

```
$ aws ec2 describe-vpc-peering-connections \  
  --filters Name=tag:TagName,Values=VpcPeeringConnection \  
  --filter Name=status-code,Values=active
```

Vergewissern Sie sich, dass Sie gültige Parameter verwenden, bevor Sie einen Befehl ausführen, um unerwartetes Verhalten zu vermeiden.

[Zurück zum Seitenanfang](#)

Fehler aufgrund einer Zugriffsverweigerung

Mögliche Ursache: Die AWS CLI Programmdatei hat keine „Ausführen“-Rechte

Stellen Sie unter Linux oder macOS sicher, dass das aws-Programm über Ausführungsberechtigungen für den aufrufenden Benutzer verfügt. In der Regel sind die Berechtigungen auf 755 festgelegt.

Um Ihrem Benutzer die Ausführungsberechtigung hinzuzufügen, führen Sie den folgenden Befehl aus und ersetzen Sie `~/.local/bin/aws` mit dem Pfad zum Programm auf Ihrem Computer.

```
$ chmod +x ~/.local/bin/aws
```

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Ihre IAM Identität ist nicht berechtigt, den Vorgang auszuführen

Beispiel-Fehlertext:

```
$ aws s3 ls
An error occurred (AccessDenied) when calling the ListBuckets operation: Access
denied.
```

Wenn Sie einen AWS CLI Befehl ausführen, werden AWS Vorgänge in Ihrem Namen ausgeführt, wobei Anmeldeinformationen verwendet werden, die Sie einem IAM Konto oder einer Rolle zuordnen. Die beigefügten Richtlinien müssen Ihnen die Erlaubnis erteilen, die API Aktionen aufzurufen, die den Befehlen entsprechen, die Sie mit dem ausführen AWS CLI.

Die meisten Befehle rufen eine einzelne Aktion mit einem Namen auf, der dem Befehlsnamen entspricht. Benutzerdefinierte Befehle wie `aws s3 sync` Call Multiple sind jedoch möglich APIs. Mithilfe der `--debug` Option können Sie sehen, welcher APIs Befehl aufgerufen wird.

Wenn Sie sicher sind, dass der Benutzer oder die Rolle über die richtigen, per Richtlinie zugewiesenen Berechtigungen verfügt, stellen Sie sicher, dass Ihr AWS CLI Befehl die Anmeldeinformationen verwendet, die Sie erwarten. Im [nächsten Abschnitt über Anmeldeinformationen](#) können Sie überprüfen, ob AWS CLI es sich bei den verwendeten Anmeldeinformationen um die von Ihnen erwarteten Anmeldeinformationen handelt.

Informationen zum Zuweisen von IAM Berechtigungen finden Sie [im IAMBenutzerhandbuch unter Überblick über die Zugriffsverwaltung: Berechtigungen und Richtlinien](#).

[Zurück zum Seitenanfang](#)

Ungültige Anmeldeinformationen und Schlüsselfehler

Beispiel-Fehlertext:

```
$ aws s3 ls
An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS
Access Key Id
you provided does not exist in our records.
```

```
$ aws s3 ls
An error occurred (InvalidClientTokenId) when calling the ListBuckets operation: The
security token
included in the request is invalid.
```

Mögliche Ursache: Der AWS CLI liest falsche Anmeldeinformationen oder liest sie von einem unerwarteten Ort

AWS CLI Möglicherweise liest er Anmeldeinformationen von einem anderen Ort als erwartet, oder Ihre Schlüsselpaarinformationen sind falsch. Sie können `aws configure list` ausführen, um anzugeben, welche Anmeldeinformationen verwendet werden.

Das folgende Beispiel zeigt, wie Sie die Anmeldeinformationen prüfen, die für das Standardprofil verwendet werden.

```
$ aws configure list
      Name                               Value                               Type    Location
      ----                               -
      profile                             <not set>                           None    None
      access_key                          *****XYVA                          shared-credentials-file
      secret_key                           *****ZAGY                          shared-credentials-file
      region                                us-west-2                             config-file  ~/.aws/config
```

Das folgende Beispiel zeigt, wie Sie die Anmeldeinformationen prüfen, die für das benannte Profil verwendet werden.

```
$ aws configure list --profile saanvi
      Name                               Value                               Type    Location
      ----                               -
profile                               saanvi                             manual  --profile
access_key                             ***** shared-credentials-file
secret_key                               ***** shared-credentials-file
region                                  us-west-2                           config-file  ~/.aws/config
```

Bestätigen Sie die Schlüsselpaar-Informationen anhand Ihrer Dateien `config` und `credentials`. Weitere Informationen zu den Dateien `config` und `credentials` finden Sie unter [the section called “Einstellungen für die Konfiguration und die Anmeldeinformationsdatei in der AWS CLI”](#). Weitere Informationen zu Anmeldeinformationen und Authentifizierung finden Sie unter [Authentifizierung und Anmeldeinformationen](#).

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Die Uhr Ihres Computers ist nicht synchronisiert.

Wenn Sie gültige Anmeldeinformationen verwenden, ist möglicherweise die Uhr nicht synchronisiert. Führen Sie unter Linux oder macOS `date` aus, um die Zeit zu überprüfen.

```
$ date
```

Wenn Ihre Systemuhr nicht innerhalb weniger Minuten korrigiert wird, synchronisieren Sie diese mit `ntpdate`.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Verwenden Sie unter Windows die Optionen für Datum und Uhrzeit in der Systemsteuerung, um Ihre Systemuhr zu konfigurieren.

[Zurück zum Seitenanfang](#)

Fehler aufgrund einer nicht übereinstimmenden Signatur

Beispiel-Fehlertext:

```
$ aws s3 ls
```

```
An error occurred (SignatureDoesNotMatch) when calling the ListBuckets operation: The request signature we calculated does not match the signature you provided. Check your key and signing method.
```

Wenn der einen Befehl AWS CLI ausführt, sendet er eine verschlüsselte Anfrage an die AWS Server, um die entsprechenden AWS Dienstoperationen auszuführen. Ihre Anmeldeinformationen (der Zugriffsschlüssel und der geheime Schlüssel) sind an der Verschlüsselung beteiligt und AWS ermöglichen die Authentifizierung der Person, die die Anfrage stellt. Es gibt mehrere Dinge, die den korrekten Vorgang dieses Prozesses folgendermaßen beeinträchtigen können.

Mögliche Ursache: Ihre Uhr ist nicht mit den AWS Servern synchron

Zum Schutz vor [Replay-Angriffen](#) kann die aktuelle Zeit während der Verschlüsselung/Entschlüsselung verwendet werden. Wenn die Zeit von Client und Server um mehr als den zulässigen Wert voneinander abweicht, schlägt der Prozess möglicherweise fehl und die Anfrage wird abgelehnt. Dies kann auch der Fall sein, wenn Sie einen Befehl in einer virtuellen Maschine ausführen, deren Uhr nicht mit der Uhr des Hostcomputers synchronisiert ist. Eine mögliche Ursache besteht darin, dass die virtuelle Maschine in den Ruhezustand versetzt wird und es nach dem erneuten Aktivieren einige Zeit dauert, die Uhr mit dem Hostcomputer zu synchronisieren.

Führen Sie unter Linux oder macOS `date` aus, um die Zeit zu überprüfen.

```
$ date
```

Wenn Ihre Systemuhr nicht innerhalb weniger Minuten korrigiert wird, synchronisieren Sie diese mit `ntpdate`.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Verwenden Sie unter Windows die Optionen für Datum und Uhrzeit in der Systemsteuerung, um Ihre Systemuhr zu konfigurieren.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Ihr Betriebssystem verarbeitet AWS Schlüssel, die bestimmte Sonderzeichen enthalten, falsch

Wenn Ihre AWS Schlüssel bestimmte Sonderzeichen wie, oder enthalten - +/, verarbeiten einige Betriebssystemvarianten die Zeichenfolge nicht ordnungsgemäß und führen dazu, dass die Schlüsselzeichenfolge falsch interpretiert wird.

Wenn Sie Ihre Schlüssel mit anderen Tools oder Skripten verarbeiten, z. B. mit Tools, die die Anmeldeinformationsdatei bei deren Erstellung auf einer neuen Instanz erstellen, können diese Tools und Skripten Sonderzeichen auf eigene Weise verarbeiten, sodass sie in etwas umgewandelt werden, das nicht AWS mehr erkannt wird.

Wir empfehlen, den geheimen Schlüssel neu zu generieren, um einen Schlüssel zu erhalten, der das Sonderzeichen nicht enthält.

[Zurück zum Seitenanfang](#)

Fehler aufgrund nicht gefundener Windows-Konsole

Beispiel-Fehlertext:

```
$ aws s3 ls
No Windows console found. Are you running cmd.exe?
```

Wenn Sie einen AWS CLI Befehl verwenden, erhalten Sie die Meldung „Keine Windows-Konsole gefunden“. Führen Sie cmd.exe aus?“. Dies ist normalerweise ein Fehler, der bei AWS CLI Version 1 auftritt, wenn das von prompt_toolkit Ihnen installierte Python veraltet ist. Um dieses Problem zu beheben, installieren Sie eine aktuelle Version von prompt_toolkit auf der [Python-Website](#).

[Zurück zum Seitenanfang](#)

SSLZertifikatsfehler

Mögliche Ursache: Der vertraut dem Zertifikat Ihres Proxys AWS CLI nicht

Beispiel-Fehlertext:

```
$ aws s3 ls
```

```
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Wenn Sie einen AWS CLI Befehl verwenden, erhalten Sie eine `[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed` Fehlermeldung. Dies liegt daran, dass Sie dem Zertifikat Ihres Proxys AWS CLI nicht vertrauen, z. B. weil das Zertifikat Ihres Proxys selbst signiert ist und Ihr Unternehmen als Zertifizierungsstelle (CA) festgelegt wurde. Dadurch wird verhindert, dass das AWS CLI CA-Stammzertifikat Ihres Unternehmens in der lokalen CA-Registrierung gefunden wird.

Um dieses Problem zu beheben, geben Sie anhand der Einstellung der `.pem` [ca_bundle](#) Konfigurationsdatei, der `--ca-bundle` Befehlszeilenoption oder der [AWS_CA_BUNDLE](#) Umgebungsvariablen an, AWS CLI wo Ihre Unternehmensdatei zu finden ist.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Ihre Konfiguration zeigt nicht auf den richtigen Speicherort des CA-Stammzertifikats.

Beispiel-Fehlertext:

```
$ aws s3 ls
SSL validation failed for regionname [Errno 2] No such file or directory
```

Dies liegt daran, dass der Speicherort der Bundle-Datei Ihrer Zertifizierungsstelle (Certification Authority, CA) in der AWS CLI falsch konfiguriert ist. Überprüfen Sie zur Behebung dieses Fehlers, an welchem Speicherort sich die `.pem`-Datei Ihres Unternehmens befindet, und aktualisieren Sie die AWS CLI -Konfiguration mithilfe der Konfigurationsdateieinstellung [ca_bundle](#), der Befehlszeilenoption `--ca-bundle` oder der Umgebungsvariablen [AWS_CA_BUNDLE](#).

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Ihre Konfiguration verwendet nicht die richtige AWS-Region

Beispiel-Fehlertext:

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Fehler oder unerwartete Ergebnisse können auftreten, wenn eine für die von Ihnen angegebene nicht AWS-Service verfügbar ist AWS-Region oder wenn sich Ihre Ressourcen in einer anderen

befinden AWS-Region. Fehlerbehandlungsschritte finden Sie unter [the section called “Überprüfen Sie, ob AWS-Region Ihr AWS CLI Befehl verwendet”](#).

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Ihre TLS Version muss aktualisiert werden

Beispiel-Fehlertext:

```
$ aws s3 ls
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

Die verwendet AWS-Service eine Version davon TLS, die nicht mit der TLS Version Ihres Geräts kompatibel ist. Um dieses Problem zu beheben, aktualisieren Sie auf eine unterstützte TLS Version. Weitere Informationen finden Sie unter [the section called “Erzwingen einer Mindestversion TLS”](#).

[Zurück zum Seitenanfang](#)

Ungültige JSON Fehler

Beispiel-Fehlertext:

```
$ aws dynamodb update-table \
  --provisioned-throughput '{"ReadCapacityUnits':15,WriteCapacityUnits':10}' \
  --table-name MyDDBTable
Error parsing parameter '--provisioned-throughput': Invalid JSON: Expecting property
name enclosed in
double quotes: line 1 column 25 (char 24)
JSON received: {"ReadCapacityUnits":15,WriteCapacityUnits":10}
```

Wenn Sie einen AWS CLI Befehl verwenden, erhalten Sie eine "Invalid JSON"-Fehlermeldung. Dies ist normalerweise ein Fehler, der auftritt, wenn Sie einen Befehl mit einem erwarteten JSON Format eingeben und der Befehl dann AWS CLI nicht JSON richtig gelesen werden kann.

Mögliche Ursache: Sie haben keinen gültigen Wert eingegeben, der JSON verwendet AWS CLI werden soll

Vergewissern Sie sich, dass Sie für Ihren Befehl eine gültige JSON Eingabe vorgenommen haben. Wir empfehlen die Verwendung eines JSON Validators, falls JSON Sie Probleme mit der Formatierung haben.

Für eine fortgeschrittenere JSON Verwendung in der Befehlszeile sollten Sie die Verwendung eines JSON Befehlszeilenprozessors in Betracht ziehen `jq`, z. B. um Zeichenketten zu erstellen JSON. Weitere Informationen dazu `jq` finden Sie im [jq-Repository](#) unter GitHub.

[Zurück zum Seitenanfang](#)

Mögliche Ursache: Die Angebotsregeln Ihres Terminals verhindern, dass gültige Dokumente JSON an die AWS CLI

Bevor der etwas von einem Befehl AWS CLI empfängt, verarbeitet Ihr Terminal den Befehl mit seinen eigenen Regeln für Anführungszeichen und Escape-Zeichen. Aufgrund der Formatierungsregeln eines Terminals kann ein Teil Ihres JSON Inhalts entfernt werden, bevor der Befehl an den übergeben wird. AWS CLI Achten Sie beim Formulieren von Befehlen darauf, die [Anführungszeichenregeln des Terminals](#) zu berücksichtigen.

Verwenden Sie zur Fehlerbehebung den `echo`-Befehl, um zu sehen, wie die Shell mit Ihren Parametern umgeht:

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}
ReadCapacityUnits:15 WriteCapacityUnits:10
```


```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

Ändern Sie Ihren Befehl so lange, bis Ihr Befehl „Bis gültig“ zurückgegeben JSON wird.

Verwenden Sie für eine umfassendere Fehlerbehebung den `--debug`-Parameter, um die Debug-Protokolle anzuzeigen, da in diesen exakt zu sehen ist, was an die AWS CLI übergeben wurde:

```
$ aws dynamodb update-table \
  --provisioned-throughput '{"ReadCapacityUnits":15,WriteCapacityUnits":10}' \
  --table-name MyDDBTable \
  --debug
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.18.147
Python/2.7.18 Linux/5.4.196-119.356.amzn2int.x86_64 botocore/1.18.6
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - Arguments entered
to CLI:
['dynamodb', 'update-table', '--provisioned-throughput',
 '{"ReadCapacityUnits":15,WriteCapacityUnits":10}',
 '--table-name', 'MyDDBTable', '--debug']
```

Verwenden Sie die Angebotsregeln Ihres Terminals, um alle Probleme zu beheben, die Ihre JSON Eingabe beim Senden an die AWS CLI hat. Weitere Informationen zu Anführungszeichenregeln finden Sie unter [the section called “Anführungszeichen mit Zeichenfolgen”](#).

 Note

Wenn Sie Probleme mit der JSON Gültigkeitsbeschaffung für haben, empfehlen wir AWS CLI, die Angebotsregeln eines Terminals für die JSON Dateneingabe zu umgehen, indem Sie Blobs verwenden, um Ihre JSON Daten direkt an das weiterzuleiten. AWS CLI Weitere Informationen zu Blobs finden Sie unter [Blob](#).

[Zurück zum Seitenanfang](#)

Weitere Ressourcen

[Weitere Hilfe bei Ihren AWS CLI Problemen erhalten Sie in der AWS CLI Community unter GitHuboder in der AWS re:Post Community.](#)

[Zurück zum Seitenanfang](#)

Dokumentverlauf für das AWS CLI-Benutzerhandbuch

In der folgenden Tabelle werden wichtige Ergänzungen zum AWS Command Line Interface-Benutzerhandbuch ab Januar 2019 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Aktualisierte Anmelde- und Authentifizierungsinformationen.	Aktualisierte Anmeldeinformationen sowie Anweisungen und Beispiele für Authentifizierungsmethoden. Dies umfasst die Aktualisierung relevanter Konfigurationsseiten. Um dieser Zunahme an Dokumentation Rechnung zu tragen, wurden die relevanten Themen zu Anmeldeinformationen in den neuen Abschnitt Authentifizierung und Anmeldeinformationen verschoben.	31. März 2023
Inhalte für AWS CLI V1 und V2 sind jetzt auf ihre jeweiligen Leitfäden verteilt	Aus Gründen der Übersichtlichkeit und Benutzerfreundlichkeit sind die Inhalte für AWS CLI Version 1 und AWS CLIVersion 2 jetzt in eigene Leitfäden unterteilt. Informationen zu AWS CLI Version 2 finden Sie im aktuellen AWS Command Line Interface-Benutzerhandbuch .	2. November 2021
AWS CLI-Alias-Informationen hinzugefügt	AWS CLI-Alias-Informationen hinzugefügt. Aliase sind Verknüpfungen, die Sie in	11. März 2021

	AWS Command Line Interface (AWS CLI) erstellen können, um häufig verwendete Befehle oder Skripte zu verkürzen.	
Filterausgabeinformationen aktualisiert	Informationen zu Filtern aktualisiert und auf eine eigene Seite verschoben.	1. Februar 2021
Ankündigung der Einstellung von Python 2.7, 3.4 und 3.5	Python 2.7 ist durch die Grundlage für Python-Software ab dem 1. Januar 2020 veraltet. Kunden, die AWS CLI Version 1 nutzen, sollten auf Python 3 mit einem Minimum von Python 3.6 übergehen . Python-2.7-Support ist für neue Versionen der AWS CLI Version 1 ab 19.07.2021 veraltet. Python 3.4 und 3.5 ist ab dem 01.02.2021 veraltet.	29. Januar 2021
Beispiel für eine Skriptsprache von Amazon S3 hinzugefügt	Ein Beispiel für die Skriptsprache für den Amazon-S3-Lebenszyklus wurde hinzugefügt.	15. Oktober 2020
Beispiel für eine Skriptsprache von Amazon EC2 hinzugefügt	Skriptsprache-Beispiel für einen Amazon-EC2-Instance-Typen hinzugefügt.	15. Oktober 2020
Wiederholungsinformationen hinzugefügt	Eine Wiederholungsseite für Funktionen und das Verhalten von Wiederholungen in AWS CLI.	17. September 2020

Seite für die serverseitige und clientseitige Paginierung	Aktualisierte Paginierungs- und zentralisierte Informationen auf einer einzigen Seite.	17. August 2020
Seite „S3-Befehle“ aktualisiert	Die Seite „S3-Befehle auf hoher Ebene“ wurde mit neuen Beispielen und Ressourcen aktualisiert.	30. Juli 2020
Installationsinformationen aktualisiert	Die Installations-, Aktualisierungs- und Deinstallationsinformationen für Linux, macOS und Windows wurden aktualisiert.	19. Mai 2020
Aktualisiert, um den Support für Python 2.6 und 3.3 von der AWS CLI Version 1 zu entfernen	Ab dem 10. Januar 2020 unterstützt AWS CLI Version 1 nicht mehr die Verwendung von Python-Version 2.6 oder 3.3. Sie müssen auf eine neuere Version von Python aktualisieren, um AWS CLI Version 1.17 oder höher verwenden zu können.	10. Januar 2020
Neuer Abschnitt zur MFA	Es wurde ein neuer Abschnitt hinzugefügt, in dem beschrieben wird, wie auf die CLI mithilfe von Multi-Faktor-Authentifizierung und Rollen zugegriffen wird.	3. Mai 2019
Aktualisierung des Abschnitts „Verwenden der CLI“	Wichtige Verbesserungen und Ergänzungen bei den Nutzungsanweisungen und -verfahren.	7. März 2019

[Aktualisierung des Abschnitts
„Installieren der CLI“](#)

Wichtige Verbesserungen und Ergänzungen bei den AWS CLI-Installationsanweisungen und -verfahren.

7. März 2019

[Aktualisierung des Abschnitts
„Konfigurieren der CLI“](#)

Wichtige Verbesserungen und Ergänzungen bei den AWS CLI-Konfigurationsanweisungen und -verfahren.

7. März 2019